EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 1, Article Number 5850
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

On the Recognition Capacity of Abelian Graph Automata

Kyriakos Papadopoulos
Department of Mathematics, Kuwait University, Safat 13060, Kuwait

Abstract. This paper explores the recognition capacity of both unitary and non-unitary Abelian graph automata through the algebraic structure of graphoids. We investigate for the first time in the literature the recognition mechanism of non-unitary Abelian graph automata and prove that they can recognize graph languages which are beyond the recognition power of unitary graph automata. Consequently, we establish that the class of graph languages recognized by unitary automata is strictly contained within the class of those recognized by Abelian graph automata. These results manifest a proper hierarchy among graph automata classes and provide new insights into the recognition capabilities of graph automata.

2020 Mathematics Subject Classifications: 05C20,05C85,68T10,68R10
Key Words and Phrases: Graph Automata, Pattern Recognition, Graph Theory

1. Introduction

Graphs serve as powerful tools for modeling relationships and dependencies across various fields from network analysis to artificial intelligence [21], [25], [26], [1]. Automata theory, with its robust framework for recognizing structured data (see [11–14]), can be naturally extended into graph theory, enabling a systematic verification of graph characteristics [18],[8]. Central to this approach is the algebraic representation of graphs through magmoids, where the operations of graph product and graph sum are employed to represent graph structures (see [7], [9]) playing a role similar to monoids in string generation. A magmoid is defined as a doubly ranked set equipped with two operations that are associative, unitary, and canonically distributed [5].

As Engelfriet and Vereijken showed in [15], every graph can be represented in an infinite number of different ways inside a magmoid. To overcome this ambiguity, the algebraic structure of graphoids was introduced in [10] by considering the quotient magmoid obtained via a finite set equivalences.

By employing a special type of graphoid, called here *unitary graphoid*, automata operating on graphs were defined for the first time in [10]. As it was shown in [19], unitary graphoids are the simplest kind of abelian relational graphoids. The class of abelian

DOI: https://doi.org/10.29020/nybg.ejpam.v18i1.5850

Email address: kyriakos.papadopoulos@ku.edu.kw (K. Papadopoulos)

graphoids is in turn included in a larger class of graphoids called *relational graphoids* which are obtained by appropriately structuring the set of all relations on the state set of the graph automaton into a graphoid [19].

The definition of graph automata given in [10], implies that the recognition capacity of graph automata may vary with respect to the specific graphoid employed in their construction. However, this remains an open question, since the recognition mechanism of the various types of graph automata with respect to their underlying graphoid has not been further investigated besides unitary graphoids. Regarding the recognition ability of unitary graph automata, i.e., graph automata employing unitary graphoids, we know that they recognize for every positive integer k, the set of all directed k-colorable graphs without inputs and outputs [20]. Graph colorability and the associated chromatic number of a graph is a fundamental concept in graph theory, representing the minimum number of colors required to color the vertices of a graph such that no two adjacent vertices share the same color (see [2–4, 6, 16]). In this respect, it is manifested that graph automata can recognize complex algebraic structures even when operating on the simplest type of graphoid.

In this paper, we investigate further unitary graph automata and we prove that the set of directed graphs of length at most k is recognized automata of this type and hence belongs to the associate class of graph languages denoted here by $Urec(\Sigma)$. Using a similar construction, we also generalize the result of [20] by showing that, for every k, the set of all directed graphs with chromatic number at most k belongs to $Urec(\Sigma)$.

Moreover, in this paper, we examine, for the first time in the literature, abelian non-unitary graph automata. We show that the set of all graphs with an odd number of edges belongs to the class $Arec(\Sigma)$ of all graph languages recognized by abelian relational graph automata. This is proved by constructing an abelian non-unitary graph automaton with two states that can be structured into a group via the operation of modulo 2 addition, as dictated by the classification theorem of [19]. In addition to that, we prove that unitary graph automata can not recognize this graph language. As a consequence it is proved that $Urec(\Sigma)$ is properly contained in $Arec(\Sigma)$.

In the next section, we review the fundamental definitions of magmoids and hypergraphs. Section 3 presents the basic algebraic structures employed for the construction of graph automata and introduces the notions of relational, abelian and unitary graphoids. In the next section, we provide the formal definition of a graph automaton and the different types of graph automata corresponding to the introduced graphoids. Moreover, in this section we construct a unitary graph automaton recognizing the set of all directed graphs with a given chromatic number.

In Section 5 we introduce a type of abelian non-unitary graph automaton which recognizes all graphs with an odd number of edges. It is then proved that this graph language can not be recognized by a unitary graph automaton establishing the proper hierarchy of the corresponding classes.

2. Magmoids and Hypergraphs

In this section we first introduce the algebraic structure of magmoids and then we define directed graphs with specified inputs and outputs. We also present two graph operations, called graph sum and graph product. It turns out that these two operations organize the set of graphs can into a magmoid.

Given a set S we denote by S^* the set of all strings constructed from the elements of S, we denote by ε the empty string, and we set $S^+ = S^* - \{\varepsilon\}$. A doubly ranked set $A = (A_{m,n})_{m,n\in\mathbb{N}}$, consists of a set A along with a function

$$\text{rank} : A \to \mathbb{N} \times \mathbb{N}$$

and is defined as $A_{m,n} = \{a \in A \mid \text{rank}(a) = (m, n)\}$. For simplicity, we will omit the subscript and denote a doubly ranked set by $A = (A_{m,n})$.

A *magmoid* is a doubly ranked set $M = (M_{m,n})$ equipped with two operations

$$\circ : M_{m,n} \times M_{n,k} \to M_{m,k}, \quad \Box : M_{m,n} \times M_{m',n'} \to M_{m+m',n+n'}$$

that are associative in the usual way and satisfy the distributive law

$$(f \circ g) \Box (f' \circ g') = (f \Box f') \circ (g \Box g')$$

whenever the operations are defined. Additionally, there exists a sequence of constants $e_n \in M_{n,n}$, called units, such that

$$e_m \circ f = f = f \circ e_n, \;\; e_0 \Box f = f = f \Box e_0, \; e_m \Box e_n = e_{m+n}$$

for all $f \in M_{m,n}$ and all $m, n \in \mathbb{N}$. The final equation implies that the elements e_n are uniquely determined by e_1, which we shall denote simply by e. The free magmoid $mag(\Sigma)$ generated by a doubly ranked set Σ is constructed in [7]. The sets $Rel_{m,n}(Q)$ of all relations from Q^m to Q^n are defined as

$$Rel_{m,n}(Q) = \{R \mid R \subseteq Q^m \times Q^n\}$$

and can be structured into a magmoid with $\circ$ as the usual relational composition and $\Box$ defined as follows: for $R \in Rel_{m,n}(Q)$ and $S \in Rel_{m',n'}(Q)$,

$$R \Box S = \{(u_1u_2, v_1v_2) \mid (u_1, v_1) \in R \text{ and } (u_2, v_2) \in S\},$$

where $u_1 \in Q^m$, $u_2 \in Q^{m'}$, $v_1 \in Q^n$, $v_2 \in Q^{n'}$. Here, $Q^0 = \{\varepsilon\}$, with ε as the empty word of Q^*. The units are defined as

$$e_0 = \{(\varepsilon, \varepsilon)\} \quad \text{and} \quad e = \{(g, g) \mid g \in Q\}.$$

We denote by $Rel(Q) = (Rel_{m,n}(Q))$ the magmoid constructed in this manner, referred to as the *relational magmoid of* Q.

An (m,n)-(hyper)graph $G = (V, E, s, t, l, \text{begin}, \text{end})$ with edge labels from a doubly ranked set $\Sigma = (\Sigma_{m,n})$ is a structure consisting of a finite set of vertices V, a finite set of edges E, the source and target functions $s : E \to V^+$ and $t : E \to V^+$, the labeling function $l : E \to \Sigma$ such that $\text{rank}(l(e)) = (|s(e)|, |t(e)|)$ and sequences of begin and end nodes, begin $\in V^*$ and end $\in V^*$, with $|\text{begin}| = m$ and $|\text{end}| = n$. Note that vertices may repeat in the begin and end sequences, as well as in edge sources and targets. The set of all (m,n)-graphs over Σ is denoted by $GR_{m,n}(\Sigma)$, and we define $GR(\Sigma) = (GR_{m,n}(\Sigma))_{m,n\in\mathbb{N}}$.

A *path* of length n inside a graph G is a sequence of edges $e_1, \ldots, e_n$, such that, for every $i < n-1$, there exists a node v_i of G, that appears in both the strings $t(e_i)$ and $s(e_{i+1})$. We say that a path $e_1, \ldots, e_n$ inside G is a *cycle*, if there exists a node v of G, that appears in both the strings $s(e_1)$ and $t(e_n)$. The *length* of a graph G is defined as the length of the longest path inside G, see [17, 24, 27, 28]. Note that if a graph G has a cycle then there will be a path of length n inside G for any $n \in \mathbb{N}$. Hence the length of a graph is finite if and only if the graph is acyclic.

Ordinary graphs are obtained as a special case of hypergraphs, where each hyperedge is binary, i.e. for every edge e of the graph it holds $|s(e)| = |t(e)| = 1$. A graph is called *unlabeled* if every edge has the same label. An ordinary, unlabeled graph is called *conventional* graph. A graph has no input (resp. output) if the begin (resp. end) sequence is ε. A graph without input and output is by definition a $(0,0)$-graph. Conventional $(0,0)$-graphs is the most commonly examined type of directed graphs in the literature.

The *product* and the *sum* of two graphs where introduced by Engelfriet and Vereijken in [15], see also [7, 10, 19]. For an (m,n)-graph G and an (n,k)-graph H, the product $G \circ H$ is the (m,k)-graph obtained by taking the disjoint union of the two graphs and identifying the i^{th} end node of G with the i^{th} begin node of H for all i. The begin and end sequences of the product are respectively the begin sequence of G and the end sequence of H. The sum $G \,\square\, H$ of two arbitrary graphs G and H is obtained by taking their disjoint union and concatenating their begin and end node sequences. For each $n \in \mathbb{N}$, let E_n represent the discrete graph of rank (n,n) with nodes $x_1, \ldots, x_n$ and begin $=$ end $= x_1 \cdots x_n$; we denote E_1 simply by E. It is straightforward to verify that $GR(\Sigma) = (GR_{m,n}(\Sigma))$ forms a magmoid with operations product and sum, the units are the graphs E_n.

3. Graphoids

In this section, we explore graph automata by utilizing the algebraic structure of graphoids as defined in [10]. Let $I_{p,q}$ denote the discrete (p,q)-graph that has a single node x with begin and end sequences both formed by x repeated p and q times, respectively. We also let Π be the discrete $(2,2)$-graph with two nodes, x and y, whose begin sequence is xy and end sequence is yx. For each $\sigma \in \Sigma_{m,n}$, we denote by σ the (m,n)-graph with a single edge and $m+n$ nodes labeled $x_1, \ldots, x_m, y_1, \ldots, y_n$. The edge is labeled σ, with the begin (resp. end) sequence of the graph as the sequence of sources (resp. targets) of the edge: $x_1 \cdots x_m$ (resp. $y_1 \cdots y_n$).

Engelfriet and Vereijken, in [15], proposed an algorithm that inductively constructs any graph $G \in GR(\Sigma)$ from the set $\Sigma \cup \{\Pi, I_{01}, I_{21}, I_{10}, I_{12}\}$ by using graph product and graph sum. However, a given graph can be constructed in infinitely many ways. This issue was addressed by identifying a finite set $\mathcal{E}$ of equations with the property that two expressions represent the same graph if and only if one can be transformed into the other using these equations [7]. Thus, the equations in $\mathcal{E}$ are valid in $GR(\Sigma)$, and magmoids satisfying this property are referred to as *graphoids*.

Formally, a graphoid $\mathbf{M} = (M, D_M)$ consists of a magmoid M and a set $D_M = \{s, d_{01}, d_{21}, d_{10}, d_{12}\}$, where $s \in M_{2,2}$ and $d_{\kappa\lambda} \in M_{\kappa,\lambda}$, such that the following equations hold:

$$s \circ s = e_2, \tag{1}$$
$$(s \,\Box\, e) \circ (e \,\Box\, s) \circ (s \,\Box\, e) = (e \,\Box\, s) \circ (s \,\Box\, e) \circ (e \,\Box\, s)\,, \tag{2}$$
$$(e \,\Box\, d_{21}) \circ d_{21} = (d_{21} \,\Box\, e) \circ d_{21}, \tag{3}$$
$$(e \,\Box\, d_{01}) \circ d_{21} = e, \tag{4}$$
$$s \circ d_{21} = d_{21}, \tag{5}$$
$$(e \,\Box\, d_{01}) \circ s = (d_{01} \,\Box\, e)\,, \tag{6}$$
$$(s \,\Box\, e) \circ (e \,\Box\, s) \circ (d_{21} \,\Box\, e) = (e \,\Box\, d_{21}) \circ s, \tag{7}$$
$$d_{12} \circ (e \,\Box\, d_{12}) = d_{12} \circ (d_{12} \,\Box\, e)\,, \tag{8}$$
$$d_{12} \circ (e \,\Box\, d_{10}) = e, \tag{9}$$
$$d_{12} \circ s = d_{12}, \tag{10}$$
$$s \circ (e \,\Box\, d_{10}) = (d_{10} \,\Box\, e)\,, \tag{11}$$
$$(d_{12} \,\Box\, e) \circ (e \,\Box\, s) \circ (s \,\Box\, e) = s \circ (e \,\Box\, d_{12})\,, \tag{12}$$
$$d_{12} \circ d_{21} = e, \tag{13}$$
$$(d_{12} \,\Box\, e) \circ (e \,\Box\, d_{21}) = d_{21} \circ d_{12}, \tag{14}$$
$$s_{m,1} \circ (p \,\Box\, e) = (e \,\Box\, p) \circ s_{n,1}, \quad \text{for all } p \in M_{m,n}. \tag{15}$$

where $s_{m,1}$ is defined inductively by s and represents the graph associated with the permutation that interchanges the last n numbers with the first one [7]. Notably, Equation (15) only needs to hold for elements of Σ to be valid for every element of a magmoid generated by Σ (see [7]). Hence, the pair $(GR(\Sigma), D_{GR(\Sigma)})$ with

$$D = \{\Pi, I_{01}, I_{21}, I_{10}, I_{12}\},$$

is a graphoid and is, in fact, the free graphoid generated by Σ as illustrated in [10].

For graphoids (M, D_M) and $(M', D_{M'})$, a magmoid morphism $H : M \to M'$ that preserves D-sets, i.e., $H(s) = s'$ and $H(d_{\kappa\lambda}) = d'_{\kappa\lambda}$, is called a morphism of graphoids.

A graphoid $(Rel(Q), D_{Rel(Q)})$ formed from the magmoid of relations $Rel(Q)$ is called *relational graphoid*. If the element $s \in D_{Rel(Q)}$ of a relational graphoid is

$$s = \{(g_1 g_2, g_2 g_1) \mid g_1, g_2 \in Q\}, \tag{16}$$

then the pair $(Rel(Q), D_{Rel(Q)})$ is called *abelian graphoid* (see [19]).

The *unitary graphoid* $\mathcal{U}(\mathbf{Q}) = (Rel(Q), D^U_{Rel(Q)})$, which was used for introducing graph automata in [10], is an abelian graphoid constructed by defining, in addition to s as above, the elements $d_{01}, d_{21}, d_{10}, d_{12}$ as follows

$$d_{01} = \{(\varepsilon, g) \mid g \in Q\}, \tag{17}$$
$$d_{21} = \{(gg, g) \mid g \in Q\}, \tag{18}$$
$$d_{10} = \{(g, \varepsilon) \mid g \in Q\}, \tag{19}$$
$$d_{12} = \{(g, gg) \mid g \in Q\}. \tag{20}$$

In [19] it is proved that a set of states Q can be structured into an abelian graphoid if and only if it can be partitioned into disjoint abelian groups with operations derived from d_{21} and unit derived from d_{10} (or equivalently from d_{12} and d_{01}). In this setup the unitary graphoid $\mathcal{U}(\mathbf{Q})$, introduced above, corresponds to the partition of the state set $Q = \{q_1, q_2, \ldots, q_k\}$ into k singleton sets $Q = \{q_i\}$ which can then be structured into k trivial groups.

4. Graph Automata

In this section we introduce different types of relational automata operating on graphs, corresponding to different types of relational graphoids.

A *relational graph automaton*, as introduced in [10], is a structure

$$\mathcal{A} = (\Sigma, Q, (Rel(Q), D_{Rel(Q)}), \delta_{\mathcal{A}}, I_{\mathcal{A}}, T_{\mathcal{A}}),$$

where Σ is the doubly ranked set of hyperedge labels, Q is a finite set of states, $\delta_{\mathcal{A}} : \Sigma \to Rel(Q)$ is the doubly ranked transition function, and $I_{\mathcal{A}}, T_{\mathcal{A}}$ are initial and final rational subsets of Q^*.

According to Theorem 3 of [10], the function $\delta_{\mathcal{A}}$ is uniquely extended to a morphism of graphoids

$$\bar{\delta}_{\mathcal{A}} : GR(\Sigma) \to (Rel(Q), D_{Rel(Q)}),$$

where $\bar{\delta}_{\mathcal{A}}(I_{ij}) = d_{ij}$ and $\bar{\delta}_{\mathcal{A}}(\Pi) = s$. The behavior of $\mathcal{A}$ is defined by

$$|\mathcal{A}| = \{F \mid F \in GR_{m,n}(\Sigma),\ \bar{\delta}_{\mathcal{A}}(F) \cap (I^{(m)}_{\mathcal{A}} \times T^{(n)}_{\mathcal{A}}) \neq \emptyset,\ m, n \in \mathbb{N}\},$$

where $I^{(m)}_{\mathcal{A}} = I_{\mathcal{A}} \cap Q^m$ and $T^{(n)}_{\mathcal{A}} = T_{\mathcal{A}} \cap Q^n$. Notice that, due to the use of the relational graphoid, the resulting graph automata are non-deterministic.

Graph automata can be obtained similarly to the above definition using non-relational graphoids, although this has never been examined in the literature. In the particular case that the relational graphoid satisfies Eq. 16, then the corresponding relational graph automaton is called *abelian graph automaton.* In addition, if it satisfies Eq. 16 and Eqs.17-20, it is called *unitary graph automaton.*

The set of all graph languages over the doubly ranked set Σ, recognized by a relational graph automaton is denoted by $Rrec(\Sigma)$. Analogously, we denote by $Arec(\Sigma)$ the set of

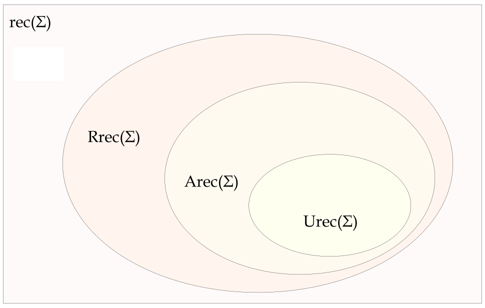


Figure 1: The class hierarchy of graph automata

graph languages recognized by abelian graph automata and by $Urec(\Sigma)$ the set of graph languages recognized by unitary graph automata.

In [20] it was shown that the set of conventional k-colorable graphs without inputs and outputs belongs to $Urec(\Sigma)$ for every positive integer k. This was achieved by constructing a unitary graph automaton able to read all ordinary $(0, 0)$-graphs with identical edge labels, and recognize those that can be assigned a proper k-coloring.

Next we will construct a unitary graph automaton recognizing all conventional graphs of length at most k.

Proposition 1. *Given $k \in \mathbb{N}$, the set of conventional graphs of length at most k belongs to $Urec(\Sigma)$.*

Proof. We are first going to construct a graph automaton recognizing all ordinary unlabeled $(0, 0)$-graphs of length at most k. Consider the unitary graph automaton

$$\mathcal{A}^k_{len} = (\Sigma, Q, \mathcal{U}(\mathbf{Q}), \delta_{\mathcal{A}^k_{len}}, I_{\mathcal{A}^k_{len}}, T_{\mathcal{A}^k_{len}})$$

with $\Sigma = \Sigma_{1,1} = \{a\}$, state set $Q = \{1, 2, ..., k + 1\}$, transition function given by

$$\delta_{\mathcal{A}^k_{len}}(a) = \{(i, j) \mid i, j \in Q, i < j\},$$

and $I_{\mathcal{A}^k_{len}} = T_{\mathcal{A}^k_{len}} = \{\varepsilon\}$.

From this construction we see that any successful transition inside this automaton will increase the state index for each edge of the graph it reads. As a result, by taking into account the path length definition, there will be not successful transition for any graph that has a path of length larger than k but every graph of smaller length will be recognized.

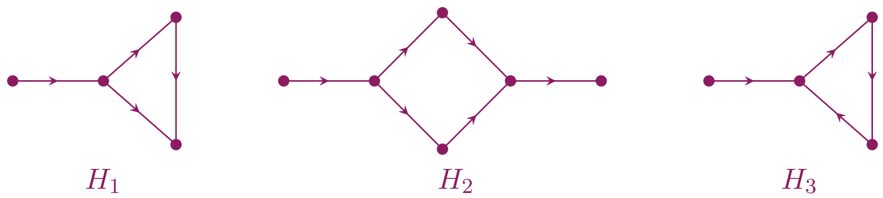


Figure 2: Three unlabeled $(0, 0)$graphs with binary edges

We will illustrate this by examining the three ordinary graphs shown in Figure 2, starting with H_1, a graph of length 3. We know that $GR(\Sigma)$ is the free graphoid, hence the operation of any automaton does not depend to the specific representation of H_1 we will employ. Below is a representation of H_1 where graph product and graph sum are denoted, for simplicity, by horizontal and vertical concatenation.

$$H_1 = I_{01}\, a\, I_{12} \begin{pmatrix} a \\ a \end{pmatrix} \begin{pmatrix} a \\ E \end{pmatrix} I_{21}\, I_{10}$$

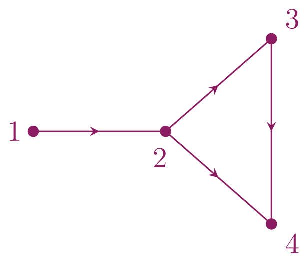
The image of H_1 by the transition function of $\mathcal{A}^3_{len}$ is

$$\overline{\delta_{\mathcal{A}^3_{len}}}(H_1) = d_{01}\, \delta_{\mathcal{A}^3_{len}}(a)\, d_{12} \begin{pmatrix} \delta_{\mathcal{A}^3_{len}}(a) \\ \delta_{\mathcal{A}^3_{len}}(a) \end{pmatrix} \begin{pmatrix} \delta_{\mathcal{A}^3_{len}}(a) \\ e \end{pmatrix} d_{21}\, d_{10}$$

and an accepting state map for it will be

$$\{\varepsilon\}d_{01}\{1\}\delta_{\mathcal{A}^3_{len}}(a)\{2\}d_{12} \begin{Bmatrix} 2 \\ 2 \end{Bmatrix} \begin{pmatrix} \delta_{\mathcal{A}^3_{len}}(a) \\ \delta_{\mathcal{A}^3_{len}}(a) \end{pmatrix} \begin{Bmatrix} 3 \\ 4 \end{Bmatrix} \begin{pmatrix} \delta_{\mathcal{A}^3_{len}}(a) \\ e \end{pmatrix} \begin{Bmatrix} 4 \\ 4 \end{Bmatrix} d_{21}\{4\}\, d_{10}\{\varepsilon\}$$

where the states are indicated in brackets. Hence the graph is recognized by $\mathcal{A}^3_{len}$. Represented on the graph, the states that the automaton reaches at each vertex are



The graph H_2 of Figure 2 has length 4. We can employ the below representation

$$H_2 = I_{01}\, a\, I_{12} \begin{pmatrix} a \\ a \end{pmatrix} \begin{pmatrix} a \\ a \end{pmatrix} I_{21}\, a\, I_{10}$$

and observe that this graph can not be accepted by $\mathcal{A}^3_{len}$ since any possible transition can reach at most to the last $\delta_{\mathcal{A}^3_{len}}(a)$ before halting as seen below.

$$\{\varepsilon\}d_{01}\{1\}\delta_{\mathcal{A}^3_{len}}(a)\{2\}d_{12} \begin{Bmatrix} 2 \\ 2 \end{Bmatrix} \begin{pmatrix} \delta_{\mathcal{A}^3_{len}}(a) \\ \delta_{\mathcal{A}^3_{len}}(a) \end{pmatrix} \begin{Bmatrix} 3 \\ 3 \end{Bmatrix} \begin{pmatrix} \delta_{\mathcal{A}^3_{len}}(a) \\ \delta_{\mathcal{A}^3_{len}}(a) \end{pmatrix} \begin{Bmatrix} 4 \\ 4 \end{Bmatrix} d_{21}\{4\}\, \delta_{\mathcal{A}^3_{len}}(a)$$

The third graph of Figure 2 has a cycle and hence its length is infinite. A possible representation of H_3 is given below.

$$H_3 = I_{01}\, a\, I_{12} \begin{pmatrix} a \\ E \end{pmatrix} \begin{pmatrix} a \\ E \end{pmatrix} \begin{pmatrix} a \\ E \end{pmatrix} I_{21}\, I_{10}$$

From this representation, we see that H_3 can not be accepted by $\mathcal{A}^3_{len}$ since any transition will halt when reaching d_{21} as illustrated below.

$$\{\varepsilon\}d_{01}\{1\}\delta_{\mathcal{A}^3_{len}}(a)\{2\}d_{12}\begin{Bmatrix}2\\2\end{Bmatrix}\begin{pmatrix}\delta_{\mathcal{A}^3_{len}}(a)\\e\end{pmatrix}\begin{Bmatrix}3\\2\end{Bmatrix}\begin{pmatrix}\delta_{\mathcal{A}^3_{len}}(a)\\e\end{pmatrix}\begin{Bmatrix}4\\2\end{Bmatrix}\begin{pmatrix}\delta_{\mathcal{A}^3_{len}}(a)\\e\end{pmatrix}\begin{Bmatrix}5\\2\end{Bmatrix}d_{21}$$

The graph automaton $\mathcal{A}^3_{len}$ can be generalized to consider every (m, n)-graph if we modify the initial and final sequences by setting

$$I_{\mathcal{A}^k_{len}} = Q^m \text{ and } T_{\mathcal{A}^k_{len}} = Q^n.$$

This will not affect the behavior of the automaton as it is evident from the Equations 17-20 which hold for every unitary automaton.

From the above Proposition we get that the graph language Len^k that consists of all conventional graphs with length at most k lies in the class $Urec(\Sigma)$, for every positive integer k. Using the same argument as in the end of the above proof we can generalize the result of [20] to obtain the following.

Proposition 2. *Given $k \in \mathbb{N}$, the set of all conventional graphs with chromatic number at most k, belongs to $Urec(\Sigma)$.*

5. Non-unitary abelian graph automata

Non-unitary graph automata have never been examined in the literature and their recognition capacity remains unknown. As a result, we don't know if the class hierarchy depicted in Figure 1 is proper.

In this section, we will introduce non-unitary abelian graph automata operating by virtue of graphoids associated to non-trivial groups. We will identify a graph language recognized by such a graph automaton and show that it doesn't belong to $Urec(\Sigma)$, demonstrating that $Urec(\Sigma)$ is properly included in $Arec(\Sigma)$.

For this we define the abelian graphoid $\mathbf{G_2(0, 1)}$ that is obtained by the trivial partition of the set $\{0, 1\}$ to a single set that is structured into a group via the operation of addition mod 2.

+	0	1
0	0	1
1	1	0

From Theorem 3 of [19], we get that the elements of $D_{\mathbf{G_2(0,1)}}$, besides s will be as follows

$$d_{21} = \{(01, 1), (10, 1)\},\ d_{10} = \{(0, \varepsilon)\},$$
$$d_{12} = \{(1, 01), (1, 10)\},\ d_{01} = \{(\varepsilon, 0)\}.$$

Clearly, graph automata operating on $\mathbf{G_2(0, 1)}$ are abelian but non-unitary. Hence the graph languages they recognize belong to $Arec(\Sigma)$ but necessarily to $Urec(\Sigma)$.

Now we consider the graph language $L^{(1,1)}_{od}$ of all conventional $(1, 1)$-graphs with an odd number of edges. We prove the following proposition.

Proposition 3. *The graph language $L_{od}^{(1,1)}$ belongs to $Arec(\Sigma)$.*

Proof. We use this $\mathbf{G_2(0,1)}$ to construct the following graph automaton

$$\mathcal{A}_{od} = (\Sigma, \{0,1\}, \mathbf{G_2(0,1)}, \delta_{\mathcal{A}_{od}}, I_{\mathcal{A}_{od}}, T_{\mathcal{A}_{od}})$$

with $\Sigma = \Sigma_{1,1} = \{a\}$, state set $Q = \{0,1\}$, $I_{\mathcal{A}_{od}} = \{0\}$, $T_{\mathcal{A}_{od}} = \{1\}$ and the transition function given by

$$\delta_{\mathcal{A}_{od}}(a) = \{(0,1),(1,0)\}.$$

This graph automaton recognizes the graph language $L_{od}^{(1,1)}$ of all conventional $(1,1)$-graphs with an odd number of edges. To illustrate this, we consider the following graphs, where the nodes of the begin and end sequences are designated by b_1 and e_1 respectively.

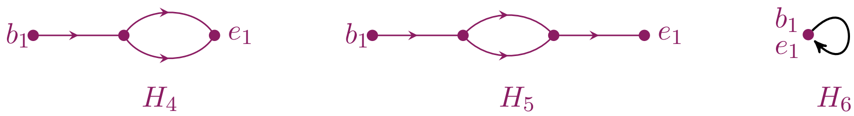


Figure 3: Three unlabeled $(1,1)$-graphs with binary edges

The image of a representation of H_4 by the transition function of $\mathcal{A}_{od}$ gives

$$\bar{\delta}_{\mathcal{A}_{od}}(H_4) = \delta_{\mathcal{A}_{od}}(a)\, d_{12} \begin{pmatrix} \delta_{\mathcal{A}_{od}}(a) \\ \delta_{\mathcal{A}_{od}}(a) \end{pmatrix} d_{21}$$

and an accepting sequence of states is

$$\{0\}\delta_{\mathcal{A}_{od}}(a)\{1\}d_{12} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \begin{pmatrix} \delta_{\mathcal{A}_{od}}(a) \\ \delta_{\mathcal{A}_{od}}(a) \end{pmatrix} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} d_{21}\{1\}$$

which shows that H_4 is recognized by $\mathcal{A}_{od}$. Similarly, we can see that for H_5 and H_6 the following sequences of states can be respectively obtained

$$\{0\}\delta_{\mathcal{A}_{od}}(a)\{1\}d_{12} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \begin{pmatrix} \delta_{\mathcal{A}_{od}}(a) \\ \delta_{\mathcal{A}_{od}}(a) \end{pmatrix} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} d_{21}\{1\}\delta_{\mathcal{A}_{od}}(a)\{0\}$$

$$\text{and } \{0\}d_{12} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \begin{pmatrix} \delta_{\mathcal{A}_{od}}(a) \\ e \end{pmatrix} \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} d_{21}\{0\},$$

which shows that H_6 is accepted but not H_5. Notice that, according to [10], the behavior of the graph automaton will be the same regardless of which representation we select.

Next we are going to prove that $L_{od}^{(1,1)}$ does not belong to $Urec(\Sigma)$, thus obtaining the following

Proposition 4. *The class $Urec(\Sigma)$ is properly contained in $Arec(\Sigma)$.*

Proof. Assume that there is a unitary automaton $\mathcal{A}^U_{od}$ recognizing $L^{(1,1)}_{od}$ and a graph $G \in L^{(1,1)}_{od}$. Then there will be states $a \in I_{\mathcal{A}^U_{od}}$ and $b \in T_{\mathcal{A}^U_{od}}$, and the following accepting transition.

$$\{a\}\delta_{\mathcal{A}^U_{od}}(G)\{b\}$$

We consider now the graph

$$G_p = I_{12} \circ (G \,\square\, G) \circ I_{21},$$

which is accepted by the graph automaton $\mathcal{A}^U_{od}$ with accepting transition as shown below.

$$\{a\}d_{12} \begin{Bmatrix} a \\ a \end{Bmatrix} \begin{pmatrix} \delta_{\mathcal{A}^U_{od}}(G) \\ \delta_{\mathcal{A}^U_{od}}(G) \end{pmatrix} \begin{Bmatrix} b \\ b \end{Bmatrix} d_{21}\{b\}$$

This is a contradiction since G_p has an even number of edges. Hence $L^{(1,1)}_{od}$ does not belong to $Urec(\Sigma)$ and from Proposition 3 we obtain the result.

6. Conclusion

We investigated unitary and non-unitary Abelian graph automata by leveraging the algebraic structure of graphoids and demonstrated that unitary graph automata effectively recognize sets of directed graphs with chromatic numbers bounded by any given integer, thereby extending existing results in the literature. In addition, we introduced and examined non-unitary Abelian graph automata, showing that they can recognize graph languages—such as graphs the set of conventionawith an odd number of edges—that lie beyond the recognition capabilities of unitary graph automata. This establishes a strict hierarchy between the graph languages recognized by these two classes of automata namely, $Urec(\Sigma) \subset Arec(\Sigma)$.

The presented results demonstrate the robust expressive power of non-unitary automata and provide new insights into the interplay between graph automata and their corresponding algebraic structures. By showing how variations in the underlying graphoid influence recognition capability, this work paves the way for further exploration of the theoretical and practical implications of graph automata.

As a continuation of this study, the recognition mechanisms of non-unitary graph automata derived from graphoids based on non-Abelian groups with more than two elements need to be investigated. Such an exploration could further expand the hierarchy of graph language recognition within the class of Abelian graph automata. Additionally, while constructing non-Abelian and non-relational graphoids is theoretically feasible, it remains an open problem. The successful development of such machines would enable comparisons with existing unitary and Abelian graph automata, thereby broadening our understanding of the recognition capabilities of graph automata.

Moreover, drawing parallels to the string case, we can compare the recognition power of graph automata to the syntactic recognizability of graphs, as introduced in [9]. We can also investigate the possible construction of fuzzy graph automata, similarly to syntactic fuzzy recognition, as described in [22, 23].

References

[1] Shaikh Ibrahim Abdullah, Sovan Samanta, Kajal De, et al. Properties of the forgotten index in bipolar fuzzy graphs and applications. *Scientific Reports*, 14:28264, 2024.

[2] Noga Alon, Janos Pach, and Josef Solymosi. Ramsey-type theorems with forbidden subgraphs. *Combinatorica*, 21:155–170, 2001.

[3] Kenneth Appel and Wolfgang Haken. Every planar map is four colorable. i. discharging. *Illinois Journal of Mathematics*, 21:429–490, 1977.

[4] Kenneth Appel, Wolfgang Haken, and John Koch. Every planar map is four colorable. ii. reducibility. *Illinois Journal of Mathematics*, 21:491–567, 1977.

[5] Andre Arnold and Max Dauchet. Théorie des magmoides. *RAIRO Theoret. Inform. Appl.*, 12:235–257, 1978.

[6] Edgar Asplund and Branko Grünbaum. On a coloring problem. *Mathematica Scandinavica*, 8:181–188, 1960.

[7] Symeon Bozapalidis and Antonios Kalampakas. An axiomatization of graphs. *Acta Inform.*, 41:19–61, 2004.

[8] Symeon Bozapalidis and Antonios Kalampakas. Automata on patterns and graphs. In *Proceedings of the 1st Conference on Algebraic Informatics*, pages 31–52, 2006.

[9] Symeon Bozapalidis and Antonios Kalampakas. Recognizability of graph and pattern languages. *Acta Inform.*, 42:553–581, 2006.

[10] Symeon Bozapalidis and Antonios Kalampakas. Graph automata. *Theoret. Comput. Sci.*, 393:147–165, 2008.

[11] Sander Bruggink and Barbara König. Recognizable languages of arrows and cospans. *Mathematical Structures in Computer Science*, 28(8):1290 – 1332, 2018.

[12] Bruno Courcelle. On recognizable sets and tree automata. In *Algebraic Techniques*, pages 93–126. Academic Press, 1989.

[13] Alexsander Andrade de Melo and Mateus de Oliveira Oliveira. Second-order finite automata. *Theory of Computing Systems*, 66(4):861 – 909, 2022.

[14] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Automata and Logic: A Microcosm of Finite Model Theory*, pages 107–118. Springer Berlin Heidelberg, 1995.

[15] Joost Engelfriet and Jan Joris Vereijken. Context-free graph grammars and concatenation of graphs. *Acta Informatica*, 34:773–803, 1997.

[16] Paul Erdős. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[17] Tibor Gallai. On directed paths and circuits. In P. Erdös and G. Katona, editors, *Theory of Graphs*, pages 115–118. Academic Press, New York, 1968.

[18] Antonios Kalampakas. The syntactic complexity of eulerian graphs. *Lecture Notes in Computer Science*, 4728:208 – 217, 2007.

[19] Antonios Kalampakas. Graph automata: The algebraic properties of abelian relational graphoids. *Lecture Notes in Computer Science*, 7020:168–182, 2011.

[20] Antonios Kalampakas. Graph automata and graph colorability. *European Journal of Pure and Applied Mathematics*, 16(1):112–120, 2023.

[21] Antonios Kalampakas. Wardrop optimal networks. *Eur. J. Pure Appl. Math.*,

17(4):2448–2466, 2024.

[22] Antonios Kalampakas, Stefanos Spartalis, and Lazaros Iliadis. Syntactic recognizability of graphs with fuzzy attributes. *Fuzzy Sets and Systems*, 229:91–100, 2013.

[23] Antonios Kalampakas, Stefanos Spartalis, Lazaros Iliadis, and Elias Pimenidis. Fuzzy graphs: algebraic structure and syntactic recognition. *Artif. Intell. Rev.*, 42:479–490, 2014.

[24] Chen-Yu Lin. Simple proofs of results on paths representing all colors in proper vertex-colorings. *Graphs and Combinatorics*, 23:201–203, 2007.

[25] Rupkumar Mahapatra, Sovan Samanta, Madhumangal Pal, et al. A study on linguistic z-graph and its application in social networks. *Mathematics*, 12(18):2898, 2024.

[26] Jambi Ratna Raja, Jeong Gon Lee, Dhanraj Dhotre, et al. Fuzzy graphs and their applications in finding the best route, dominant node and influence index in a network under the hesitant bipolar-valued fuzzy environment. *Complex & Intelligent Systems*, 10(4):5195–5211, 2024.

[27] Bernard Roy. Nombre chromatique et plus longs chemins d'un graph. *Rev. AFIRO*, 1:127–132, 1967.

[28] Douglas West. *Introduction to Graph Theory.* Prentice-Hall, New Jersey, 1996.