Mathematics and Statistics 12(6): 545-552, 2024 http://www.hrpub.org
DOI: 10.13189/ms.2024.120605

Exploring the Connections of Finite Group Automata, Group Machines and Group Machine Recognizer: Analyzing Their Characteristics

Sathiyasorubini G, Venkatesan R*

Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu - 603203, Tamil Nadu, India

Received July 30, 2024; Revised November 10, 2024; Accepted November 21, 2024

Cite This Paper in the Following Citation Styles

***(a)**: [1] Sathiyasorubini G, Venkatesan R, "Exploring the Connections of Finite Group Automata, Group Machines and Group Machine Recognizer: Analyzing Their Characteristics," Mathematics and Statistics, Vol.12, No.6, pp. 545-552, 2024. DOI: 10.13189/ms.2024.120605*

***(b)**: Sathiyasorubini G, Venkatesan R (2024). Exploring the Connections of Finite Group Automata, Group Machines and Group Machine Recognizer: Analyzing Their Characteristics, Mathematics and Statistics, 12(6), 545-552. DOI: 10.13189/ms.2024.120605*

Abstract This paper presents a comprehensive examination of the underlying structures and behaviours of finite group automata and group machines, delving into their intricate relationships and properties. Researchers have developed a comprehensive structure for finite group automata applicable to any finite group. Their work utilizes state complexity (SC) and accepting state complexity (ASC) as key metrics. They have also computed syntactic and quotient complexities (SNC and QC) specifically for cyclic groups. While the primary emphasis lies on cyclic groups, their versatile methodology establishes a foundation for broadening these complexity analyses to encompass other categories of finite groups. Building upon existing research on finite group automata, this study investigates the structures of group machines, group machine recognizers, and their properties, including strong connectivity, cyclicality, perfection, bideterminism, and permutation behaviours. Our analysis reveals the interconnected nature of group machines and group machine recognizers, shedding light on their distinct characteristics. A key finding of this research is that all group machines are finite group automata, although the converse is not always true. By differentiating these structures based on their properties, we can effectively handle machines according to their characteristics. This study contributes to the advancement of research in this field by providing a thorough understanding of the fundamental structures and behaviours of finite group automata and group machines, ultimately enriching the theoretical foundations of computational models.

Keywords Finite Group Automata, Group Machine, Group Machine Recognizer

1 Introduction

The concept of finite group automata, pioneered by Plotkin et.al [1], has found significant applications in diverse mathematical domains. Kelarev [2] further extended this concept to elucidate the Jacobson radicals of incidence rings, offering insights applicable across all finite p-groups. Sathiyasorubini and Venkatesan [3] explored deterministic finite automata intricacies, particularly focusing on regular languages derived through operations such as union, difference, reversal etc. The authors in [4, 5], also elucidated these operations, underscoring the multifaceted nature and broad applicability of finite group automata in theoretical computer science (TCS) and related fields.

The analysis of the aforementioned research involved delineating three distinct finite group automata structures and conducting a comprehensive examination of their complexities. Specifically, the study scrutinized four dimensions of complexity associated with finite group automata studied in terms of four complexities [6, 7, 8]. It has been shown that the accepting state complexity remains consistent across all operations within finite group automata structures [9]. They determined SNC and QC (syntactic and quotient complexity) of languages obtained by finite group automata, where G is cyclic. The corresponding ranges of other groups were also specified [10].

In the exploration of algebraic automata theory, various authors [11], [12] have delved into the concept of group machines, with

notable contributions by Holocombe. Masami Ito significantly broadened this field by introducing several types of automata, including strongly connected, cyclic, perfect, permutation, and bideterministic varieties. A bideterministic automaton, a subset of deterministic automata, has the unique property - its A^R (reversal automaton) is also deterministic. Bideterministic automaton is the unique minimal automaton among all automata, including nondeterministic ones, that accept the same language.
Tamm et al. [13] demonstrated this uniqueness and also provided a primary result. They showed that, under certain conditions, a minimal deterministic automaton accepting a language or the reversal of the minimal deterministic automaton for the reversed language is a minimal automaton description of the language. This paper delves into the correlations between finite group automata and group machines, thoroughly analyzing their respective characteristics. Furthermore, we introduce the concept of the group machine recognizer. We have demonstrated the distinctive properties of these two types of automata as mentioned earlier. Notably, we provide examples for each characteristic to facilitate a deeper understanding. And also, we define two structures of group machines when they take the states and inputs as group elements, where the group is abelian and non-abelian. The organization of this paper is as follows: Section 2 offers preliminaries and includes illustrative examples to aid in understanding the concepts. Section 3 examines the distintive features of finite group automata. Section 4 describes the configuration of the group machine. Section 5 presents the main results related to group machines. Lastly, Section 6 provides a brief summary of insights and suggests directions for future research.

2 Introductory concepts and initial findings

In this section we present the necessary definition and the results that are essential to prove the primary results.

Definition 1 *[12] A state machine, also known as a semiautomaton, $\mathcal{M} = (Q, \Sigma, \delta)$, where Q and Σ represent non-empty finite sets, and δ is a partial function. If the partial function δ is a function, the state machine $\mathcal{M} = (Q, \Sigma, \delta)$ is termed complete.*

Definition 2 *[12] A transformation semigroup (TS) is denoted as (Q, S), comprising a finite set Q, a finite semigroup S, and an action of S on Q. This action is represented by a partial function $\lambda : Q \times S \to Q$ satisfying two conditions:*

1. $\lambda(q, s \cdot s_1) = \lambda(\lambda(q, s), s_1)\ \forall\, q \in Q,\ s, s_1 \in S.$
2. $\lambda(q, s) = \lambda(q, s_1)\ \forall\ q \in Q$, *implying* $s = s_1$ *where* $s, s_1 \in S.$

Notation $\lambda\ (q,\ s) = q\ s = q \cdot s,\ q \in Q,\ s \in S.$

Note 1 *[12] Associated with any state machine $\mathcal{M} = (Q, \Sigma, \delta)$ - transformation semigroup $(Q, \mathbf{S}(\mathcal{M}))$, denoted by $\mathbf{TS}(\mathcal{M})$ - the transformation semigroup of $\mathcal{M}$.*
Let $\mathcal{A} = (Q, S)$ be a transformation semigroup ($\mathbf{TS}$). The state machine $\mathcal{M} = (Q, S, \delta)$, in which $\delta(q, s) = qs\ \forall\ q \in Q$ and $s \in S$.
The semigroup $\mathbf{S}(\mathcal{M})$ of a state machine $\mathcal{M}$ is indeed a monoid, and $\mathbf{TS}(\mathcal{M})$ is a transform monoid.
For a given state machine $\mathcal{M} = (Q, S, \delta)$, we may define the transformation monoid of $\mathcal{M}$, denoted $\mathbf{TM}(\mathcal{M})$, as $(Q, \mathbf{M}(\mathcal{M}))$.

Definition 3 *[12] The transformation group (G, G) is defined where the group G acts on itself via right multiplication, represented as $g'g = g' \cdot g$ for $g' \in G$ and $g \in G$. This transformation group is denoted as $\mathscr{G}$.*

Definition 4 *[12] For any finite group G, we create a transformation semigroup $\mathscr{G} = (G, G)$ and a corresponding state machine $\mathbf{SM}(\mathscr{G}) = (G, G, F)$, where $g_1 F_g = g_1 g$ for $g_1, g \in G$. This state machine is referred to as the state machine associated with G.*

Definition 5 *[2] A finite group semiautomaton $\mathfrak{FGSA} = (Q,\ G,\ \delta)$, where*

- *Q represents the finite set of states.*
- *G represents the finite group of input symbols.*
- *$\delta : Q \times G \to Q$ represents the transition function meets $\delta\ (q,\ g\, h) = \delta\ (\delta\ (q,\ g),\ h)\ \forall\ q \in Q,\ g,\ h \in G.$*

Definition 6 *[11] A semiautomaton $\mathcal{M} = (Q, \Sigma, \delta)$ is said to be strongly connected if for any $s, t \in Q$, $\exists$ words $u, v \in \Sigma^*$ such that $\delta(s, u) = t$ and $\delta(t, v) = s$.*

Definition 7 *[11] A semiautomaton $\mathcal{M} = (Q, \Sigma, \delta)$ is said to be cyclic if the following conditions are satisfied:*

1. *There exists $s_0 \in Q$ which is called a generator of $\mathcal{M}$.*
2. *For any $s \in Q$, there exists $u \in \Sigma^*$ such that $\delta(s_0, u) = s$.*

Definition 8 *[11] A semiautomaton $\mathcal{M} = (Q, \Sigma, \delta)$ is considered commutative if $\delta(s, xy) = \delta(s, yx)$ holds for every $s \in Q$ and any $x, y \in \Sigma^*$. If, in addition, $\mathcal{M}$ is strongly connected, it is termed perfect.*

Definition 9 *[11] A semiautomaton $\mathcal{M} = (Q, \Sigma, \delta)$ is classified as a permutation automaton if $\delta(s, a)$ represents a permutation of Q for every $a \in \Sigma$.*

Definition 10 *[12] Let $\mathcal{M} = (Q, \Sigma, F)$ be a state machine. Let $i \in Q$ be a fixed state called the initial state and suppose that $T \subseteq Q$ represents the set of states called terminal states. The collection $\mathfrak{M} = (\mathcal{M}, i, T)$ is called an automaton or a recognizer.*

Definition 11 *[13] An automaton A is deemed deterministic if it possesses a singular initial state and, for each state $q \in Q$ and each symbol $a \in \Sigma$, the transition function $\delta(q, a)$ corresponds to precisely one state. If these conditions are not satisfied, the automaton is considered nondeterministic.*

Definition 12 *[13] The reversal of an automaton A, denoted by $A^R = (Q, \Sigma, \delta^R, F, I)$, has the transition function δ^R as $\delta^R(p, a) = \{q \mid p \in \delta(q, a)\}$ for all $p \in Q$ and $a \in \Sigma$. Here, I represents the initial state and F the final state. An automaton A is considered bideterministic if both A and A^R are deterministic.*

Example 1 *Consider the semiautomaton $\mathcal{M}_1 = (Q, \Sigma, \delta)$ where:*

- *$Q = \{s_1, s_2, s_3\}$ represents the set of states,*
- *$\Sigma = \{a, b\}$ represents the alphabet,*
- *δ represents the transition function as:*

$$\delta(s_1, a) = s_2$$
$$\delta(s_2, b) = s_3$$
$$\delta(s_3, a) = s_1$$

In this automaton, for any $s, t \in Q$, $\exists$ words $u, v \in \Sigma^* \ni \delta(s, u) = t$ and $\delta(t, v) = s$:

- *For s_1 and s_2, let $u = a$ and $v = b$. From the representation $\delta(s_1, a) = s_2$ and $\delta(s_2, b) = s_3$.*
- *For s_2 and s_3, let $u = b$ and $v = a$. From the representation $\delta(s_2, b) = s_3$ and $\delta(s_3, a) = s_1$.*
- *For s_3 and s_1, let $u = a$ and $v = b$. From the representation $\delta(s_3, a) = s_1$ and $\delta(s_1, a) = s_2$.*

Hence, the automaton $\mathcal{M}_1$ is strongly connected. Figure 1 is the automaton diagram of $\mathcal{M}_1$.

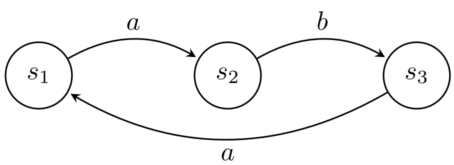

Figure 1. Automaton $\mathcal{M}_1$

Example 2 *Consider the cyclic semiautomaton $\mathcal{M}_2 = (Q, \Sigma, \delta)$ where:*

- *$Q = \{s_0, s_1, s_2\}$ represent thes set of states,*
- *$\Sigma = \{a, b, c\}$ represent thes alphabet,*
- *δ represents the transition function as:*

$$\delta(s_0, b) = s_0$$
$$\delta(s_0, a) = s_1$$
$$\delta(s_0, c) = s_2$$
$$\delta(s_1, b) = s_2$$
$$\delta(s_2, a) = s_3$$

- *s_0 is the generator of $\mathcal{M}_2$.*

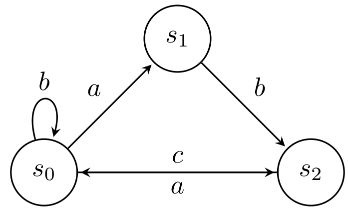

Figure 2. Automaton $\mathcal{M}_2$

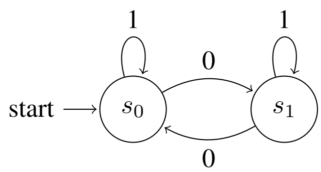

Figure 3. Automaton $\mathcal{M}_3$

In this automaton, s_0 is a generator, and for any state $s_i \in Q$, $\exists$ a word $u \in \Sigma^* \ni \delta(s_0, u) = s_i$. Hence, $\mathcal{M}_2$ is a cyclic automaton. Here Figure 2 exhibits the automaton diagram of $\mathcal{M}_2$.

Example 3 *Consider semiautomaton $\mathcal{M}_3 = (Q, \Sigma, \delta)$ where $Q = \{s_0, s_1\}$, $\Sigma = \{0, 1\}$, δ as:*

$$\delta(s_0, 0) = s_1$$
$$\delta(s_0, 1) = s_0$$
$$\delta(s_1, 0) = s_0$$
$$\delta(s_1, 1) = s_1$$

This automaton $\mathcal{M}_3$ is commutative since $\delta(s, xy) = \delta(s, yx)$ for all $s \in Q$ and $x, y \in \Sigma^*$. Furthermore, $\mathcal{M}$ is strongly connected, as a path exists between every pair of states. Therefore, $\mathcal{M}_3$ is a perfect automaton. Figure 3 is the automaton diagram of $\mathcal{M}_3$.

Example 4 *Consider semiautomaton $\mathcal{M}_4 = (Q, \Sigma, \delta)$ where $Q = \{s_0, s_1, s_2\}$, $\Sigma = \{0, 1\}$, and δ is defined as follows:*

$$\delta(s_0, 0) = s_1$$
$$\delta(s_0, 1) = s_2$$
$$\delta(s_1, 0) = s_0$$
$$\delta(s_1, 1) = s_2$$
$$\delta(s_2, 0) = s_1$$
$$\delta(s_2, 1) = s_0$$

This automaton $\mathcal{M}_4$ is a permutation automaton since $\delta(s, a)$ represents a permutation of Q for every $a \in \Sigma$. Each tran-

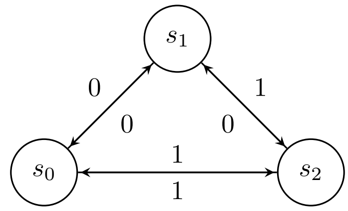

Figure 4. Automaton $\mathcal{M}_4$

sition function $\delta(s, a)$ corresponds to a distinct permutation of the states in Q, satisfying the definition of a permutation automaton. Here Figure 4 exhibits the automaton diagram of $\mathcal{M}_4$.

3 Distinctive Features of Finite Group Automata

The discussion based on the characteristics of finite group automata, analyzes their properties and provides counter-examples for automata that do not satisfy respective characteristics. Here $\mathfrak{FGSA} = (Q, G, \delta)$ is a finite group semiautomaton [3].

Definition 13 *[3] Type-1 automaton : Q - set with only one element, G - group of order n, and δ - transition function. As $\delta\ (a,\ g) = a$, $\forall\ a \in Q$, $g \in G$ generates $\mathfrak{GA} = (Q,\ G,\ \delta)$ a finite group automaton.*

Definition 14 *[3] Type-2 automaton : Q - set of two elements, G represents group of order n, having transition function δ provides $\mathfrak{GA} = (Q,\ G,\ \delta)$ is finite group automata as $\delta\ (a,\ g) = b$, $\delta\ (b,\ g) = b$ for $a, b \in Q$, $g \in G$. On the other, $\delta\ (a,\ g) = a$, $\delta\ (b,\ g) = a$ for $a, b \in Q$, $g \in G$.*

Definition 15 *[3] Type-3 automaton : Q - set of n- states, G - group of order n and δ - transition function. As $\delta\ (a_i,\ g_i) = a_i + 1$, $\delta\ (a_i,\ e) = a_i$, $\delta\ (a_i,\ g_j) = a_{i+j}$, in which $i,\ j \in \mathbb{N}$ generates $\mathfrak{GA} = (Q,\ G,\ \delta)$ a finite group automata.*

Theorem 1 *Structure/Type 1 and Type 3 of finite group semiautomaton are strongly connected.*

Proof 1 *To prove that $\mathfrak{FGSA}$ is strongly connected, i.e., for all $a_i, a_j \in Q$, there exist $g_k, g_l \in G$ such that $\delta(a_i, g_k) = a_j$ and $\delta(a_j, g_l) = a_i$.*

Type-1 Semiautomaton: By the definition and structure of a finite group semiautomaton of Type-1 [3]: Let $Q = \{a\}$ and G be a group with n elements. Define $\delta : Q \times G \to Q$ such that $\delta(a, g) = a$. Since the semiautomaton has only one state, it is trivially strongly connected.

Type-3 Semiautomaton: By the definition and structure of a finite group semiautomaton of Type-3 [3]: Let $Q = \{a_1, a_2, \ldots, a_n\}$ be a set of n states and G represents group of order n, having the transition function $\delta(a_i, e) = a_i$ (where e - identity element of G) and $\delta(a_i, g_j) = a_{i+j}$. Also, define $\delta(a_i, g_k g_l) = \delta(a_i, g_{k+l})$ and $\delta(a_i, g_k) = a_{i+k}$.
Now, choose $k = j - i$ for all $a_i, a_j \in Q$. We have:

$$\begin{aligned}\delta(a_i, g_k) &= \delta(a_i, g_{j-i}) \\ &= \delta(a_i, g_j g_{-i}) \\ &= \delta(\delta(a_i, g_j), g_{-i}) \\ &= \delta(a_{i+j}, g_{-i}) \\ &= a_{i+j-i} \\ &= a_j\end{aligned}$$

Similarly, choose $l = i - j$ for all $a_i, a_j \in Q$. We have:

$$\begin{aligned}\delta(a_j, g_l) &= \delta(a_j, g_{i-j}) \\ &= \delta(a_j, g_i g_{-j}) \\ &= \delta(\delta(a_j, g_i), g_{-j}) \\ &= \delta(a_{j+i}, g_{-j}) \\ &= a_{j+i-j} \\ &= a_i\end{aligned}$$

Thus, for any pair of states a_i and a_j, we can find group elements g_k and g_l such that $\delta(a_i, g_k) = a_j$ and $\delta(a_j, g_l) = a_i$, proving that the semiautomaton is strongly connected.
Hence, the proof is complete.

Theorem 2 *Type-1 and Type-3 finite group semiautomata are cyclic automata.*

Proof 2 *Let $\mathfrak{FGSA} = (Q, G, \delta)$ be a finite group semiautomaton. To prove that $\mathfrak{FGSA}$ is cyclic, we need to show that $\exists$ a state $q_0 \in Q$, called a generator of $\mathfrak{FGSA}$, $\ni$ for any state $q \in Q$, $\exists$ an element $g \in G$ with $\delta(q_0, g) = q$.*

Type-1 Automaton: By the definition of a Type-1 automaton [3], let $Q = \{a\}$ and G be a group. As $\delta(a, g) = a\ \forall\ g \in G$. Since the automaton has only one state, it trivially satisfies the definition of a cyclic automaton, as a is the generator.

Type-3 Automaton: In a Type-3 automaton [3], let $Q = \{a_1, a_2, \ldots, a_n\}$ be a set of n states, and G represents group of order n. As

$$\delta(a_i, e) = a_i, \qquad \delta(a_i, g_j) = a_{i+j}$$

in which $i, j \in \{1, 2, \ldots, n\}$ and arithmetic is taken modulo n.

Fixing any state a_i and varying j from 1 to n, we can reach every other state in Q. Thus, for any $q \in Q$, there exists a group element $g_j \in G$ such that $\delta(a_i, g_j) = q$, which satisfies the cyclic automaton definition. Since this holds for any a_i, every state in Q can be considered a generator.
Therefore, both Type-1 and Type-3 semiautomata are cyclic.

Corollary 1 *A Type-2 finite group semiautomaton is neither strongly connected nor cyclic.*

Proof 3 *By the definition of a Type-2 semiautomaton [3], δ will be either $\delta(a, g) = b$, $\delta(b, g) = b$, in which $a, b \in Q$, $g \in G$, or $\delta(a, g) = a$, $\delta(b, g) = a$, for some $a, b \in Q$, $g \in G$.*

In both cases, $\nexists$ any group element $g_i \in G$ such that $\delta(q_i, g_i) = a$ or $\delta(q_i, g_i) = b$ for all $q_i \in Q$. Therefore, the automaton is not strongly connected, as not all states are reachable from every other state.

Furthermore, the conditions for a cyclic automaton are not satisfied in a Type-2 semiautomaton. Specifically, $\nexists$ a group element $g \in G \ni \delta(a, g) = a$ or $\delta(a, g) = b\ \forall\ a, b \in Q$.

Hence, Type-2 becomes not cyclic.

Thus, Type-2 finite group semiautomata are neither strongly connected nor cyclic.

Theorem 3 *If G is abelian, then the finite group semiautomaton is commutative.*

Proof 4 *We need to prove that the finite group semiautomaton $\mathfrak{FGSA} = (Q, G, \delta)$ is commutative. This means that for every state $s \in Q$ and for any $x, y \in G$, the equality $\delta(s, xy) = \delta(s, yx)$ holds.*

Since G is abelian, $xy = yx\ \forall x, y \in G$. Therefore, for any state $s \in Q$ and $\forall x, y \in G$, it follows:

$$\delta(s, xy) = \delta(s, yx),$$

which satisfies the commutativity condition for the semiautomaton.

Thus, $\mathfrak{FGSA}$ is commutative when G is abelian.

Theorem 4 *If G is abelian, then Type-1 and Type-3 finite group semiautomata are perfect.*

Proof 5 *By the previous theorem, we know that if G is abelian, then the finite group semiautomaton is commutative.*

To prove that $\mathfrak{FGSA} = (Q, G, \delta)$ is perfect, we need to verify that it is both commutative and strongly connected, as required by the definition of a perfect semiautomaton.

Since we have already established that the semiautomaton is commutative, we now need to show that it is strongly connected. By Theorem 1, we know that both Type-1 and Type-3 finite group semiautomata are strongly connected.

Therefore, since the semiautomaton is both commutative and strongly connected, we can conclude that $\mathfrak{FGSA}$ is perfect for both Type-1 and Type-3 semiautomata.

Theorem 5 *If the type-1 and 3 of finite group semiautomaton are a permutation automaton.*

Proof 6 *To prove $\mathfrak{FGSA}$ of type-1 and 3 are permutation automaton , i.e, $\delta(s, a)$ represents a permutation of S for every $a \in G$.*
Since the definition and structure of finite group semiautomata of type-1: it has only one state it is trivially permutation automaton.
In the definition and structure of finite group semiautomata of type-3, by fixing i and varying j we get all the states, and by proceeding this by varying i we get for every state a_i and for every g_j $\delta(a_i, g_j)$ represents a permutation of Q. So type-3 of finite group semiautomaton is a permutation automaton. Hence proved.

Theorem 6 *Type-1 and Type-3 finite group semiautomata are permutation automata.*

Proof 7 *We need to prove that the finite group semiautomaton $\mathfrak{FGSA}$ of Type-1 and Type-3 are permutation automata i.e., $\forall a \in G$, the transition function $\delta(s, a)$ represents a permutation of the set Q.*

Type-1 Automaton: By the definition and structure of Type-1 semiautomata, there is only one state in Q. Since a permutation on a set with a single element is trivial, the Type-1 automaton is trivially a permutation automaton.

Type-3 Automaton: For the Type-3 semiautomaton, consider the set of states $Q = \{a_1, a_2, \ldots, a_n\}$. By fixing i and varying j, the transition function $\delta(a_i, g_j)$ allows us to reach every state in Q. This implies that for every state $a_i \in Q$ and for every group element $g_j \in G$, the function $\delta(a_i, g_j)$ represents a permutation of the states in Q. Therefore, the Type-3 semiautomaton is a permutation automaton.

Thus, both Type-1 and Type-3 finite group semiautomata are permutation automata.

Theorem 7 *A deterministic finite group semiautomaton $\mathfrak{FGSA} = (Q, G, \delta)$ is cyclic with every state as a generator if and only if it is a permutation automaton.*

Proof 8 *Assume that $\mathfrak{FGSA} = (Q, G, \delta)$ is cyclic, meaning $\exists$ a state $q_0 \in Q$ called a generator of $\mathfrak{FGSA}$. For any state $q \in Q$, $\exists$ a group element $g \in G \ni \delta(q_0, g) = q$, and every state is a generator.*

We need to prove that $\mathfrak{FGSA}$ is a permutation automaton. Since every state is a generator, for any $q, q_0 \in Q$, there exists $g \in G$ such that $\delta(q_0, g) = q$. Furthermore, since the semiautomaton is deterministic, for every $g \in G$, the transition function $\delta(q_i, g)$ maps each state to exactly one state, implying that $\delta(q_i, g)$ represents a permutation on Q. Therefore, $\mathfrak{FGSA}$ is a permutation automaton.

Conversely, assume that $\mathfrak{FGSA} = (Q, G, \delta)$ is a permutation automaton. This means that for every $g \in G$, the transition function $\delta(a_i, g)$ represents a permutation on Q. As every state is accessible from any other state under the action of some element of G, this satisfies the cyclic condition. Hence, every state is a generator.

Thus, $\mathfrak{FGSA}$ is cyclic with every state as a generator if and only if it is a permutation automaton.

4 Structure of group machine

In this section, we provide the structure of the group machine. With the help of this structure, we can characterize the behaviour of the group machine. Here we first provide the definitions and followed by the structures figure 5 and figure 6.

Definition 1
The transformation group (G, G) is established where the group G operates on itself using its binary operation, denoted

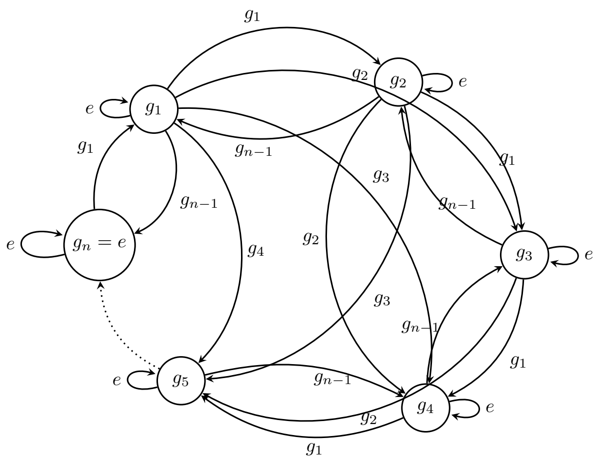

Figure 5. Stucture/Type-1 (Abelian)

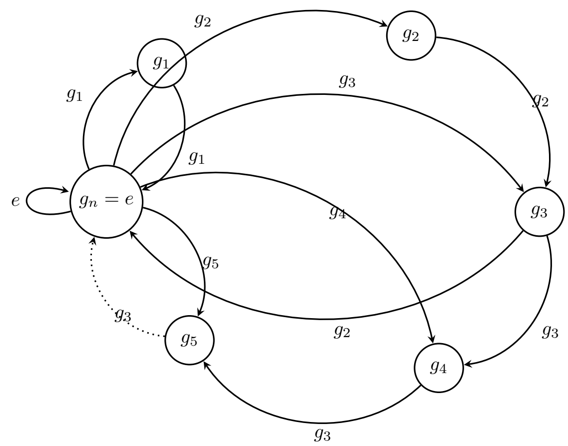

Figure 6. Stucture/Type-2 (Non-Abelian)

as $g^{'}g = g^{'} \circ g$ for $g^{'} \in G$ and $g \in G$. This transformation group is denoted as $\mathscr{TG}$. By defining the transformation group in this manner, we ensure that the first condition of the transformation group aligns with the associative property of the group.

Definition 2
For any finite group G, we establish a transformation group $\mathscr{TG} = (G, G)$ and a corresponding state machine $\mathbf{SM}(\mathscr{TG}) = (G, G, F)$, where $g_1 F_g = g_1 \circ g$ for $g_1, g \in G$, with $\circ$ representing the group's binary operation. This state machine is referred to as the state machine associated with G, also known as the group machine.

Definition 3
Let $\mathbf{SM}(\mathscr{TG})$ be a group machine, $i \in Q$ be a fixed state called the initial state and suppose that $T \subseteq Q$ represent the set of states called terminal states. The collection $\mathfrak{M} = (\mathbf{SM}(\mathscr{TG}), i, T)$ is called an group machine automaton or a group machine recognizer.

5 Group machine and group machine recognizer

In this section, we will discuss the characteristics of group machines and their connection to finite group automata. Here $\mathbf{SM}(\mathscr{TG}) = (G, G, F)$ denotes group machine.

Theorem 8 *All group machines are finite group semiautomata.*

Proof 9 *To prove that $\mathbf{SM}(\mathscr{TG})$ is a finite group semiautomaton, we will consider the definitions and properties given.*

A finite group semiautomaton $\mathfrak{FGSA} = (Q, G, \delta)$ consists of:

- *Q represents the finite set of states,*
- *G represents the finite group of input symbols,*
- *δ represents the transition function meeting $\delta(q, gh) = \delta(\delta(q, g), h)$ for all $q \in Q$ and $g, h \in G$.*

For the group machine $\mathbf{SM}(\mathscr{TG}) = (G, G, F)$, we have:

- *$Q = G$, in which G represents the finite group,*
- *The transition function $F(g, g_1) = g \cdot g_1$, where $\cdot$ denotes the group operation in G.*

To show that $\mathbf{SM}(\mathscr{TG})$ is a finite group semiautomaton, verify the properties of δ, where $\delta(g, g_1) = g \cdot g_1$:

1. *Associativity:*

$$\delta(q, g \cdot g_1) = \delta(\delta(q, g), g_1)$$

Since the group operation $\cdot$ is associative in G, we have $q \cdot (g \cdot g_1) = (q \cdot g) \cdot g_1$.

2. *Uniqueness:*

$$\delta(q, g) = \delta(q, g_1) \text{ implies } g = g_1$$

Because G is a group, right multiplication by a group element is a bijective function. Therefore, if $q \cdot g = q \cdot g_1$, then it follows that $g = g_1$.

3. *Identity Element:*

$$\delta(q, e) = q, \ \forall \, q \in G$$

Here, e - identity element of the group G, so this holds true.

Given these properties, we can now compare $\mathbf{SM}(\mathscr{TG})$ with $\mathfrak{FGSA}$. By setting $Q = G$ and $\delta = F$, we observe that every group machine is indeed a finite group semiautomaton. Specifically, the transition function $F(g, g_1) = g \cdot g_1$ satisfies the required conditions for δ in $\mathfrak{FGSA}$.

Therefore, a group machine $\mathbf{SM}(\mathscr{TG})$ with $Q = G$ and transition function defined by right multiplication ($\delta = F$) is a particular case of a finite group semiautomaton, $\mathfrak{FGSA}$. Thus, we conclude that every group machine is a finite group semiautomaton.

Proposition 1 *Not every finite group semiautomaton is a group machine.*

Proof 10 *Consider a Type-2 finite group semiautomaton* $\mathfrak{FGSA} = (Q, G, \delta)$*. In this case,* $\delta(q, s) = \delta(q, s_1)$ *holds* $\forall$ $q \in Q$ *and for distinct elements* s *and* s_1 *in* G *(i.e.,* $s \neq s_1$*).*

This situation demonstrates that the automaton does not satisfy the second condition of the definition of a group machine, which requires that each input symbol must produce a unique transition from a given state. Specifically, the existence of multiple inputs leading to the same state violates the requirement for distinct transitions from the same state.

Consequently, we conclude that this type of automaton cannot be classified as a group machine.

Theorem 9 *Every group machine is strongly connected.*

Proof 11 *To prove that* $\mathbf{SM}(\mathscr{TG})$ *is strongly connected, we need to show that for any* $g_i, g_j \in G$*, there exist elements* $g_k, g_l \in G$ *such that* $F(g_i, g_k) = g_j$ *and* $F(g_j, g_l) = g_i$*.*

Since $F(g_i, g_k) = g_i \circ g_k$ *and* G *is a group, we can select* $g_k = g_j \circ (g_i)^{-1}$*. Similarly, we can choose* $g_l = g_i \circ (g_j)^{-1}$*.*

By these choices of g_k *and* g_l*, we can demonstrate that* $\mathbf{SM}(\mathscr{TG})$ *is strongly connected:*

1. *For the transition from* g_i *to* g_j*:*

$$\begin{aligned} F(g_i, g_k) &= g_i F g_k \\ &= g_i \circ g_k \\ &= g_i \circ (g_i)^{-1} \circ g_j \\ &= g_j \end{aligned}$$

2. *For the transition from* g_j *to* g_i*:*

$$\begin{aligned} F(g_j, g_l) &= g_j F g_l \\ &= g_j \circ g_l \\ &= g_j \circ (g_j)^{-1} \circ g_i \\ &= g_i \end{aligned}$$

Since we have shown that for any g_i *and* g_j *in the group* G*, there exist elements* g_k *and* g_l *such that* $F(g_i, g_k) = g_j$ *and* $F(g_j, g_l) = g_i$*, it follows that* $\mathbf{SM}(\mathscr{TG})$ *is strongly connected.*

Theorem 10 *Every group machine is cyclic.*

Proof 12 *To prove* $\mathbf{SM}(\mathscr{TG})$ *is cyclic that is there exists* $g_j \in G$ *which is called a generator of* $\mathbf{SM}(\mathscr{TG})$*. For any* $g_i \in G$*,* $\exists\, g_l \in G \ni F(g_j, g_l) = g_i$*. Now choose* $g_l = (g_j)^{-1} \circ g_i$*.*

$$\begin{aligned} F(g_j, g_l) &= g_j F g_l \\ &= g_j \circ g_l \\ &= g_j \circ (g_j)^{-1} \circ g_i \\ &= g_i \end{aligned}$$

Moreover, every state is a generator since it works for every state.

Theorem 11 *If* G *is abelian then the group machine commutative. Moreover it is perfect.*

Proof 13 G *is abelian implies* $g_j \circ g_l = g_l \circ g_j$*.*
To prove $\mathbf{SM}(\mathscr{TG})$ *is commutative and perfect that is* $F(s, xy) = F(s, yx)$ *holds for every* $s \in G$ *and any* $x, y \in G$*. If, in addition,* A *is strongly connected, it is termed perfect.*

$$\begin{aligned} F(g_j, g_l) &= g_j F g_l \\ &= g_j \circ g_l \\ &= g_l \circ g_j \\ &= g_l F g_j \\ &= F(g_l, g_j) \end{aligned}$$

This implies $\mathbf{SM}(\mathscr{TG})$ *is commutative and perfect. Since group machine is strongly connected.*

Theorem 12 *Every group machine is a permutation automaton.*

Proof 14 *To prove it is a permutation automaton that is* $F(s, a)$ *represents a permutation of* G *for every* $a \in G$*. Since* $g_i, g_j \in G$ *it is possile to choose* $g = (g_j)^{-1} \circ g_i$*.*

$$\begin{aligned} F(g_j, g) &= g_j F g \\ &= g_j \circ g \\ &= g_i \end{aligned}$$

By varying j*,* $F(g_j, g)$ *represents a permutation of* G *for every* $g \in G$*. So every group machine is a permutation automaton.*

Theorem 13 *Every group machine is a bideterministic automaton.*

Proof 15 *Since the state and inputs of a group machine are elements of a group,* $\delta(g, g') = g \circ g'$*, where* $\circ$ *is the respective binary operation of* G*. It is evident that a group machine is a deterministic automaton.*

The reversal of a group machine $\mathbf{SM}(\mathscr{TG})$ *is denoted as* $\mathbf{SM}(\mathscr{TG})^R = (G, G, \delta^R, F, I)$ *where*

$$\delta^R(p, a) = \{q \mid p \in \delta(q, a), \forall p \in G,\, a \in G\},$$

I *- initial state and* F *- final state.*

Based on the structure of a group machine, there exists only one p *that satisfies* $p = \delta(q, a) = q \circ a$*. According to the definition of a group, for each* q*, we obtain a unique* p*, and for each* p*, we get a unique* q*. Therefore,*

$$\delta^R(p, a) = \{q\}$$

is also deterministic. This implies that a group machine is a bideterministic automaton.

6 Conclusions

Our research on finite group automata and group machines has established significant connections between these concepts, enhancing our understanding of their interplay and characteristics. By analyzing the properties of finite group automata and extending them through the concept of group machines, we have developed a comprehensive framework for understanding the behaviour of automata within the context of group theory. This research opens new avenues for further exploration and the application of automata in various domains.

REFERENCES

[1] Plotkin, Boris Isaakovich, L. Ja Greenglaz, and Abchase A. Gvaramija, “Algebraic structures in automata and databases theory,” World Scientific, vol. 13, 1992, pp. 001-277.

[2] Kelarev, A. V, “On incidence rings of group automata,” Bulletin of the Australian Mathematical Society, vol. 67, no. 3, pp. 407-411, 2003. DOI: https://doi.org/10.1017/S0004972700037217.

[3] G. Sathiyasorubini, R. Venkatesan, “Complexity Study of Language Operations Using Finite Group Automata,” Asia Pacific Journal of Mathematics, vol. 11, no. 6, pp. 01-20, 2024. DOI: 10.28924/APJM/11-6.

[4] Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman, “Introduction to automata theory, languages, and computation,” Acm Sigact News, 2nd ed, vol. 32, no. 1, 2002, pp. 60-65.

[5] Martin, John C, “Introduction to Languages and the Theory of Computation,” 4th ed, NY: McGraw-Hill, vol. 4, 1991, pp. 001-428.

[6] Geffert, Viliam, Pališínová, Dominika, Szabari, Alexander, “State complexity of binary coded regular languages,” Theoretical Computer Science, vol. 990, pp. 114399, 2024. DOI: https://doi.org/10.1016/j.tcs.2024.114399.

[7] Brzozowski, Janusz A., Marek Szykuła, and Yuli Ye, “Syntactic complexity of regular ideals,” Theory of Computing Systems, vol. 62, pp. 1175-1202, 2018. DOI: https://doi.org/10.1007/s00224-017-9803-8.

[8] Brzozowski, Janusz, Galina Jirásková, and Baiyu Li, “Quotient complexity of ideal languages,” Theoretical Computer Science, vol. 470, pp. 36-52, 2013. DOI: https://doi.org/10.1016/j.tcs.2012.10.055.

[9] Hospodár, Michal, and Markus Holzer, “The ranges of accepting state complexities of languages resulting from some operations,” International Journal of Foundations of Computer Science, vol. 31, no. 08, pp. 1159-1177, 2020. DOI: https://doi.org/10.1142/S0129054120420083.

[10] Joseph A. Gallian, “Contemporary abstract algebra,” 9th ed, Chapman and Hall/CRC, 2021, pp. 001-161.

[11] Ito, Masami, “Algebraic theory of automata and languages,” World Scientific, 2004, pp. 001-197.

[12] Holcombe, Mike, “Algebraic automata theory,” Cambridge University Press, 2004, pp. 001-226.

[13] Tamm, Hellis, and Esko Ukkonen, “Bideterministic automata and minimal representations of regular languages,” Theoretical Computer Science, vol. 328, no. 1-2, pp. 135-149, 2004. DOI: https://doi.org/10.1016/j.tcs.2004.07.010.