Generating finite cyclic and dihedral groups using sequential insertion systems with interactions

Wan Heng Fong, Nor Haniza Sarmin, Sherzod Turaev, and Ahmad Firdaus Yosman

Citation: AIP Conference Proceedings **1830**, 070005 (2017); doi: 10.1063/1.4980954
View online: http://dx.doi.org/10.1063/1.4980954
View Table of Contents: http://aip.scitation.org/toc/apc/1830/1
Published by the American Institute of Physics

Articles you may be interested in

On the energy of non-commuting graph of dihedral groups
AIP Conference Proceedings **1830**, 070011070011 (2017); 10.1063/1.4980960

Generating Finite Cyclic and Dihedral Groups using Sequential Insertion Systems with Interactions

Wan Heng Fong[1,a)], Nor Haniza Sarmin[1,b)], Sherzod Turaev[2,c)] and Ahmad Firdaus Yosman[1,d)]

[1] *Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia*
[2] *Department of Computer Science, Kulliyyah of Information & Communication Technology, International Islamic University Malaysia, 53100 Jalan Gombak, Selangor D.E., Malaysia*

[a)] Corresponding author: fwh@utm.my
[b)] nhs@utm.my
[c)] sherzod@iium.edu.my
[d)] firdausyosman@yahoo.com

Abstract. The operation of insertion has been studied extensively throughout the years for its impact in many areas of theoretical computer science such as DNA computing. First introduced as a generalization of the concatenation operation, many variants of insertion have been introduced, each with their own computational properties. In this paper, we introduce a new variant that enables the generation of some special types of groups called sequential insertion systems with interactions. We show that these new systems are able to generate all finite cyclic and dihedral groups.

INTRODUCTION

The operation of insertion is a generalization of the operation of concatenation. Insertion allows a word to be inserted in arbitrary positions in an axiom or an iterated/generated word, where before, in the case of concatenation, the addition of a word could only be done at the right extremity of an axiom word. Kari in [1] had introduced many variants of insertion, that include sequential, parallel, controlled, permuted, and scattered. From there, the concept of bonded insertion systems were introduced in [2, 3]. Not only that, by combining the operations of contextual insertion and contextual deletion, the insertion-deletion systems were introduced in [4], which was later shown to be of high computability [5, 6, 7, 8].

The combination of formal languages and group theory has been done before as seen in [9, 10, 11, 12], where it was shown that automata diagrams could be used to describe different types of groups, which include Abelian and permutation groups.

The interdisciplinary work between formal languages and group theory motivated the work in this paper, where we use the insertion operation to introduce a new system that will enable us to generate several types of groups called sequential insertion systems with interactions.

This paper is organized as follows: in Section 2 we provide the preliminaries necessary for the work in this paper. Next, we present our findings in Section 3. Lastly, we give our conclusion and some suggestions for future work in Section 4.

The 4th International Conference on Mathematical Sciences
AIP Conf. Proc. 1830, 070005-1–070005-6; doi: 10.1063/1.4980954
Published by AIP Publishing. 978-0-7354-1498-3/$30.00

PRELIMINARIES

In this section, we only recall some notations and definitions used in this paper as we assume that the reader is knowledgeable regarding the basic concepts in formal languages and group theory. For further details, we direct the reader to [13] and [14], respectively.

The cardinality of a set S is denoted by $|S|$. The inclusion of a set A in a set B is denoted by $A \subseteq B$ and the proper inclusion by $A \subset B$.

An *alphabet* is a finite nonempty set of symbols and is denoted by Σ, while Σ^* is the set of strings over the alphabet Σ. A *language* L over an alphabet Σ is a subset of Σ^*. The empty word is denoted by λ. For a word w, the length is denoted by $|w|$.

An *insertion system*, introduced in [1], is a triple $\gamma = (\Sigma, A, I)$, where Σ is the alphabet, $A \subseteq \Sigma^*$ is a finite set of axioms and $I \subseteq \Sigma^*$ is a finite set of insertion rules. The derivation relation $\Rightarrow_\gamma$ of an insertion system $\gamma = (\Sigma, A, I)$ is defined as follows: let $\alpha, \beta \in \Sigma^*$. Then $\alpha \Rightarrow_\gamma \beta$ if and only if $\alpha = \alpha_1\alpha_2$ for $\alpha_1, \alpha_2 \in \Sigma^*$ and there is an $\alpha' \in I$ such that $\beta = \alpha_1\alpha'\alpha_2$. The reflexive and transitive closure of $\Rightarrow_\gamma$ is denoted by $\Rightarrow_\gamma^*$. Should there be no danger of confusion, we write $\Rightarrow$ and $\Rightarrow^*$ instead of $\Rightarrow_\gamma$ and $\Rightarrow_\gamma^*$, respectively.

The language generated by an insertion system $\gamma = (\Sigma, A, I)$ is defined as

$$L(\gamma) = \{\beta \mid \text{there exists an axiom } \alpha \in A \text{ such that } \alpha \Rightarrow_\gamma^* \beta\}.$$

A *sequential insertion system* is an insertion system that works sequentially, that is, only one insertion is done at one position at every derivation step.

The following definitions are as found in [14].

A *group* $\langle G, * \rangle$ is a set G, closed under a binary operation $*$ such that the following axioms are satisfied:

1. For all $a, b, c \in G$ we have
$$(a * b) * c = a * (b * c).$$
2. There is a unique element $e \in G$ such that for all $x \in G$,
$$x * e = e * x = x.$$
3. Corresponding to each $a \in G$, there is an element $a^{-1} \in G$ such that
$$a * a^{-1} = a^{-1} * a = e.$$

The order of a group G is denoted by $|G|$ and is equal to the number of elements in G.

A group $G = \langle a \rangle$ is said to be a *cyclic group* if for an element $a \in G$,

$$G = \{a^n \mid n \in \mathbb{Z}\},$$

where a is the generator of G. The order of the group G is the order of the cyclic group generated by its generator i.e. $|G| = |\langle a \rangle|$.

A *multiplicative group of integers modulo n*, denoted by $U(n)$ is the set of non-negative integers less than n and relatively prime to n under the operation multiplication modulo n.

A *group of integers modulo n*, denoted by $\mathbb{Z}_n$ is the set of non-negative integers less than n under the operation addition modulo n.

A *permutation of a set A* is a function $\phi : A \to A$ that is both one to one and onto. For a finite set $A = \{1, 2, \cdots, n\}$, the group of all permutations of A is the *symmetric group* on n letters, and is denoted by S_n with an order of $n!$.

The *nth dihedral group* D_n is the group of symmetries of the regular n-gon, where the order of D_n is $2n$.

MAIN RESULTS

Sequential Insertion Systems with Interactions

In this section we introduce a new variant of insertion systems called sequential insertion systems with interactions. This system will enable us to generate languages that mimic groups with specific binary operations by defining interactions between two consecutive symbols in a derivation step. The idea behind using the concept of interactions comes from the concept of L-systems with interactions (IL-systems for short) [15].

The formal definition of this new system is as follows.

Definition 1 *A sequential insertion system with interaction (∗SINS-system for short) is a quadruple* $\zeta = (\Sigma, A, I, *)$*, where* Σ *is an alphabet,* $A \subseteq \Sigma^*$ *is a finite set of axioms,* $I \subseteq \Sigma^*$ *is a finite set of insertion rules and* $*$ *is a binary operation, such that for all* $\beta \in \Sigma^*$, $\beta = \alpha_1\alpha_2 = \alpha_1 * \alpha_2$*, where* $\alpha_1, \alpha_2 \in \Sigma^*$.

The derivation relation $\Rightarrow_\zeta$ *is defined as follows: for* $\alpha, \beta \in \Sigma^*$, $\alpha \Rightarrow_\zeta \beta$ *if and only if there exists an* $\alpha' \in I$ *such that* $\alpha * \alpha' = \beta$.

The reflexive and transitive closure of $\Rightarrow_\zeta$ *is denoted by* $\Rightarrow_\zeta^*$*. Should there be no danger of confusion, we write* $\Rightarrow$ *and* $\Rightarrow^*$ *instead of* $\Rightarrow_\zeta$ *and* $\Rightarrow_\zeta^*$*, respectively.*

The language generated by a ∗SINS-system $\zeta = (\Sigma, A, I, *)$ *is defined as*

$$L(\zeta) = \{\beta \mid \text{there exists an axiom } \alpha \in A \text{ such that } \alpha \Rightarrow_\zeta^* \beta\}.$$

The following examples demonstrate how sequential insertion systems with interactions work.

Example 1 *Let* $\zeta_1 = (\{0, 1, 2, 3\}, \{0\}, \{1\}, \text{addition modulo } 4)$ *be a ∗SINS-system. The derivation steps of* ζ_1 *are as follows:*

$$0 \Rightarrow 0 * 1 = 1 \Rightarrow 1 * 1 = 2 \Rightarrow 2 * 1 = 3 \Rightarrow 3 * 1 = 0.$$

Hence, we obtain the language generated by ζ_1,

$$L(\zeta_1) = \{0, 1, 2, 3\} = \mathbb{Z}_4.$$

Example 2 *Let* $\zeta_2 = (\{1, 2, 3, 4\}, \{1\}, \{2\}, \text{multiplication modulo } 5)$ *be a ∗SINS-system. The derivation steps of* ζ_2 *are as follows:*

$$1 \Rightarrow 1 * 2 = 2 \Rightarrow 2 * 2 = 4 \Rightarrow 4 * 2 = 3 \Rightarrow 3 * 2 = 1.$$

Hence, we obtain the language generated by ζ_2,

$$L(\zeta_2) = \{1, 2, 3, 4\} = U(5).$$

Example 3 *Let* $\zeta_3 = (S_3, \{(1)\}, \{(123), (12)\}, \text{composition})$ *be a ∗SINS-system. The derivation steps of* ζ_3 *are as follows:*

$$(1) \Rightarrow (1) * (123) = (123) \Rightarrow (123) * (123) = (132) \Rightarrow (132) * (123) = (1),$$

$$(1) \Rightarrow (1) * (12) = (12),$$

$$(1) \Rightarrow (1) * (123) = (123) \Rightarrow (123) * (12) = (13),$$

$$(1) \Rightarrow (1) * (123) = (123) \Rightarrow (123) * (123) = (132) \Rightarrow (132) * (12) = (23).$$

Hence, we obtain the language generated by ζ_3,

$$L(\zeta_3) = \{(1), (123), (132), (12), (13), (23)\} = S_3.$$

Generating Finite Cyclic and Dihedral Groups

We have seen in Examples 1–3 how a $*$SINS-system can generate the cyclic groups $\mathbb{Z}_4$ or $U(5)$ or the symmetric group, S_3, which is also known as the dihedral group D_3 of order 6. Those results provided the inspiration towards obtaining the results presented in this section.

In this section, we show that $*$SINS-systems can generate all finite cyclic groups and dihedral groups. Firstly, we show the former.

Theorem 1 *For every finite cyclic group G, there exists a $*$SINS-system $\zeta = (\Sigma, A, I, *)$ such that $L(\zeta) = G$.*

Proof. Let G be a finite cyclic group which is closed under a binary operation $*$. By definition, G contains the identity element e and for all g in G, there exists an element g^{-1} the inverse of g such that $gg^{-1} = e$. Also, since G is cyclic, there exists an element a in G such that G is generated by a.

Now, we define a $*$SINS-system $\zeta_G = (\Sigma, A, I, *)$ to generate a language that is equal to a finite cyclic group G. We do this by constructing the system as follows:

Let

Σ be the set of all the elements in G i.e. $\Sigma = G$,

$A \subseteq \Sigma^*$ be the set containing the identity element e of G,

$I \subseteq \Sigma^*$ be the set containing the generator a of G and its inverse a^{-1},

$*$ be the binary operation of G.

Then, the language generated by ζ_G is

$$L(\zeta_G) = \{a^n, a^{-n} \mid n \in \mathbb{Z}\} = G.$$

□

The following example provides a clearer simulation of our proof.

Example 4 *Let $\mathbb{Z}_6$ be the set of non-negative integers less than 6. By definition, $\mathbb{Z}_6$ is a finite cyclic group under addition modulo 6, generated by 1 and 5 and has the identity element 0. We may construct the $*$SINS-system to generate $\mathbb{Z}_6$ as follows:*

Let $\zeta_{\mathbb{Z}_6} = (\{0, 1, 2, 3, 4, 5\}, \{0\}, \{1, 5\}, \text{addition modulo } 6)$ *be a $*$SINS-system. The derivation steps of $\zeta_{\mathbb{Z}_6}$ are as follows:*

$$0 \Rightarrow 0 * 1 = 1 \Rightarrow 1 * 1 = 2 \Rightarrow 2 * 1 = 3 \Rightarrow 3 * 1 = 4 \Rightarrow 4 * 1 = 5 \Rightarrow 5 * 1 = 0,$$

$$0 \Rightarrow 0 * 5 = 5 \Rightarrow 5 * 5 = 4 \Rightarrow 4 * 5 = 3 \Rightarrow 3 * 5 = 2 \Rightarrow 2 * 5 = 1 \Rightarrow 1 * 5 = 0.$$

Hence, we obtain the language generated by $\zeta_{\mathbb{Z}_6}$,

$$L(\zeta_{\mathbb{Z}_6}) = \{0, 1, 2, 3, 4, 5\} = \mathbb{Z}_6.$$

Note that the inclusion of both the generator and its inverse in the set of insertion rules is necessary to guarantee that all the elements in a group G are generated. However, in some cases it suffices to only include the generator of the group without its inverse, as shown in Examples 1, 2 and 4. Thus, we obtain at the following corollaries.

Corollary 1 *For all cyclic groups $\mathbb{Z}_n$, it suffices to only include the generator of $\mathbb{Z}_n$ in the set I of insertion rules of a $*$SINS-system $\zeta = (\Sigma, A, I, *)$ to obtain $L(\zeta) = \mathbb{Z}_n$.*

Corollary 2 *For all cyclic groups $U(n)$, it suffices to only include the generator of $U(n)$ in the set I of insertion rules of a $*$SINS-system $\zeta = (\Sigma, A, I, *)$ to obtain $L(\zeta) = U(n)$.*

Next, we show that $*$SINS-systems can generate all dihedral groups.

Theorem 2 *For every dihedral group D_n, there exists a $*$SINS-system $\zeta = (\Sigma, A, I, *)$ such that $L(\zeta) = D_n$.*

Proof. Let D_n be a dihedral group which is closed under its binary operation $*$. By definition, D_n contains n number of rotations and n number of reflections, which includes the identity element e.

Now, we define a $*$SINS-system $\zeta_{D_n} = (\Sigma, A, I, *)$ to generate a language that is equal to a dihedral group D_n. We do this by constructing the system as follows:

Let

Σ be the set of all the elements in D_n i.e. $\Sigma = D_n$,

$A \subseteq \Sigma^*$ be the set containing the identity element e of D_n,

$I \subseteq \Sigma^*$ be the set containing a rotation ρ which is not the identity and a reflection μ,

$*$ be the binary operation of D_n, which is composition.

Then, the language generated by ζ_{D_n} is

$$L(\zeta_{D_n}) = \{\rho^n, \mu^n, \rho^m \mu^n \mid m, n \in \mathbb{Z}, m \leq n\} = D_n.$$

□

For further clarification, we provide the following example.

Example 5 *Let D_4 be the dihedral group of order 8. By definition, D_4 contains 4 rotations and 4 reflections, with the identity element ρ_0. To generate D_4, we construct the $*$SINS-system $\zeta_{D_4} = (D_4, \{\rho_0\}, \{\rho_{90}, \mu_H\}, \text{composition})$. The derivation steps of ζ_{D_4} are as follows:*

$$\rho_0 \Rightarrow \rho_0 * \rho_{90} = \rho_{90} \Rightarrow \rho_{90} * \rho_{90} = \rho_{180} \Rightarrow \rho_{180} * \rho_{90} = \rho_{270},$$

$$\rho_0 \Rightarrow \rho_0 * \mu_H = \mu_H,$$

$$\rho_0 \Rightarrow \rho_0 * \rho_{90} = \rho_{90} \Rightarrow \rho_{90} * \mu_H = \mu_D,$$

$$\rho_0 \Rightarrow \rho_0 * \rho_{90} = \rho_{90} \Rightarrow \rho_{90} * \rho_{90} = \rho_{180} \Rightarrow \rho_{180} * \mu_H = \mu_V,$$

$$\rho_0 \Rightarrow \rho_0 * \rho_{90} = \rho_{90} \Rightarrow \rho_{90} * \rho_{90} = \rho_{180} \Rightarrow \rho_{180} * \rho_{90} = \rho_{270} \Rightarrow \rho_{270} * \mu_H = \mu_{D'}.$$

Hence, we obtain the language generated by ζ_{D_4},

$$L(\zeta_{D_4}) = \{\rho_0, \rho_{90}, \rho_{180}, \rho_{270}, \mu_H, \mu_D, \mu_V, \mu_{D'}\} = D_4.$$

In both of the proofs in Propositions 1 and 2, we require the alphabet Σ to include all the elements in the desired group. Although it may seem trivial i.e. the inclusion of all the elements in the alphabet Σ will automatically generate a language equal to the desired group, this is in fact not the case. Instead, we require the alphabet Σ to include all the elements in the desired group so that the system is able to identify the elements in the input and sentential form and also that the system does not skip any desired element in the final output.

CONCLUSION

In this paper, a new variant of insertion systems was introduced, namely the sequential insertion systems with interactions ($*$SINS-system). We have simulated the generation of languages that are equal to some well-known groups, namely two cyclic groups of order four and the symmetric group of order six. Other than that, we have also shown that the $*$SINS-system is able to generate all finite cyclic and dihedral groups, as seen in Theorems 1 and 2.

In the future, we hope to extend the idea of generating groups using the $*$SINS-system to other types of groups and also subgroups. We also hope to determine the generative power of the $*$SINS-system with respect to the Chomsky hierarchy.

Our results have shown a significant relation between formal languages and group theory, which we hope will spur even more interdisciplinary research and collaborations between theoretical computer scientists and mathematicians. Not only that, due to its immense potential, we also foresee the application of $*$SINS-system in DNA computing.

ACKNOWLEDGMENTS

The first and second author would like to acknowledge MOHE and Research Management Centre (RMC) of Universiti Teknologi Malaysia (UTM) for the financial funding through Fundamental Research Grant Scheme (FRGS) Vote No. 4F590 and Research University Grant Vote No. 13H18. The third author would like to thank the International Islamic University Malaysia through MOHE for his financial funding of IIUM Research Initiative Grant Scheme RIGS16-368-0532. The fourth author would like to thank the Ministry of Higher Education Malaysia (MOHE) for his MyBrainSC scholarship.

REFERENCES

[1] L. Kari, "On Insertion and Deletion in Formal Languages", Ph.D. thesis, University of Turku (1991).

[2] W. H. Fong, M. Holzer, B. Truthe, S. Turaev, and A. F. Yosman, "On Bonded Sequential and Parallel Insertion Systems," in *Eighth Workshop on Non-Classical Models of Automata and Applications (NCMA)*, edited by H. B. et al., books@ocg.at, Österreichische Computer Gesellschaft, Austria, 2016, pp. 163–178.

[3] A. F. Yosman, M. Holzer, B. Truthe, W. H. Fong, and S. Turaev, "Two Variants of Bonded Parallel Insertion Systems and Their Generative Power," in *Proceedings of the 6th International Graduate Conference on Engineering, Science and Humanities (IGCESH) 2016*, Universiti Teknologi Malaysia, Johor Bahru, Malaysia, 2016, pp. 418–420.

[4] L. Kari, and G. Thierrin, Information and Computation **131**, 47–61 (1996).

[5] L. Kari, Gh. Păun, G. Thierrin, and Sh. Yu, "At the Crossroads of DNA Computing and Formal Languages: Characterizing RE using Insertion-Deletion Systems," in *Proceedings of 3rd DIMACS Workshop on DNA Based Computing*, 1999, pp. 318–333.

[6] A. Takahara, and T. Yokomori, Natural Computing **2**, 321–336 (2003).

[7] S. Verlan, Computer Science Journal of Moldova **18**, 210–245 (2010).

[8] A. Krassovitskiy, "Complexity and Modeling Power of Insertion-Deletion Systems", Ph.D. thesis, Universitat Rovira i Virgili (2011).

[9] Y. S. Gan, W. H. Fong, N. H. Sarmin, and S. Turaev, Malaysian Journal of Fundamental and Applied Sciences **8**, 24–30 (2012).

[10] Y. S. Gan, W. H. Fong, N. H. Sarmin, and S. Turaev, Malaysian Journal of Fundamental and Applied Sciences **9**, 35–40 (2013).

[11] W. H. Fong, Y. S. Gan, N. H. Sarmin, and S. Turaev, "Automata Diagram over Abelian Groups," in *AIP Conference Proceedings*, 2013.

[12] W. H. Fong, Y. S. Gan, N. H. Sarmin, and S. Turaev, "Automata for Subgroups," in *AIP Conference Proceedings*, 2014.

[13] G. Rozenberg, and A. Salomaa, *Handbook of Formal Languages*, Springer-Verlag, Berlin, 1997.

[14] J. B. Fraleigh, *A First Course in Abstract Algebra, 7th Edition*, Addison Wesley, 2003.

[15] G. Rozenberg, and A. Salomaa, *The Mathematical Theory of L-systems*, Academic Press, 1980.