




DNA origami words, graphical structures and their rewriting systems

James Garrett¹ · Nataša Jonoska¹ · Hwee Kim²  · Masahico Saito¹

Accepted: 18 November 2020 / Published online: 4 January 2021

© The Author(s), under exclusive licence to Springer Nature B.V. part of Springer Nature 2021

Abstract

We classify rectangular DNA origami structures according to their scaffold and staples organization by associating a graphical representation to each scaffold folding. Inspired by well studied Temperley–Lieb algebra, we identify basic modules that form the structures. The graphical description is obtained by ‘gluing’ basic modules one on top of the other. To each module we associate a symbol such that gluing of modules corresponds to concatenating the associated symbols. Every word corresponds to a graphical representation of a DNA origami structure. A set of rewriting rules defines equivalent words that correspond to the same graphical structure. We propose two different types of basic module structures and corresponding rewriting rules. For each type, we provide the number of all possible structures through the number of equivalence classes of words. We also give a polynomial time algorithm that computes the shortest word for each equivalence class.

Keywords DNA origami · Rewriting systems · Jones monoid

1 Introduction

Self-assembly is a process where smaller components (usually molecules) autonomously assemble to form a larger structure. This process is essential in building biomolecular structures and high order polymers (Whitesides and Boncheva 2002). Applications of self-assembly range from electric circuits at nano level (Bhuvana et al. 2009; Eichen et al. 1998) to smart drug delivery systems (Li et al. 2013; Verma and Hassan 2013). A well-known self-assembly variant is the DNA origami system

introduced by Rothemund (2006). In DNA origami, a single-stranded DNA plasmid, called the *scaffold*, outlines a shape, while short DNA strands, called *staples*, connect different parts of the scaffold, fixing the terminal rigid structure. The top of Fig. 1 shows a segment of schematic DNA origami where the scaffold is depicted by a black line while staples are represented by colored lines with arrows. Experimental results of several DNA origami shapes from Rothemund’s original paper (Rothemund 2006) are shown at the bottom of Fig. 1.

Theoretical approaches to analyze DNA origami have been mainly focused on efficient sequence design of staples as well as synthetic scaffolds that fold into the target shape (Rothemund 2005; Veneziano et al. 2016). However, the same outlined shape can be obtained by different scaffold and staple organizations. In this paper, we use graphical description to describe different scaffold/staple organization within the same origami shape. We identify unit building blocks (modules) for the graphical representations whose composition (one on top of another) through connecting the corresponding staple/scaffold strands builds up a larger structure. The unit blocks correspond to symbols in an alphabet, and concatenation of symbols correspond to composition of the modules. We observe that the unit structures within DNA origami resemble the diagram representation of the generators of the Jones monoid, a monoid

✉ Hwee Kim
hweekim@inu.ac.kr

James Garrett
jgarrett1@mail.usf.edu

Nataša Jonoska
jonoska@mail.usf.edu

Masahico Saito
saito@usf.edu

¹ Department of Mathematics and Statistics, University of South Florida, 4202 E. Fowler Ave., Tampa, FL 33620, USA

² Department of Computer Science and Engineering, Incheon National University, 119 Academy-Ro, Yeonsu-Gu, Incheon 22012, Republic of Korea

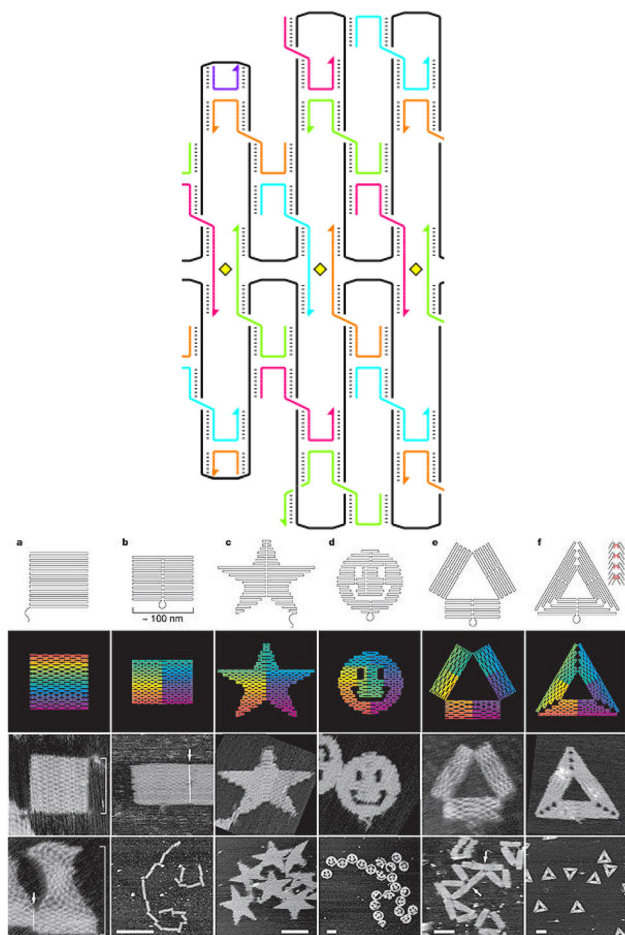


Fig. 1 (Top) A schematic representation of a DNA origami structure. The scaffold is a black line and staples are colored lines with arrows. (Bottom) Various shapes made by DNA origami. Both figures are from Rothemund (2006). (Color figure online)

variant of the well studied Temperley–Lieb algebras (Borisavljević et al. 2002; Jones 1983; Kauffman 2001). We assign symbols to the unit structures, and define rewriting rules that provide equivalence of words corresponding to their graphical representation equivalence. Winfree et al. (2001) proposed word representation of simple DNA tiles by modulization, but this is the first approach to describe general DNA origami structures by words. We propose two types of basic module structures and their corresponding rewriting rules. For each type, we provide the number of distinct equivalence classes of words, which corresponds to the possible DNA origami structures. We also compute the size of the shortest word within each class.

An extended abstract of this work had been published in the proceedings of the 18th International Conference on Unconventional Computation and Natural Computation (Garrett et al. 2019).

2 Preliminaries

An alphabet Σ is a non-empty finite set of symbols. A word $w = w_1w_2 \cdots w_n \in \Sigma^n$ is a finite sequence of n symbols over Σ , and $|w| = n$ denotes the size of the word. We use ϵ to denote the empty word. A *subword*, or a *factor*, of a word $w = w_1w_2 \cdots w_n$ is $w' = w_i \cdots w_j$ where $1 \leq i \leq j \leq n$. We use Σ^* to denote the set of all words over Σ . Concatenation of two words x and y is denoted by $x \cdot y$, or simply xy .

A word rewriting system (Σ, R) consists of an alphabet Σ and a set $R \subseteq \Sigma^* \times \Sigma^*$ of rewriting rules. In this paper, both Σ and R are finite. R generates an equivalence relation \hat{R} on Σ^* . An element (x, y) of R is called a rewriting rule, and is written as $x \leftrightarrow y$. In general, we rewrite uxv as uyv for $u, v \in \Sigma^*$ if $(x, y) \in R$, and denote such rewriting by $uxv \leftrightarrow uyv$. For a sequence of words $u = x_1 \leftrightarrow x_2 \leftrightarrow \cdots \leftrightarrow x_n = v$ in a rewriting system (Σ, R) , we write $u \sim v$. We consider \hat{R} and denote an *equivalence class* of a word w as $[w]$. A word $w_0 \in [w]$ is *irreducible* if $|w_0| \leq |w'|$ for all $w' \in [w]$. For an ordered alphabet Σ , we use the lexicographically first irreducible word \hat{w} of $[w]$ as the *representative word* of $[w]$. We consider the set of equivalence classes \mathcal{O} . The reader may refer to Book and Otto (1993) for more information about word rewriting systems.

The Temperley–Lieb algebra has been extensively studied in physics and knot theory (Kauffman 2001). A monoid version of Temperley–Lieb algebras, called the Jones monoid \mathcal{J}_n , has also been studied (Borisavljević et al. 2002; Jones 1983; Lau and FitzGerald 2006). The generators of \mathcal{J}_n are h_1, \dots, h_{n-1} and satisfy three classes of relations:

1. $h_i h_j h_i = h_i$ for $|i - j| = 1$
2. $h_i h_i = h_i$
3. $h_i h_j = h_j h_i$ for $|i - j| \geq 2$

The generators and relations can be represented graphically as in Fig. 2 (Lau and FitzGerald 2006). There are n endpoints at the top and the bottom of graphical representations of elements of \mathcal{J}_n . A generator h_i connects the i th and $i+1$ st top endpoints and the i th and $i+1$ st bottom endpoints, while other endpoints are connected by vertical lines. The generator h_3 in \mathcal{J}_5 is presented in Fig. 2a, connecting the top 3rd and 4th and the bottom 3rd and 4th points, respectively. Multiplication of two elements corresponds to concatenation of diagrams, placing the diagram of the first element on top of the second, and removing closed loops. The relations 1, 2 and 3 can also be expressed graphically as in Fig. 2b, d, respectively. Two elements in the Jones monoid are equal if their graphical representations are equivalent, that is, they have the same set of top-



Fig. 2 Graphical representation of the Jones monoid \mathcal{J}_5 . **a** The generator h_3 . **b** The relation $h_1h_2h_1 = h_1$. **c** The relation $h_1h_1 = h_1$. **d** The relation $h_1h_3 = h_3h_1$

bottom connecting segments after deleting internal loops. For any two words that have equivalent diagrams, one word can be rewritten to the other using the sequence of relations 1 to 3. In simplification of the DNA origami structure, we take a similar approach where we only take into account the endpoints of scaffolds and staples that are visible at the top and the bottom borderline of the whole structure. Thus, we use the Jones monoid as a base to construct DNA origami words and corresponding rewriting systems.

3 DNA origami words and rewriting systems

3.1 DNA origami words

We focus on rectangular DNA origami structures. They can be formed by a variety of scaffold-strand folds and connecting staples. We introduce an algebraic way to distinguish these different folds yielding the same overall shape. We use basic unit structures (modules) that build the shape and associate symbols (generators) to these basic modules. Based on graphical diagrams, and inspired by the Jones monoid diagrams, we define equivalence of two origami structures, and define corresponding rewriting rules that realize the equivalence in the graphical diagrams.

In this schematics of the DNA origami structure, we consider columns made of scaffolds, and staples that go along the scaffolds as follows: there are places where two adjacent scaffolds connect the two columns, and also places where two adjacent staples connect the two columns. In addition, because DNA is oriented, the scaffolds and staples have directions: adjacent scaffolds are anti-parallel, and a staple is anti-parallel to a scaffold it connects to. A graphical structure corresponding to DNA origami is thus presented with types of directed segments and the corresponding end-point connections. In addition, in order to define composition of structures when some parts of the structures are missing, we consider ‘virtual’ staples and scaffold. We use $s = i_t$ (i_b) to represent a point at the top (bottom) of the i th column. We assume that scaffolds at the i th column go upward if i is odd, and downward if i is even. The structure of width n is defined as follows: To the

set $E_n = \{i_t, i_b \mid 1 \leq i \leq n\}$ of points, we associate two partitions (P_c, Q_c) and (P_p, Q_p) where each set in the partitions consists of n points. The partition (P_c, Q_c) is associated to the scaffold strands and (P_p, Q_p) is associated to the staples. We define a bijection from P_c to Q_c which we describe as a set of ordered pairs $\{(s, t) \mid s, t, \in E_n\}$. Each pair (s, t) corresponds to a line segment that starts at point s and terminates at point t . This set of pairs is further partitioned to $(\mathcal{R}_c, \mathcal{V}_c)$. We call the line segments \mathcal{R}_c *real scaffolds* and \mathcal{V}_c *virtual scaffolds*. Similarly a bijection from P_p to Q_p represented by ordered pairs is partitioned to $(\mathcal{R}_p, \mathcal{V}_p)$ where \mathcal{R}_p is the set of *real staples* and \mathcal{V}_p is the set of *virtual staples*. Then a *graphical structure* is defined as a tuple $(\mathcal{R}_c, \mathcal{V}_c, \mathcal{R}_p, \mathcal{V}_p)$ of four sets of pairs or four sets of line segments. Note that such virtual scaffolds and staples are merely used as auxiliary tools for concise definition of concatenation of graphical structures, where the resulting graphical structure is defined by a set of connections of different types of scaffolds and staples—otherwise, we may leave such virtual scaffolds and staples empty, which might look more natural, but such emptiness leads to introduction of complicated ‘extensions’ of scaffolds and staples aside from connections in the definition of concatenation of graphical structures, and we choose to use the former.

For given width n , we define basic modules and corresponding generators $\Sigma_n = \{\alpha_i, \beta_i \mid 1 \leq i \leq n - 1\}$ as an alphabet for DNA origami words with the order $\alpha_1 < \dots < \alpha_{n-1} < \beta_1 < \dots < \beta_{n-1}$. We say that α_i is *complementary* to β_i , and vice versa. For each generator α_i, β_i , Table 1 shows the functions that describe their structures between the i th and the $i+1$ st columns. The four maps that describe α_i (resp. β_i) are called *units* for α_i (resp. β_i).

Table 1 Units for generators of odd i 's (maps are inverted for even i 's)

| | \mathcal{R}_c | \mathcal{R}_p |
|------------|------------------------------|------------------------------|
| α_i | $(i_b, i_t), (i+1_t, i+1_b)$ | $(i_t, i+1_t), (i+1_b, i_b)$ |
| β_i | $(i+1_t, i_t), (i_b, i+1_b)$ | $(i_t, i_b), (i+1_b, i+1_t)$ |

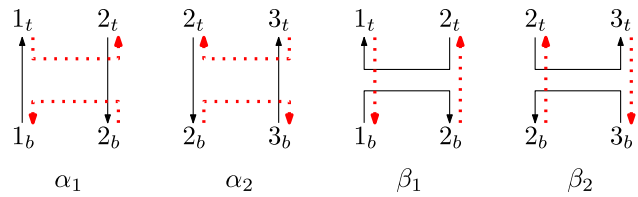


Fig. 3 Graphical representation of units of α_i and β_i ($i = 1, 2$). Scaffolds are represented by black lines and staples are represented by red dotted lines. For better visibility, staples are shifted right. (Color figure online)

The units of the generators α_i and β_i ($i = 1, 2$) are shown in Fig. 3.

In addition to the units for each generator, we must also define maps corresponding to the columns left of the i th and right of the $i+1$ st column. Unlike the Jones monoid, the choice of structure for these surrounding columns is not trivial. We define three possible systems for this surrounding structure through choices of real or virtual scaffolds and staples. The structure of each generator $\gamma_i \in \Sigma_n$ has a context $\mathcal{C}(\gamma_i)$ which consists of pairs (k_t, k_b) and their inverses where $k \notin \{i, i + 1\}$. The context $\mathcal{C}(\gamma_i)$ can have real or virtual pairs. Depending on the choice of virtual versus real context, there can be different structural descriptions. Table 2 defines three situations that can be used for three different descriptions of graphical structures $\mathcal{G}_{max(n)}$, $\mathcal{G}_{mid(n)}$, $\mathcal{G}_{min(n)}$, each representing a possible choice for the generator γ_i . We note that in $\mathcal{G}_{max(n)}$ the context $\mathcal{C}(\gamma_i)$ has both \mathcal{V}_c and \mathcal{V}_p empty, i.e. the whole context is real. In $\mathcal{G}_{mid(n)}$, the context $\mathcal{C}(\gamma_i)$ has $\mathcal{V}_c = \emptyset$, that is, the scaffold context is real but the staple context is virtual. In the case of $\mathcal{G}_{min(n)}$, the whole context $\mathcal{C}(\gamma_i)$ is virtual. The corresponding graphical structures of α_2 's in different \mathcal{G} 's are shown in Fig. 4.

Concatenation of words and the corresponding graphical structure is defined similarly as in the Jones monoid diagrams. Graphical structures that correspond to words in Σ_n^* are obtained by joining graphical structures of generators as explained below. The graphical structure corresponding

Table 2 Definition of three real and virtual scaffold and staple contexts for γ_i when i is odd. The maps are inverted for even i 's

| | $k \notin \{i, i + 1\}$ | \mathcal{R}_c | \mathcal{V}_c | \mathcal{R}_p | \mathcal{V}_p |
|------------------------|-------------------------|-----------------|-----------------|-----------------|-----------------|
| $\mathcal{G}_{max(n)}$ | Odd k | (k_b, k_t) | | (k_t, k_b) | |
| | Even k | (k_t, k_b) | | (k_b, k_t) | |
| $\mathcal{G}_{mid(n)}$ | Odd k | (k_b, k_t) | | | (k_t, k_b) |
| | Even k | (k_t, k_b) | | | (k_b, k_t) |
| $\mathcal{G}_{min(n)}$ | Odd k | | (k_b, k_t) | | (k_t, k_b) |
| | Even k | | (k_t, k_b) | | (k_b, k_t) |

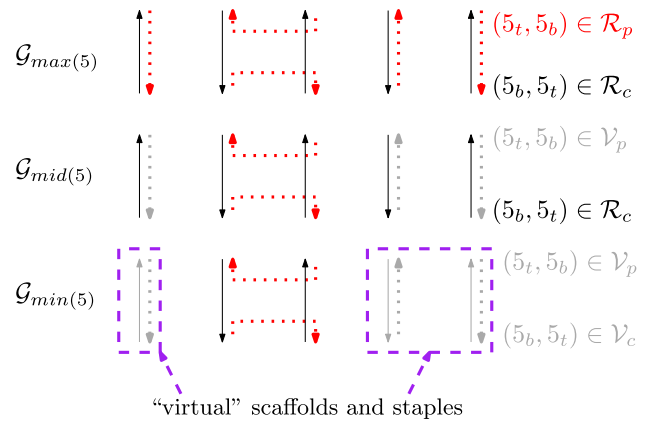


Fig. 4 Different graphical structures of α_2 's in $\mathcal{G}_{max(5)}$, $\mathcal{G}_{mid(5)}$ and $\mathcal{G}_{min(5)}$. Virtual scaffolds and staples are colored in gray

to concatenation of two words is obtained by placing the graphical structure of the first word on top of the graphical structure of the second and connect the vertical lines that meet. In the case of virtual staples or scaffolds the connection follows the rule: If a real scaffold (staple) meets a virtual scaffold (staple), then the virtual scaffold (staple) becomes real. This process simulates the real structure extending through the empty space represented by the virtual structure.

Figure 5 shows concatenation of $\alpha_1\beta_2$ and α_1 under $\mathcal{G}_{min(3)}$. Formally, the graphical structure of a word $w = w_1w_2$ is defined as follows: Suppose $G(w_1) = (\mathcal{R}_{c1}, \mathcal{V}_{c1}, \mathcal{R}_{p1}, \mathcal{V}_{p1})$ and $G(w_2) = (\mathcal{R}_{c2}, \mathcal{V}_{c2}, \mathcal{R}_{p2}, \mathcal{V}_{p2})$ are graphical structures of w_1 and w_2 , respectively. The graphical structure $G(w) = G(w_1w_2) = (\mathcal{R}_c, \mathcal{V}_c, \mathcal{R}_p, \mathcal{V}_p)$ is obtained with the following: The scaffold sets (\mathcal{R}_c and \mathcal{V}_c) are obtained as follows (the staples follow an equivalent procedure):

1. For all pairs in $\mathcal{R}_{c1} \cup \mathcal{V}_{c1}$, replace the subscript b by m .
2. For all pairs in $\mathcal{R}_{c2} \cup \mathcal{V}_{c2}$, replace the subscript t by m .

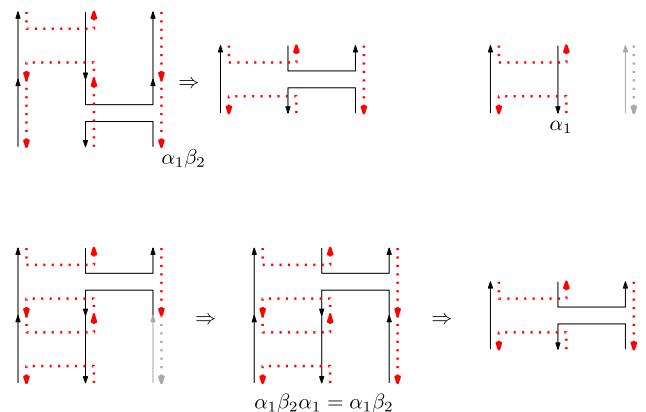


Fig. 5 Concatenation of $\alpha_1\beta_2$ and α_1 under $\mathcal{G}_{min(3)}$

3. Given set $\mathcal{R}_{c1} \cup \mathcal{V}_{c1} \cup \mathcal{R}_{c2} \cup \mathcal{V}_{c2}$, for each sequence $(q_0, q_1), (q_1, q_2), \dots, (q_{v-1}, q_v)$ of pairs where points q_0 and q_v have subscripts t or b ,
 - if any of these pairs is in $\mathcal{R}_{c1} \cup \mathcal{R}_{c2}$, add (q_0, q_v) to \mathcal{R}_c .
 - Otherwise, add (q_0, q_v) to \mathcal{V}_c .

We observe that the composition of graphical structures is associative, that is $G(w_1w_2w_3) = G((w_1w_2)w_3) = G(w_1(w_2w_3))$.

Figure 6 describes the concatenation process of scaffolds. We replace the subscripts for the bottom points of w_1 and the top points of w_2 by m , which denotes the middle points in the concatenation. Then, connect pairs of scaffolds that meet in the middle. We regard the connected scaffold to be virtual only if both original scaffolds were virtual (step 4 (c)). Finally, we delete all pairs of scaffolds whose endpoints are at the middle, this includes all internal loops. Based on Tables 1 and 2, we define the set $\mathcal{G} \in \{\mathcal{G}_{max}, \mathcal{G}_{mid}, \mathcal{G}_{min}\}$ as the set of all graphical structures that can be constructed by concatenation of generators in the model \mathcal{G} . We denote the graphical structure of a word w under \mathcal{G}_{max} as $G_{max}(w)$, and similarly define $G_{mid}(w)$ and $G_{min}(w)$.

3.2 DNA origami rewriting systems

It is straightforward that for any alphabet, given two words, the graphical structure of their concatenation is unique. Thus, if $G(w_1) = G(w_2)$ under a model \mathcal{G} , we say $w_1 \sim w_2$ in \mathcal{G} and define a rewriting rule $w_1 \leftrightarrow w_2$ between two equivalent words. Due to difference of context structures, rewriting rules for $\mathcal{G}_{max(n)}$, $\mathcal{G}_{mid(n)}$ and $\mathcal{G}_{min(n)}$ differ from each other. For each structure, we find the set of basic rewriting rules that generate the equivalence and analyze the set of distinct equivalence classes.

3.2.1 $\mathcal{G}_{max(n)}$ case

We first observe that all staples and scaffolds in $\mathcal{G}_{max(n)}$ are real. We observe that except for added directions, their

concatenation results in a bijection between scaffolds and staples without lasting conflict of the directions. Moreover, scaffolds in α_i (and staples in β_i) are straight and do not affect the structure of scaffolds (staples) when concatenated. For convenience, we use γ and δ to represent an arbitrary generator, and $\bar{\gamma}$ to denote the complementary generator of γ . By concatenating generators we obtain the following rules that form the set $R_{max(n)}$ (Fig. 7 shows the inter-commutation rule):

1. (inter-commutation rule) $\gamma_i\bar{\gamma}_j \leftrightarrow \bar{\gamma}_j\gamma_i$
2. (idempotency rule) $\gamma_i\gamma_i \leftrightarrow \gamma_i$
3. (intra-commutation rule) $\gamma_i\gamma_j \leftrightarrow \gamma_j\gamma_i$ for $|i - j| \geq 2$
4. (TL relation rule) $\gamma_i\gamma_j\gamma_i \leftrightarrow \gamma_i$ for $|i - j| = 1$

We define the set $\mathcal{O}_{max(n)}$ of equivalence classes based on $\hat{R}_{max(n)}$. We say that a rule is *non-increasing* if the right-handed side word is not longer than the left-handed side word. A sequence of rewriting rules with only non-increasing rules is said to be a non-increasing rewriting.

We partition Σ_n into $\Sigma_{(\alpha)_n} = \{\alpha_1, \dots, \alpha_{n-1}\}$ and $\Sigma_{(\beta)_n} = \{\beta_1, \dots, \beta_{n-1}\}$. Using the inter-commutation rule, we may rewrite any word w to $w_a w_b$, where $w_a \in \Sigma_{(\alpha)_n}^*$ and $w_b \in \Sigma_{(\beta)_n}^*$. We say that such a word $w_a w_b$ is in an inter-commutation-free form. Also, using intra-commutation rules, we may set additional conditions for $w_a = u_1 u_2 \dots u_p$, where $u_i = (\alpha_{j_i} \alpha_{j_i-1} \dots \alpha_{k_i})$, j_p is the maximum subscript in w_a , $j_{i+1} > j_i$ and $k_{i+1} > k_i$ for $1 \leq i < p$, and a similar condition for w_b . Such w_a and w_b are unique (Jones 1983), and we call such $w_a w_b$ a commutation-free form of w .

We regard the graphical structure of a word as pairs of scaffolds and staples, which can be seen as two independent Jones monoid diagrams. Knowing that the relations 1 to 3 of the Jones monoid can sufficiently describe equivalence of diagrams, we have the following theorem:

Theorem 1 For all $w_1, w_2 \in \Sigma_n^*$, $G_{max}(w_1) = G_{max}(w_2)$ if and only if $w_1 \sim w_2$ under $\hat{R}_{max(n)}$. In other words, there exists a bijection between $\mathcal{G}_{max(n)}$ and $\mathcal{O}_{max(n)}$.

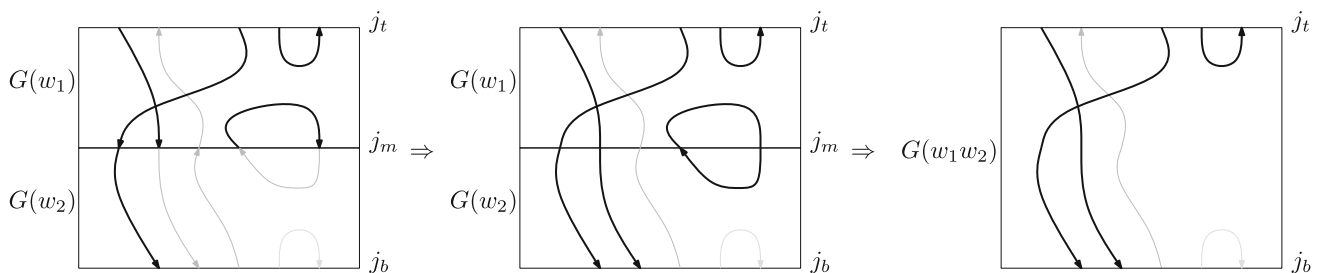


Fig. 6 Scaffold concatenation of $G(w_1)$ and $G(w_2)$. Real scaffolds are represented by thick lines for better visibility

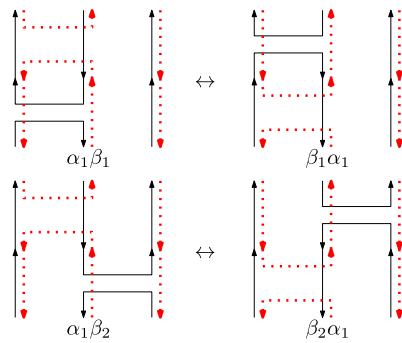


Fig. 7 Inter-commutation rewriting rule for $\mathcal{G}_{max(3)}$

Given n , the number of elements of \mathcal{J}_n is equal to the Catalan number $C_n = \frac{1}{n+1} \binom{2n}{n}$ (Lau and FitzGerald 2006), and the maximum size of an element in \mathcal{J}_n is $\lfloor \frac{n^2}{4} \rfloor$ (Dolinka and East 2017; Jones 1983). Thus, the following remark holds.

Remark 1 Given n , $|\mathcal{O}_{max(n)}| = \left(\frac{1}{n+1} \binom{2n}{n}\right)^2$, and the maximum size of an irreducible word in $\mathcal{O}_{max(n)}$ is $2 \lfloor \frac{n^2}{4} \rfloor$.

Since the graphical structures in $\mathcal{G}_{max(n)}$ correspond to products of two Jones monoid diagrams, we use the following Lemma to obtain the proposed optimization algorithm.

Lemma 1 Given two elements $w_1, w_2 \in \mathcal{J}_n$ where $|w_2| \leq |w_1|$, there is a non-increasing rewriting $w_1 \sim w_2$.

Proof We recall the rewriting rules for \mathcal{J}_n .

1. $h_i h_i \leftrightarrow h_i$
2. $h_i h_j \leftrightarrow h_j h_i$ for $|i - j| \geq 2$
3. $h_i h_j h_i \leftrightarrow h_i$ for $|i - j| = 1$

Suppose $w_1, w_2 \in \mathcal{J}_n$, $|w_2| \leq |w_1|$ and we can rewrite w_1 as w_2 . Let w'_1 be the last word in the sequence of rewriting such that an increasing rule is applied to w'_1 . If such w'_1 does not exist, then the whole rewriting is non-increasing. Otherwise, we claim that there exists another sequence of non-increasing rewriting from w'_1 to w_2 . Note that rule 2 is the only rule that changes location of generators. Moreover, applying rules 1 or 3 on a word does not result in an additional pair of generators that rule 2 can be applied.

- If the first rule is rule 1 ($h_i \leftrightarrow h_i h_i$), the resulting two h_i 's should be involved in non-increasing rule 1 or 3 in the following sequence. The vertical sequence of Fig. 8a shows an example of such sequences where the left h_i is used in rule 1 and the right h_i is used in rule 3, where blue areas represent generators that h_i can

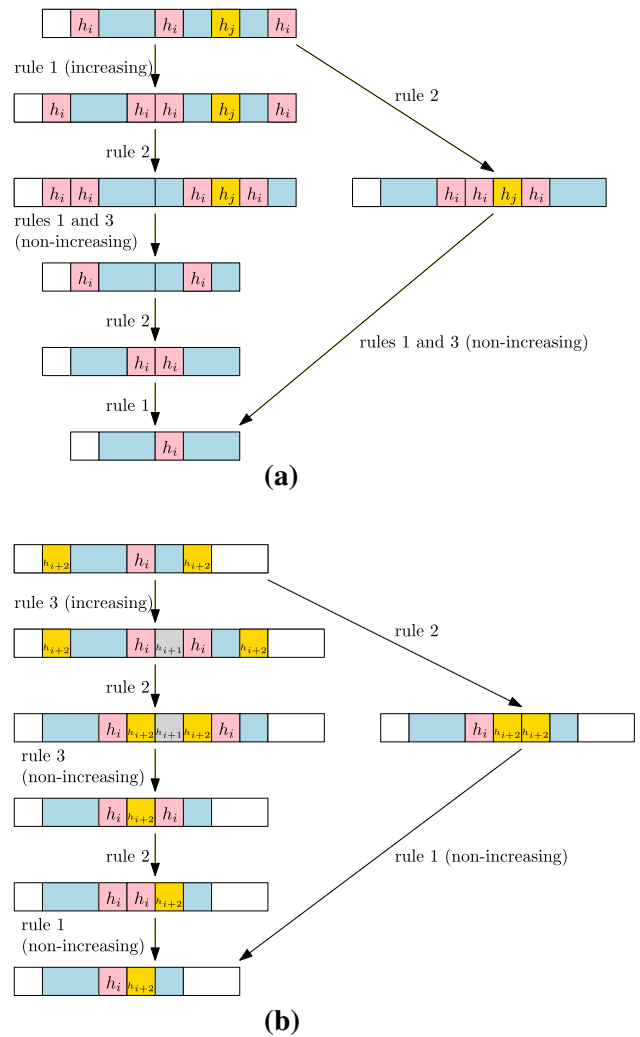


Fig. 8 Examples of sequences of rewriting rules that leads to the same element, where one uses only non-increasing rules. **a** The first rule is rule 1. **b** The first rule is rule 3

switch the location with using rule 2. For such sequences, we can always find another sequence without increasing rules as the right sequence of Fig. 8a.

- If the first rule is rule 3 ($h_i \leftrightarrow h_i h_j h_i$ for $|i - j| = 1$), the resulting h_j should be involved in non-increasing rule 1 or 3 in the following sequence. Without loss of generality, we assume that $j = i + 1$. Since $h_{i+1} h_i \neq h_i h_{i+1}$, h_j cannot be used in rule 1. The vertical sequence of Fig. 8b shows an example of such sequence where the middle $h_j = h_{i+1}$ is used in rule 3, and blue areas represent generators that h_{i+2} can switch the location with. For such a sequence, we can always find another non-increasing sequence as the right sequence of Fig. 8b.

We may continuously find an increasing rewriting in the sequence and an alternate sequence of non-increasing rewriting to w_2 . \square

Theorem 2 Given a word $w_0 \in \Sigma_n^*$ of size m , an irreducible word in $[w_0]$ can be obtained within $O(nm^2)$ time.

Proof For a given w_0 , we rewrite w_0 as w in the inter-commutation-free form, which takes $O(m^2)$ time. For each $i = 1, \dots, n$, determine whether one of the following conditions in w holds and rewrite w accordingly:

1. If $w = v_1\gamma_i v_2 \gamma_i v_3$ where $v_1, v_2, v_3 \in \Sigma_n^*$ and v_2 does not have γ_{i+1}, γ_i and γ_{i-1} , then rewrite w as $v_1 v_2 \gamma_i v_3$. This is possible by the rule 3 and then 2.
2. If $w = v_1 \gamma_i v_2 \gamma_j v_3 \gamma_i v_4$ where $v_1, v_2, v_3, v_4 \in \Sigma_n^*$, $|i - j| = 1$ and v_2, v_3 do not have γ_{i+1}, γ_i and γ_{i-1} , then rewrite w as $v_1 v_2 \gamma_i v_3 v_4$. This is possible by the rule 3 and then 4.

It is straightforward that both rewritings are non-increasing. For each i , the iteration checking whether w satisfies the given form in the conditions 1 and 2 for $(i, j = i + 1$ or $i - 1)$'s is done in $O(m)$ time. It takes $O(nm)$ time to do one rewriting in 1 or 2, which decreases the size of the word by one or two symbols. Thus, it takes at most $O(nm^2)$ to finish the whole process. For the final word w' , conditions in 1 and 2 are no longer satisfied and there is no sequence of non-increasing rewriting that decreases the size of the word. Thus, w' is irreducible by Lemma 1. \square

3.2.2 $\mathcal{G}_{mid(n)}$ case

Similar to the $\mathcal{G}_{max(n)}$ case, we have the following rewriting rules:

1. (Inter-commutation) $\gamma_i \bar{\gamma}_j \leftrightarrow \bar{\gamma}_j \gamma_i$
2. (Idempotency) $\gamma_i \gamma_i \leftrightarrow \gamma_i$
3. (Intra-commutation) $\gamma_i \gamma_j \leftrightarrow \gamma_j \gamma_i$ for $|i - j| \geq 2$

Due to the lack of default real staples in generators, we cannot directly introduce the rewriting rule $\gamma_i \gamma_j \gamma_i \leftrightarrow \gamma_i$ for $|i - j| = 1$. For example, we cannot rewrite $\alpha_1 \alpha_2 \alpha_1$ as α_1 , since $\alpha_1 \alpha_2 \alpha_1$ has a straight real staple at the third column while α_1 does not (see Fig. 9a). We introduce the *span* of a word w as the set of columns $span(w) = \bigcup_{\gamma_{j_{inw}}} \{j, j + 1\}$.

Lemma 2 Under $\mathcal{G}_{mid(n)}$, the *span* equals to the set of columns where real staples exist.

Proof We can prove the statement by induction on the size of the word. It is straightforward that the statement holds for generators. Assume that the statement holds for all $|w| = m$. For a word $w' = w \gamma_i$, if i and $i + 1$ are both in $span(w)$, concatenation does not change the span and the

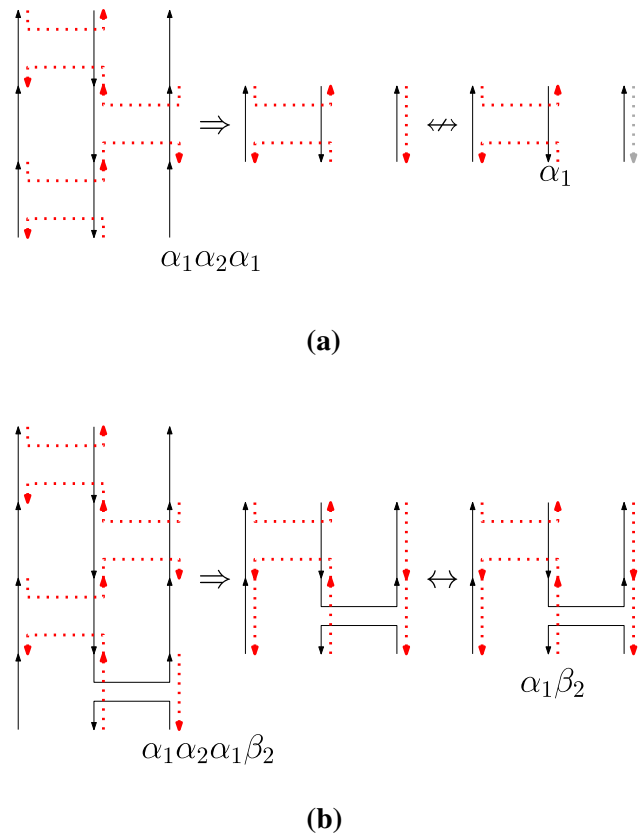


Fig. 9 Examples of equivalence in $\mathcal{G}_{mid(3)}$. **a** $\alpha_1 \alpha_2 \alpha_1 \sim \alpha_1$ **b** $\alpha_1 \alpha_2 \alpha_1 \beta_2 \sim \alpha_1 \beta_2$

statement holds. If i or $i + 1$ are not in $span(w)$, then the span has new columns, and the graphical structure of w' has real staples for these columns, which makes the statement true. \square

Directly from Lemma 2, it follows that two equivalent words have the same span, and rewriting rules in the Jones monoid can be applied when both sides have the same span. For example, $\alpha_1 \alpha_2 \alpha_1 \beta_2 \leftrightarrow \alpha_1 \beta_2$ holds as in Fig. 9b because β_2 adds 3 to $span(\alpha_1 \beta_2)$ and $span(\alpha_1 \alpha_2 \alpha_1 \beta_2) = span(\alpha_1 \beta_2)$. In general, we have the following additional rewriting rules, where $\delta \in \{\alpha, \beta\}$ and $v \in \Sigma_n^*$:

- 4*. $\delta_j v \gamma_i \gamma_{i-1} \gamma_i \leftrightarrow \delta_j v \gamma_i$ if $j = i - 1$ or $i - 2$
- 5*. $\delta_j v \gamma_i \gamma_{i+1} \gamma_i \leftrightarrow \delta_j v \gamma_i$ if $j = i + 1$ or $i + 2$
- 6*. $\gamma_i \gamma_{i-1} \gamma_i v \delta_j \leftrightarrow \gamma_i v \delta_j$ if $j = i - 1$ or $i - 2$
- 7*. $\gamma_i \gamma_{i+1} \gamma_i v \delta_j \leftrightarrow \gamma_i v \delta_j$ if $j = i + 1$ or $i + 2$

The symbol δ_j in the right hand side of rules 4* to 7* ensures that the span of the two words is the same. Note that $\{v\} = \Sigma_n^*$ in rules 4* to 7* is infinite, but we prove that $\{v\}$ can be reduced to a finite set of words while maintaining the equivalent rewriting system in Theorem 3. For better description of such a subset, we define a zig-zag

word $w \in \Sigma_n^*$ to be a word where each pair of adjacent generators in w have adjacent indices. We call a maximal subword of increasing (decreasing) indices as zig (zag). For example, $w = \alpha_3\alpha_4\alpha_3\alpha_2\alpha_1\alpha_2$ is a zig-zag word with a zig-zag-zig sequence in the word: $\alpha_3\alpha_4$ being the zig, $\alpha_4\alpha_3\alpha_2\alpha_1$ being the zag, and $\alpha_1\alpha_2$ being the zig. Using rules 2 to 7*, we can rewrite any zig-zag word that consists of single generators type either in $\Sigma_{(\alpha)n}$, or $\Sigma_{(\beta)n}$, as a zig-zag word with at most three zigs or zags, which we call the *zig-zag normal form*, or ZNF in short.

Theorem 3 Rules 1 to 3 and 4* to 7* generate the same equivalence relation on Σ^* as rules 1 to 7 below where $v = \epsilon$, or $v \in \Sigma_{(\gamma)n}^*$ and $\gamma_j v \gamma_i$ in rules 4 and 5 ($\gamma_i v \gamma_j$ in rules 6 and 7) is in ZNF.

4. $\gamma_j v \gamma_i \gamma_{i-1} \gamma_i \leftrightarrow \gamma_j v \gamma_i$ if $j = i - 1$ or $i - 2$
5. $\gamma_j v \gamma_i \gamma_{i+1} \gamma_i \leftrightarrow \gamma_j v \gamma_i$ if $j = i + 1$ or $i + 2$
6. $\gamma_i \gamma_{i-1} \gamma_i v \gamma_j \leftrightarrow \gamma_i v \gamma_j$ if $j = i - 1$ or $i - 2$
7. $\gamma_i \gamma_{i+1} \gamma_i v \gamma_j \leftrightarrow \gamma_i v \gamma_j$ if $j = i + 1$ or $i + 2$

Proof Given rule 4*: $w_1 = \delta_j v \gamma_i \gamma_{i-1} \gamma_i \leftrightarrow w_2 = \delta_j v \gamma_i$ if $j = i - 1$ or $i - 2$, here we prove that rule 4* can be replaced with rules 4 to 7 to generate the same equivalence relation. Similar simplification works for rules 5* to 7*, and we focus on rule 4* in the proof. The proof consists of establishing three assumptions on δ and v .

First, we prove that we may assume $\delta = \gamma$. If $\delta = \bar{\gamma}$, we can rewrite v as a concatenation of a prefix $v_\gamma \in \Sigma_{(\gamma)n}^*$ and a suffix $v_\delta \in \Sigma_{(\delta)n}^*$. Then,

$$\delta_j v_\gamma v_\delta \gamma_i \gamma_{i-1} \gamma_i \leftrightarrow v_\gamma \delta_j \gamma_i \gamma_{i-1} \gamma_i v_\delta \leftrightarrow v_\gamma \delta_j \gamma_i v_\delta \leftrightarrow \delta_j v_\gamma v_\delta \gamma_i = \delta_j v \gamma_i$$

, and vice versa.

Second, we prove that we may assume $v \in \Sigma_{(\gamma)n}^*$. If $v \notin \Sigma_{(\gamma)n}^*$, we can rewrite v as a concatenation of a prefix $v_\gamma \in \Sigma_{(\gamma)n}^*$ and a suffix $v_{\bar{\gamma}} \in \Sigma_{(\bar{\gamma})n}^*$. Then, $\gamma_j v_\gamma v_{\bar{\gamma}} \gamma_i \gamma_{i-1} \gamma_i \leftrightarrow v_{\bar{\gamma}} \gamma_j v_\gamma \gamma_i \gamma_{i-1} \gamma_i \leftrightarrow v_{\bar{\gamma}} \gamma_j v_\gamma \gamma_i \leftrightarrow \gamma_j v_\gamma v_{\bar{\gamma}} \gamma_i = \gamma_j v \gamma_i$, and vice versa.

Third, we prove the given statement under the previously proved assumptions $\delta = \gamma$ and $v \in \Sigma_{(\gamma)n}^*$, by induction on $|v|$. It is straightforward that the statement holds for $v = \epsilon$. For $|v| = 1$, we may assume that $v = \gamma_h$. According to h , we have the following cases (see Fig. 10):

- (1) If $h > j + 1$ or $h < j - 1$, we can switch γ_h and γ_j .
- (2) If $h = j + 1$, then $h = i$ or $i - 1$.
 - (2-a) If $h = i$, $\gamma_h \gamma_i$ can be rewritten as γ_i .
 - (2-b) If $h = i - 1$, $\gamma_j \gamma_h \gamma_i$ is in ZNF.
- (3) If $h = j$, we can rewrite $\gamma_j \gamma_h$ as γ_j .
- (4) If $h = j - 1$, then $h = i - 2$ or $i - 3$.

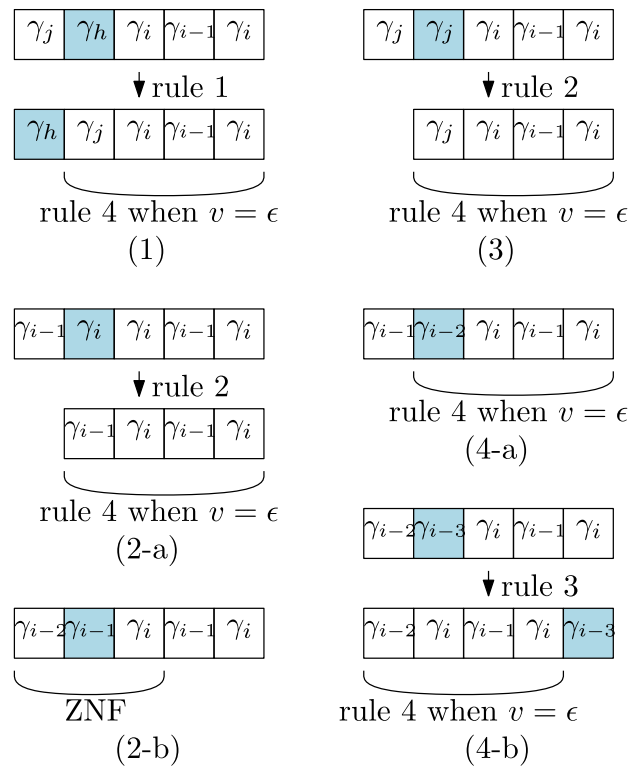


Fig. 10 Base case analysis when $|v| = 1$

- (4-a) If $h = i - 2$, we may regard γ_h as new γ_j .
- (4-b) If $h = i - 3$, we can move γ_h to the right of $\gamma_i \gamma_{i-1} \gamma_i$.

Then, assume that the claim holds for all v where $|v| \leq m$. For $|v| = m + 1$, we will prove that we can rewrite w_1 to find a subword $w'_1 = \gamma_j v' \gamma_i \gamma_{i-1} \gamma_i$ such that $|v'| \leq m$, which leads to the induction hypothesis. Let $\gamma_j p$ be the maximal zig-zag prefix of $\gamma_j v \gamma_i$, and assume that $p \neq v$. Let γ_t be the last generator of p , and γ_h be the first generator after p , which is in v . Let $\max(\gamma_j p)$ ($\min(\gamma_j p)$) be the largest (smallest) index of generators in $\gamma_j p$. According to h , we have the following cases (see Fig. 11):

- (1) If $h > \max(\gamma_j p) + 1$, we can move γ_h to the left of γ_j , which reduces the size of v by 1.
- (2) If $h = \max(\gamma_j p) + 1$, since $\gamma_j p$ is the maximal zig-zag prefix of v , $t \neq \gamma_{\max(\gamma_j p)}$. We consider two cases:
 - (2-a) If $j = \max(\gamma_j p)$, $h = j + 1$.
 - (2-a-i) If $j = i - 1$, then the first generator of p is γ_{i-2} , and we may regard it as a new γ_j .
 - (2-a-ii) If $j = i - 2$, then $h = i - 1$, and we may regard γ_h as a new γ_j .

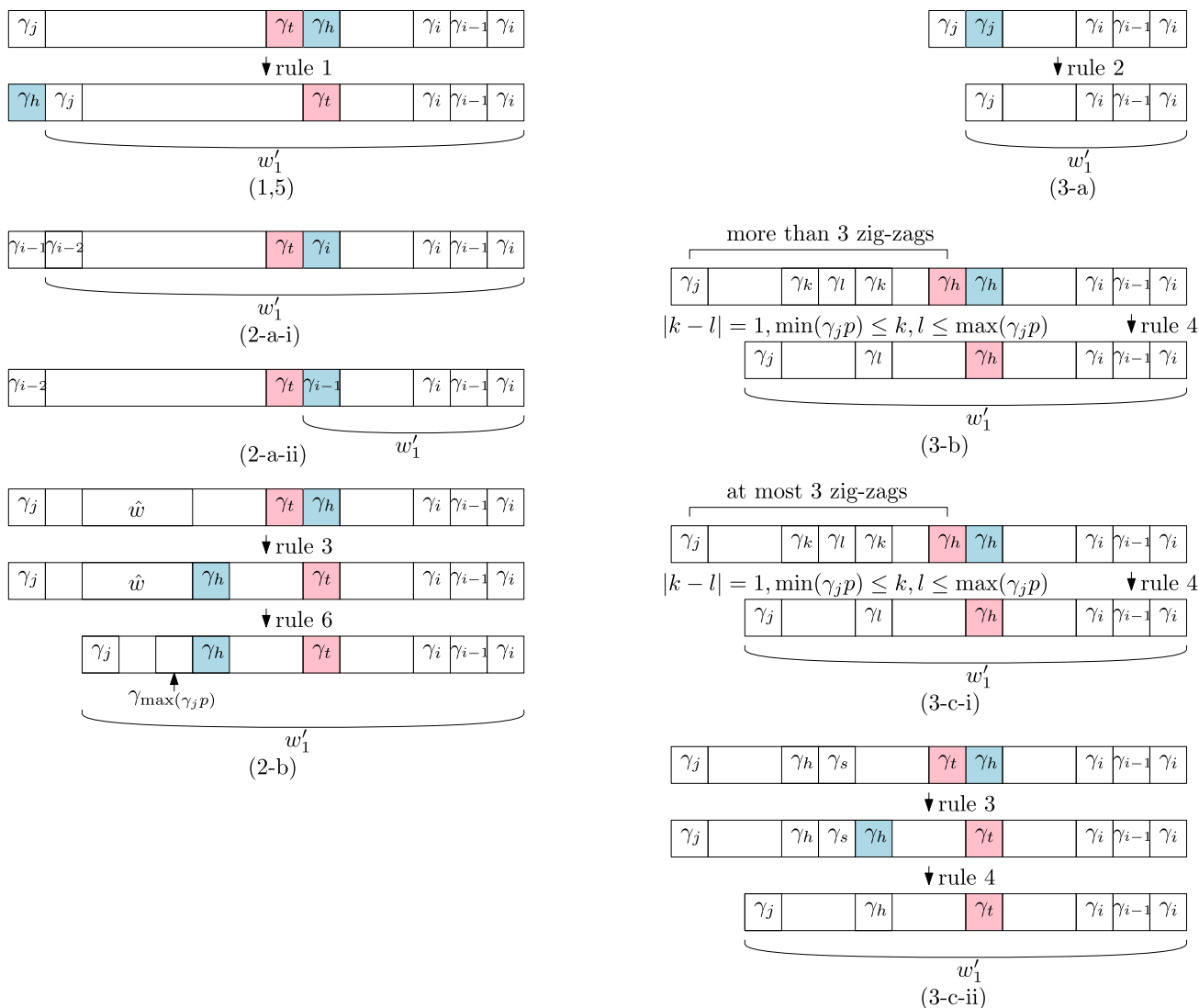


Fig. 11 Inductive step

(2-b) If $j \neq \max(\gamma_j p)$, then there exists the rightmost subword $\hat{w} = \gamma_{\max(\gamma_j p)-1} \gamma_{\max(\gamma_j p)}$ in $\gamma_j p$. We can move γ_h to the right of \hat{w} , and reduce $\hat{w} \gamma_h$ to $\gamma_{\max(\gamma_j p)} \gamma_h$, which reduces the size of v by 2.

(3) If $\min(\gamma_j p) \leq h \leq \max(\gamma_j p)$, we consider four cases:

(3-a) If $\min(\gamma_j p) = \max(\gamma_j p)$, $p = \epsilon$, $h = j$ and we may reduce $\gamma_j \gamma_h$ as γ_j , which reduces the size of v by 1.

(3-b) If $\gamma_j p$ has more than three zigs or zags, it can be rewritten using rule 4 to 7, which reduces the size of v by at least 2.

(3-c) If $\gamma_j p$ has at most three zigs or zags,

(3-c-i) If $j = \max(\gamma_j p)$ or $j = \min(\gamma_j p)$ or $t = \max(\gamma_j p)$ or $t = \min(\gamma_j p)$, then $\gamma_j p$ can be rewritten using rule 4 to 7, which reduces the size of v by at least 2.

(3-c-ii) Otherwise, we can move γ_h into $\gamma_j p$ so that $\gamma_j p$ has a subword $\gamma_h \gamma_s \gamma_h$ where $|s-h|=1$, since $\gamma_j p$ is the maximal zig-zag prefix of $\gamma_j v \gamma_i$ and γ_h cannot 'extend' the zig-suffix or the zag-suffix of $\gamma_j p$. Such subword can be rewritten as γ_h , which reduces the size of v by 2.

- (4) If $h = \min(\gamma_j p) - 1$, the case is similar to $h = \max(\gamma_j p) + 1$ case.
- (5) If $h < \min(\gamma_j p) - 1$, we can move γ_h to the left of γ_j , which reduces the size of v by 1.

When $p = v$, $\gamma_h = \gamma_i$. The case analysis is similar to the above, but we cannot move γ_h to the left of γ_j , which breaks the subword $\gamma_i \gamma_{i-1} \gamma_i$. Thus, we check the following two cases such that we moved γ_h to the left of γ_j (see Fig. 12):

- (1) If $h = i > \max(\gamma_j p) + 1$, since $j = i - 1$ or $i - 2$, $\max(\gamma_j p) = i - 2$.
 - (1-a) If $t = i - 2$, we may regard γ_t as a new γ_j .
 - (1-b) If $t \leq i - 3$, we may move γ_t to the right of $\gamma_i \gamma_{i-1} \gamma_i$, which reduces the size of v by 1.
- (2) Since $j = i - 1$ or $i - 2$, $h = i$ cannot be less than $\min(\gamma_j p) - 1$.

□

Thus the rules 4* to 7* reduce to the following rewriting rules for $v \in \Sigma_{(v)}^*$ such that either $v = \epsilon$, or $\gamma_j v \gamma_i$ is in ZNF in rule 4 and 5 ($\gamma_i v \gamma_j$ in rule 6 and 7):

- 4. $\gamma_j v \gamma_i \gamma_{i-1} \gamma_i \leftrightarrow \gamma_j v \gamma_i$ if $j = i - 1$ or $i - 2$
- 5. $\gamma_j v \gamma_i \gamma_{i+1} \gamma_i \leftrightarrow \gamma_j v \gamma_i$ if $j = i + 1$ or $i + 2$
- 6. $\gamma_i \gamma_{i-1} \gamma_i v \gamma_j \leftrightarrow \gamma_i v \gamma_j$ if $j = i - 1$ or $i - 2$
- 7. $\gamma_i \gamma_{i+1} \gamma_i v \gamma_j \leftrightarrow \gamma_i v \gamma_j$ if $j = i + 1$ or $i + 2$

Comparing the rewriting rules with $R_{max(n)}$, rules 4 to 7 are extensions of rule 4 of $R_{max(n)}$, but the ruleset stays to be finite. Note that if a zig-zag word that consists of single generators type has more than three zigs or zags, it can be successfully reduced to a ZNF word using rules 2 to 7.

For given n , let the set $R_{mid(n)}$ of rewriting rules consist of the above seven kinds of rules for $1 \leq i, j \leq n$. As observed, since there are only finitely many words in ZNF, the set $R_{mid(n)}$ is finite. Then $\mathcal{O}_{mid(n)} = \Sigma_n^* / \hat{R}_{mid(n)}$.

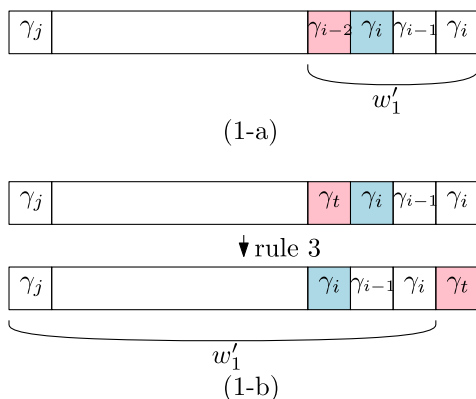


Fig. 12 Inductive step when $p = v$

Theorem 4 For all $w_1, w_2 \in \Sigma_n^*$, $G_{mid}(w_1) = G_{mid}(w_2)$ if and only if $w_1 \sim w_2$ under $\hat{R}_{mid(n)}$. In other words, there exists a bijection between $\mathcal{G}_{mid(n)}$ and $\mathcal{O}_{mid(n)}$.

Proof Note that if $w_1 \sim w_2$ under $\hat{R}_{mid(n)}$, then $G_{mid}(w_1) = G_{mid}(w_2)$ from definitions of rewriting rules. If $G_{max}(w_1) = G_{max}(w_2)$ and $span(w_1) = span(w_2)$, then $G_{mid}(w_1) = G_{mid}(w_2)$ from Lemma 2. Moreover, if $G_{max}(w_1) = G_{max}(w_2)$ and $span(w_1) \neq span(w_2)$, then $G_{mid}(w_1) \neq G_{mid}(w_2)$ from Lemma 2. Thus, $G_{mid}(w_1) = G_{mid}(w_2)$ if and only if $G_{max}(w_1) = G_{max}(w_2)$ and $span(w_1) = span(w_2)$. From Theorem 1, the set of the following (general) rules are sufficient to describe equivalence under $\mathcal{G}_{max(n)}$ when $v_1, v_2 \in \Sigma_n^*$.

- (i) $v_1 \gamma_i \overline{\gamma_j} v_2 \leftrightarrow v_1 \overline{\gamma_j} \gamma_i v_2$
- (ii) $v_1 \gamma_i \gamma_i v_2 \leftrightarrow v_1 \gamma_i v_2$
- (iii) $v_1 \gamma_i \gamma_j v_2 \leftrightarrow v_1 \gamma_j \gamma_i v_2$ for $|i - j| \geq 2$
- (iv) $v_1 \gamma_i \gamma_j \gamma_i v_2 \leftrightarrow v_1 \gamma_i v_2$ for $|i - j| = 1$

In the proof, $R_{max(n)}$ denotes the set of these general rules. Now, there exists a set P of pairs of words (w_1, w_2) where $G_{max}(w_1) = G_{max}(w_2)$ and $G_{mid}(w_1) = G_{mid}(w_2)$, a set H of pairs of words (w_3, w_4) where $G_{max}(w_3) = G_{max}(w_4)$ and $G_{mid}(w_3) \neq G_{mid}(w_4)$, and another set N of pairs of words (w_5, w_6) where $G_{max}(w_5) \neq G_{max}(w_6)$ and $G_{mid}(w_5) = G_{mid}(w_6)$. The sets P, H, N are disjoint and $P \cup H \cup N = \Sigma_n^* \times \Sigma_n^*$. Now, $R_{max(n)}$ can be partitioned into two sets R_{diff} and R_{same} , where all rules in R_{diff} have different span for two sides and all rules in R_{same} have the same span for two sides. Then, the following statements hold:

- 1. For a pair $(w_1, w_2) \in P$, there exists a sequence of rewriting that rewrites w_1 as w_2 only using rules in R_{same} : For a pair $(w_1, w_2) \in P$, from Lemma 1, there exists a sequence of non-increasing rewriting that rewrite w_1 as w_2 given $|w_2| \leq |w_1|$. We observe that the only rules that change the size of the word are rules (ii) and (iv). Rule (ii) does not change the span, and non-increasing rules in rule (iv) do not increase the size of the span. Since $w_1 \sim w_2$ in $\mathcal{G}_{mid(n)}$, $span(w_1) = span(w_2)$. Thus, there exists a sequence of rewriting that rewrites w_1 as w_2 only using rules in R_{same} .
- 2. For a pair $(w_3, w_4) \in H$, all sequences of rewriting that rewrite w_3 as w_4 have a rule from R_{diff} . In other words, we cannot rewrite w_3 as w_4 only using rules in R_{same} .
- 3. For a pair $(w_5, w_6) \in N$, w_5 cannot be rewritten as w_6 using R_{same} , since $R_{same} \subseteq R_{max(n)}$.

Based on these statements, we can claim that a pair (w_1, w_2) is in P if and only if $w_1 \sim w_2$ under \hat{R}_{same} . Rules (i) to (iii) from $R_{max(n)}$ have the same span for the both sides, and they are in R_{same} . For rule (iv) from $R_{max(n)}$,

the subset of the rules where both sides have the same span is rules 4 to 7 in $R_{mid(n)}$. Thus, $\hat{R}_{mid(n)} = \hat{R}_{same}$ and $G_{mid}(w_1) = G_{mid}(w_2)$ if and only if $w_1 \sim w_2$ under $\hat{R}_{mid(n)}$. \square

We compute the number of equivalence classes of words in $\mathcal{O}_{mid(n)}$. We use a binary string of length n to represent the graphical structures in $\mathcal{G}_{mid(n)}$ such that the i th bit equals 1 if and only if the i th staple and the i th scaffold is a straight line. Each binary string is uniquely determined with a tuple $(a_1, b_1, \dots, a_k, b_k)$ where a_i (b_i) represents the number of i th consecutive 0's (1's). For example, the 8-bit binary string 00111000 corresponds to a tuple $(2, 3, 3, 0)$. In particular, the bit 0 corresponds to $(1, 0)$ and the bit 1 to $(0, 1)$. Let $\mathbb{N}^0 = \mathbb{N} \cup \{0\}$. The set of tuples corresponding to binary strings of length n is denoted $T_n = \{p = (a_1, b_1, \dots, a_k, b_k) \mid k \geq 1, \text{ for all } i, a_i, b_i \in \mathbb{N}^0 \text{ and } \sum_{i=1}^k (a_i + b_i) = n\}$. Note that T_n is the set of partitions of n in $2k$ summands.

Theorem 5 Given $n \in \mathbb{N}^0$, for each tuple $p \in \bigcup_{1 \leq i \leq n} T_i$, let $D(p) \in \mathbb{N}^0$ be recursively defined as follows:

- for $p \in T_0$, $D(0, 0) = 1$.
- for $p \in T_1$, $D(p) = 1$ if $p = (0, 1)$ and $D(p) = 0$ if $p = (1, 0)$.
- for $p = (a_1, b_1, \dots, a_k, b_k) \in T_n$, ($n > 0$) we have $D(p) = \prod_{i=1}^k D(a_i, 0)$.
- for $n > 1$, we have $D(0, n) = 1$ and

$$D(n, 0) = \left(\frac{1}{n+1} \binom{2n}{n} \right)^2 - \sum_{p \in T_n \setminus \{(n, 0)\}} D(p).$$

Then, $|\mathcal{O}_{mid(n)}|$ is given as

$$|\mathcal{O}_{mid(n)}| = d(n) = \sum_{p \in T_n} (D(p) \times x(p)) - n,$$

where $x(a_1, b_1, \dots, a_k, b_k) =$

- $(b_1 + 1)$ if $k = 1$,
- $(b_1 + 1) \cdot \prod_{i=2}^{k-1} \left(\frac{b_i(b_i + 1)}{2} + 1 \right) \cdot (b_k + 1)$ if $k \neq 1, a_1 = 0$,
- $\prod_{i=1}^{k-1} \left(\frac{b_i(b_i + 1)}{2} + 1 \right) \cdot (b_k + 1)$ if $k \neq 1, a_1 > 0$.

Proof Figure 13 enumerates representative words in $\mathcal{O}_{mid(3)}$. We observe that the set of representative words in $\mathcal{O}_{mid(n)}$ is a superset of the set of representative words in $\mathcal{O}_{max(n)}$. Graphically, we observe that $\mathcal{G}_{mid(n)}$ is a superset

| | ϵ | α_1 | α_2 | $\alpha_1\alpha_2$ | $\alpha_2\alpha_1$ | $\alpha_1\alpha_2\alpha_1$ | $\alpha_2\alpha_1\alpha_2$ |
|-------------------------|-------------------------|--------------------------|--------------------------|----------------------------------|----------------------------------|-----------------------------------|-----------------------------------|
| ϵ | ϵ | α_1 | α_2 | $\alpha_1\alpha_2$ | $\alpha_2\alpha_1$ | $\alpha_1\alpha_2\alpha_1$ | $\alpha_2\alpha_1\alpha_2$ |
| β_1 | β_1 | $\alpha_1\beta_1$ | $\alpha_2\beta_1$ | $\alpha_1\alpha_2\beta_1$ | $\alpha_2\alpha_1\beta_1$ | $\alpha_1\alpha_2\alpha_1\beta_1$ | |
| β_2 | β_2 | $\alpha_1\beta_2$ | $\alpha_2\beta_2$ | $\alpha_1\alpha_2\beta_2$ | $\alpha_2\alpha_1\beta_2$ | | $\alpha_2\alpha_1\alpha_2\beta_2$ |
| $\beta_1\beta_2$ | $\beta_1\beta_2$ | $\alpha_1\beta_1\beta_2$ | $\alpha_2\beta_1\beta_2$ | $\alpha_1\alpha_2\beta_1\beta_2$ | $\alpha_2\alpha_1\beta_1\beta_2$ | | |
| $\beta_2\beta_1$ | $\beta_2\beta_1$ | $\alpha_1\beta_2\beta_1$ | $\alpha_2\beta_2\beta_1$ | $\alpha_1\alpha_2\beta_2\beta_1$ | $\alpha_2\alpha_1\beta_2\beta_1$ | | |
| $\beta_1\beta_2\beta_1$ | $\beta_1\beta_2\beta_1$ | | | | | | |
| $\beta_2\beta_1\beta_2$ | $\beta_2\beta_1\beta_2$ | | | | | | |

Fig. 13 The set of representative words in $\mathcal{O}_{mid(3)}$. Gray headers represent representative words corresponding to elements in \mathcal{J}_3 , and the thick box represents the set of representative words in $\mathcal{O}_{max(3)}$

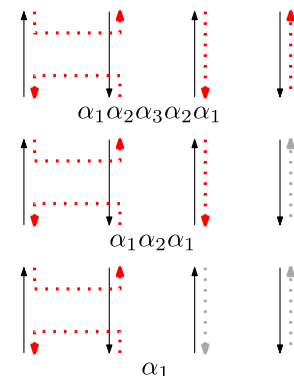
of the set of a pair of diagrams of \mathcal{J}_n , where we regard one as scaffolds and the other as staples. In particular, when a consecutive set of columns adjacent to the span is occupied with both real straight scaffold and staple, there also exists a structure with virtual straight staples in these columns as in Fig. 14.

We can classify graphical structures in $\mathcal{G}_{max(n)}$ by using a binary b of length n , where the i th digit has 1 if the i th column has both straight scaffold and staple, and 0 otherwise. The set of binaries of length n has bijection with the set T_n previously defined. Thus, let $D(p)$ be the number of equivalence classes of words whose graphical structures correspond to $p \in T_n$. It is straightforward that $D(0, 1) = 1$ and $D(1, 0) = 0$. To calculate $D(a_1, b_1, \dots, a_k, b_k)$, columns that has 1's in the binary has only one case (straight scaffold and staple), so we need to multiply all $D(a_i, 0)$'s. For $D(a_1, 0)$, once we know all $D(p)$ where $p \in T_{a_1}$, we can calculate $D(a_1, 0)$ using the fact that

$$|\mathcal{G}_{max(a_1)}|^2 = \left(\frac{1}{a_1 + 1} \binom{2a_1}{a_1} \right)^2 = \sum_{p \in T_{a_1}} D(p).$$

For each $D(p)$, we calculate the number of distinct graphical structures in $\mathcal{G}_{mid(n)}$. Suppose we have j consecutive 1's at the start or the end of the binary that corresponds to p . For such sequence, we can have $j + 1$ distinct sets of virtual straight staples. When we have j consecutive 1's between two consecutive 0's, there are $\sum_{i=1}^j i + 1 = \frac{j(j+1)}{2} + 1$

Fig. 14 Since the third and the fourth columns of $\alpha_1\alpha_2\alpha_3\alpha_2\alpha_1$ are occupied with both real straight scaffold and staple, we also have structures where the staple at the fourth column is virtual ($\alpha_1\alpha_2\alpha_1$) and staples at the third and the fourth columns are virtual (α_1)



distinct sets of virtual straight staples. The number of cases for each consecutive 1's should be multiplied, which results in $x(p)$ in the Theorem. The only exception for this calculation is $D(0, n)$ case, where we have real straight scaffolds and staples for all columns. The only word that corresponds to the structure is the empty word ϵ , and we need to subtract n from $D(0, n) \cdot x(0, n) = n + 1$, which results in the formula in the theorem. Figure 15 shows how we count the number of cases for each $D(p)$.

To justify the counting of virtual staples cases, we claim the following statement: For a graphical structure in $\mathcal{G}_{mid(n)}$, let t_i be 1 if the i th column has real straight scaffold and straight staple, and 0 otherwise. Then, a set of maximal consecutive columns with real straight scaffolds and staples should be adjacent to the i th column where $t_i = 0$, as in Fig. 16a. In other words, there is no set of maximal consecutive columns with real straight scaffolds and staples where both ends are adjacent to straight scaffolds with straight virtual staples, as in Fig. 16b. We prove the statement by induction on the size of the word. It is straightforward that the statement holds for the generators. Assume that the statement holds for all $|w| = m$. For a word $w' = w\gamma_i$, we observe that a column in the graphical structure of w' can have a virtual straight staple only if it had a virtual straight staple in the graphical structure of w . Thus, if i or $i + 1$ is in $span(w)$, the statement holds since the columns that have virtual straight staples do not change. If i and $i + 1$ are not in $span(w)$, we insert the unit of the generator in a set of consecutive columns with straight scaffolds and virtual staples. In that case, concatenation does not create columns that have straight scaffolds and real staples, and the statement holds. \square

For each graphical structure in $\mathcal{G}_{max(n)}$, we may assign a unique binary number of size n , where the i th digit is 1 if the i th column has both straight vertical scaffolds and

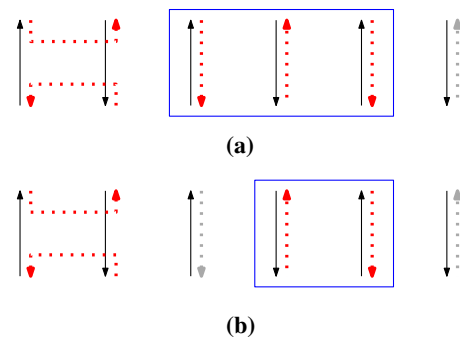


Fig. 16 The set of maximal consecutive columns with real straight scaffolds and staples is represented by a blue box. **a** The set is adjacent to the i th column with $t_i = 0$ on the left. **b** Both ends are adjacent to straight scaffolds with straight virtual staples, which is impossible. (Color figure online)

staples, and 0 otherwise. Then, $D(p)$ is the number of graphical structures in $\mathcal{G}_{max(n)}$ whose assigned binary number corresponds to p . In particular, the sum of all $D(p)$ for $p \in T_n$ is equal to the square of the n th Catalan number by Remark 1. For each p , the term $x(p)$ gives the number of possible combinations of virtual straight staples within the segment of the graphical representation that consist of only vertical straight lines.

The sequence $d(n)$ for $1 \leq n \leq 10$ is 1, 4, 31, 253, 2247, 21817, 227326, 2499598, 28660639, 339816259. It is not listed in the OEIS (<https://oeis.org/>) list of sequences, and the non-recursive formula of $d(n)$ is still open.

Theorem 6 An upper bound of the size $u(n)$ of an irreducible word in $\mathcal{O}_{mid(n)}$ is given by 2^{n-1} for $n \geq 2$.

Proof There exists a representative word $w_a w_b \in \mathcal{O}_{mid(n)}$ in an inter-commutation-free form where $|w_a| = |w_b| = \lfloor \frac{n^2}{4} \rfloor$, and $u(n) \geq \lfloor \frac{n^2}{4} \rfloor$. Aside from $w_a w_b$, we may have representative words in $\mathcal{O}_{mid(n)}$ that exploit rules 4 to 7, not satisfying the condition that $\gamma_i \gamma_j \gamma_i$ is

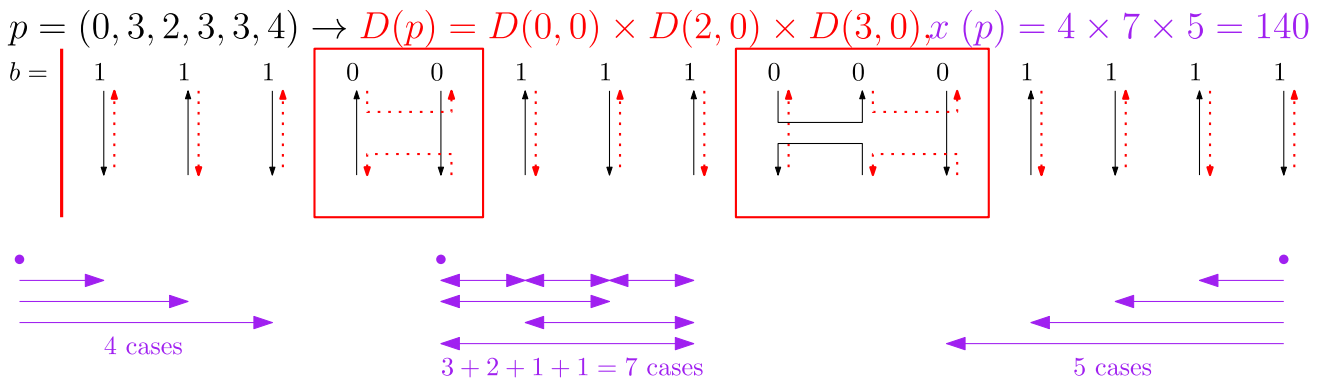


Fig. 15 A graphical structure corresponding to $p = (0, 3, 2, 3, 3, 4)$, and calculation of $D(p)$ and $x(p)$. The binary that corresponds to the structure is stated on the structure. For consecutive 1's in the binary,

we may have arbitrary consecutive virtual staples within, which is counted by purple arrows to calculate $x(p)$. (Color figure online)

reduced to γ_i when $|i - j| = 1$. Namely, we may rewrite γ_i in a word w as $\gamma_i\gamma_j\gamma_i$ and have a distinct word if staples in the resulting word occupy at least one new column.

Given an irreducible word v of size $l(v)$, let $s(v)$ be the size of the span of v . Then, $\frac{s(v)}{2} \leq l(v) \leq u(s(v) - 1)$ holds when $v \neq \epsilon$. Now, suppose for a word $v = v_\alpha v_\beta$ in an inter-commutation-free form, we want to continuously rewrite γ_i as $\gamma_i\gamma_j\gamma_i$ as far as possible while making the resulting words distinct. Let $v_\alpha = \alpha_{i_1} \cdots \alpha_{i_p}$ and $v_\beta = \beta_{j_1} \cdots \beta_{j_q}$. If $v_\alpha = \epsilon$, there exists a longer word $\alpha_1 v_\beta$ which is distinct. Thus, without loss of generality, we may assume that $v_\alpha, v_\beta \neq \epsilon$. Now, for the word v , $l(v) = l(v_\alpha) + l(v_\beta)$ and $\max(s(v_\alpha), s(v_\beta)) \leq s(v) \leq s(v_\alpha) + s(v_\beta)$. Each rewriting of γ_i to $\gamma_i\gamma_j\gamma_i$ increases $s(v)$ by 1 and l by 2, and such rewriting becomes impossible once $s(v)$ becomes n . Without loss of generality, we assume that $s(v_\alpha) \geq s(v_\beta)$. Then, we may have at most $n - s(v_\alpha)$ number of rewriting steps, which results in a word of size $l(v_\alpha) + l(v_\beta) + 2(n - s(v_\alpha)) = 2n - 2s(v_\alpha) + l(v_\alpha) + l(v_\beta) \leq 2n - 2s(v_\alpha) + 2u(s(v_\alpha) - 1)$. Since $u(n) \geq \left\lfloor \frac{n^2}{4} \right\rfloor$, increases as $s(v_\alpha)$ increases. For $s(v_\alpha) = n$, $u(n) \leq 2n - 2n + 2u(n - 1) = 2u(n - 1)$. From $u(2) = 2$, we have the upper bound $u(n) = 2^{n-1}$. \square

The bound in Theorem 6 is not tight, and the exact size of a maximum irreducible word is open.

Theorem 7 *Given a word $w_0 \in [w_0] \in \mathcal{O}_{mid(n)}$ of size m , we can find an irreducible word of $[w_0]$ within $O(nm^2)$ time.*

The proofs of Lemma 1 and Theorem 2 work similarly for Theorem 7. We first rewrite w_0 as w in the inter-commutation-free form. Then we repeatedly find one of the following conditions in w if possible and rewrite w accordingly:

1. If $w = v_1\gamma_i v_2\gamma_i v_3$ where $v_1, v_2, v_3 \in \Sigma_n^*$ and v_2 does not have γ_{i+1} , γ_i and γ_{i-1} , then rewrite w as $v_1 v_2 \gamma_i v_3$.
2. If $w = v_1\gamma_i v_2\gamma_{i+1} v_3\gamma_i v_4$ where $v_1, v_2, v_3, v_4 \in \Sigma_n^*$, v_2, v_3 do not have γ_{i+1} , γ_i and γ_{i-1} , there exists δ_{i+1} in v_1 or v_4 , or δ_{i+2} in v_1, v_2, v_3 or v_4 , then rewrite w as $v_1 v_2 \gamma_i v_3 v_4$.
3. If $w = v_1\gamma_i v_2\gamma_{i-1} v_3\gamma_i v_4$ where $v_1, v_2, v_3, v_4 \in \Sigma_n^*$, v_2, v_3 do not have γ_{i+1} , γ_i and γ_{i-1} , there exists δ_{i-1} in v_1 or v_4 , or δ_{i-2} in v_1, v_2, v_3 or v_4 , then rewrite w as $v_1 v_2 \gamma_i v_3 v_4$.

3.3 Concluding remarks

We have proposed modules and corresponding generators for DNA origami structures, defined concatenation of words and rewriting rules, and analyzed equivalence classes based on graphical equivalence. One model that we have not discussed is $\mathcal{G}_{min(n)}$. For $\mathcal{G}_{min(n)}$, seven types of rewriting rules for $\mathcal{G}_{mid(n)}$ hold. Moreover, we may prove that Theorem 4 holds for $\mathcal{G}_{min(n)}$ using the similar proof. It turns out that there is bijection between $\mathcal{G}_{mid(n)}$ and $\mathcal{G}_{min(n)}$, and $\mathcal{O}_{mid(n)} = \mathcal{O}_{min(n)}$.

Note that the set of words in the first row of Fig. 13 is bijective with the set of distinct elements in the “modified” Jones monoid corresponding to the staple structure, assuming that there are two types (real and virtual) of lines in the graphical representation. However, $\mathcal{O}_{mid(3)}$ is not a bijection of the cross product of the original Jones monoid (for scaffolds) and such modified Jones monoid (for staples), since β_1 and β_2 also have virtual staples.

Graphical structures corresponding to generators α_i 's and β_i 's in Fig. 3 describe crossing of scaffolds and staples in DNA origami well, while using only two types of generators. Here we explore possible further development of generators that are more plausible to DNA origami.

The first observation on the current generators is that they are vertically and horizontally symmetric (without directions), which causes the graphical structure to always have a cup-shaped fragment of a real scaffold at the top as in Fig. 17a. DNA origami does not have such fragments at the border of the structure, which leads us to revise generators to define such borders. Figure 17b proposes four different generators that are used to substitute α_1 . In these generators, we introduce asymmetric structures that can be used to construct borders of the structure. We may define generators for β similarly. Under the assumption that we use the same concatenation procedure, for a graphical structure that corresponds to α_1 , we can make an arbitrary number of scaffolds and staples virtual by concatenation of four new generators as in Fig. 17c. Now, suppose we define the rewriting system based on equivalence under such generators. For each pair of diagrams of \mathcal{J}_n , we have $2n$ staples and scaffolds which can become virtual. From analysis similar to the proof of Theorem 5, the size of the set of equivalence classes becomes $\left(\left(\frac{1}{n+1} \binom{2n}{n} \right)^2 - 1 \right) \cdot 2^{2n} + 1$. The set of rewriting rules that are sufficient to describe equivalence under such generators is open.

The second observation on the current generators is that we do not consider which side of the scaffold the staple is on. In the DNA origami structure, staples can be on the left

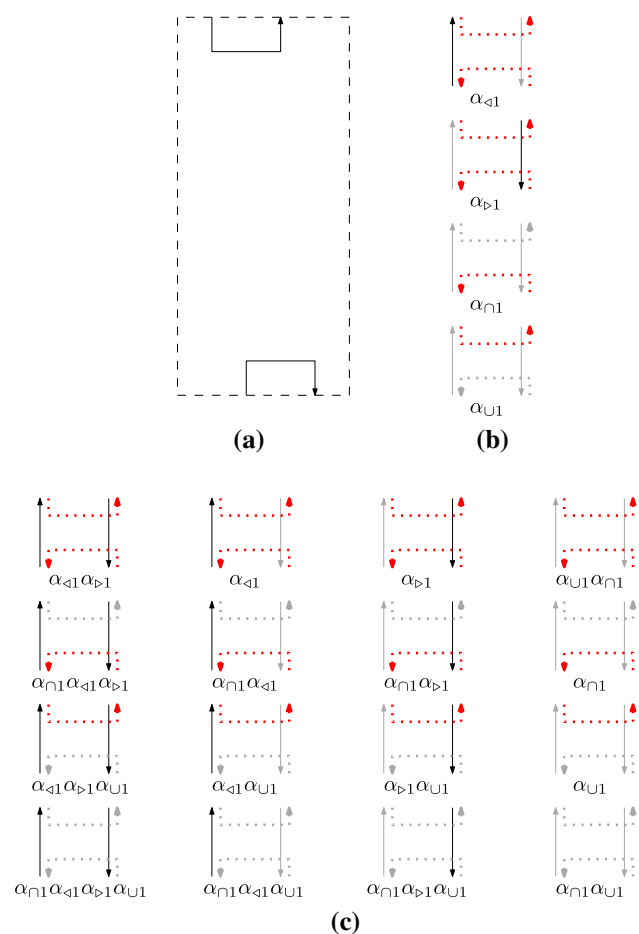


Fig. 17 **a** In a graphical structure generated from α_i 's and β_i 's, there always exists a cup-shaped fragment of a real scaffold at the top (and cap-shaped fragment at the bottom). **b** Four revised generators that substitutes α_1 . Virtual scaffolds and staples are colored in gray. **c** We can make arbitrary staples and scaffolds in α_1 virtual. (Color figure online)

or the right of the scaffold, and these two cases are distinguished. Moreover, for two adjacent staples at the opposite side of the same scaffold, they either disconnect or connect by crossing the scaffold. To model this observation, we may introduce revised graphical structures for α and β as in Fig. 18a. Staples are either at the left or the right of the scaffold, and some staple ends are extending which can be connected to other staples regardless of the side. We assume that two adjacent staples can be connected except when two are non-extending ends and at the opposite side. This additional condition for staple connection changes some of the commutation rewriting rules—for example, $\alpha_1\beta_1 \leftrightarrow \beta_1\alpha_1$ as in Fig. 18b. Algebraic analysis on relations based on such generators is done by Garrett et al. (2019). The set of rewriting rules that are sufficient to describe equivalence under such generators is still open.

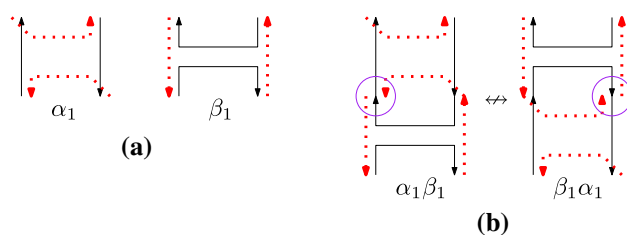


Fig. 18 **a** Revised generators α_1 and β_1 . Two diagonal ends of staples in α_1 represent extending staple-ends. **b** Adjacent staples can be connected when two are non-extending ends and at the opposite side

Acknowledgements This research was (partially) supported by the Grants NSF DMS-1800443/1764366 and the Southeast Center for Mathematics and Biology, an NSF-Simons Research Center for Mathematics of Complex Biological Systems, under National Science Foundation Grant No. DMS-1764406, Simons Foundation Grant No.594594, and Incheon National University (International Cooperative) Research Grant in 2020.

References

Bhuvana T, Smith KC, Fisher TS, Kulkarni GU (2009) Self-assembled CNT circuits with ohmic contacts using Pd hexadecanethiolate as in situ solder. *Nanoscale* 1(2):271–275

Book RV, Otto F (1993) *String-rewriting systems*. Springer, Berlin

Borisavljević M, Došen K, Petric Z (2002) Kauffman monoids. *J Knot Theory Ramif* 11(2):127–143

Dolinka I, East J (2017) The idempotent-generated subsemigroup of the Kauffman monoid. *Glasgow Math J* 59(3):673–683

Eichen Y, Braun E, Sivan U, Ben-Yoseph G (1998) Self-assembly of nanoelectronic components and circuits using biological templates. *Acta Polymerica* 49(10–11):663–670

Garrett J, Jonoska N, Kim H, Saito M (2019) Algebraic systems motivated by DNA origami. In: *Proceedings of the 8th international conference on algebraic informatics*, pp 164–176

Garrett J, Jonoska N, Kim H, Saito M (2019) DNA origami words and rewriting systems. In: *Proceedings of the 18th international conference on unconventional computation and natural computation*, vol 11493, pp 94–107

Jones VFR (1983) Index for subfactors. *Inventiones Mathematicae* 72:1–25

Kauffman LH (2001) *Knots and physics*. World Scientific, Singapore

Lau KW, FitzGerald DG (2006) Ideal structure of the Kauffman and related monoids. *Commun Algebra* 34(7):2617–2629

Li J, Fan C, Pei H, Shi J, Huang Q (2013) Smart drug delivery nanocarriers with self-assembled DNA nanostructures. *Adv Mater* 25(32):4386–4396

Rothmund PWK (2005) Design of DNA origami. In: *Proceedings of 2005 international conference on computer-aided design*, pp 471–478

Rothmund PWK (2006) Folding DNA to create nanoscale shapes and patterns. *Nature* 440(7082):297–302

The on-line encyclopedia of integer sequences. <https://oeis.org/>

Veneziano R, Ratanalert S, Zhang K, Zhang F, Yan H, Chiu W, Bathe M (2016) Designer nanoscale DNA assemblies programmed from the top down. *Science* 352(6293):1534

Verma G, Hassan PA (2013) Self assembled materials: design strategies and drug delivery perspectives. *Phys Chem Chem Phys* 15(40):17016–17028

- Whitesides GM, Boncheva M (2002) Beyond molecules: self-assembly of mesoscopic and macroscopic components. *Proc Natl Acad Sci U S A* 99(8):4769–4774
- Winfrey E, Eng T, Rozenberg, G (2001) String tile models for DNA computing by self-assembly. In: *Proceedings of the 6th international workshop on DNA-based computers*, pp 63–88

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.