The Algebra of Gene Assembly in Ciliates

DOI: 10.1007/978-3-642-40193-0_13
Handle: http://hdl.handle.net/1942/18454

The Algebra of Gene Assembly in Ciliates

Robert Brijder and Hendrik Jan Hoogeboom

Abstract The formal theory of intramolecular gene assembly in ciliates is fitted into the well-established theories of Euler circuits in 4-regular graphs, principal pivot transformations, and delta-matroids.

1 Introduction

Gene assembly is an intricate process occurring in unicellular organisms called ciliates. During this process a nucleus, called the *micronucleus*, is transformed into a functionally and structurally different nucleus called the *macronucleus*. It is accomplished using involved DNA splicing and recombination operations. Gene assembly has been formally studied on the level of individual genes, see, e.g., [20] and [10].

The theory of Euler circuits in 4-regular graphs was initiated in a seminal paper by Kotzig [31]. Bouchet further developed the theory by relating it to delta-matroids [6] and isotropic systems [5, 7]. In [6], Bouchet uses a matrix transformation that turns out to be "almost" principal pivot transform (PPT) [42]. PPT, delta-matroids, and isotropic systems enjoy many interesting properties which have direct consequences for the theory of Euler circuits in 4-regular graphs.

Although, at first glance, the formal theory of gene assembly seems to be related to the theory of Euler circuits in 4-regular graphs (this will become clear when we consider Fig. 5 in Section 3), there have been little attempts to fit the former theory into the latter. In this paper we do exactly this. We show that the formal model of gene assembly can be defined quite efficiently in terms of 4-regular graphs. In

Robert Brijder
Hasselt University and Transnational University of Limburg, Belgium, e-mail: robert.brijder@uhasselt.be

Hendrik Jan Hoogeboom
Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands, e-mail: hoogeboom@liacs.nl

a survey-style fashion we discuss consequences of known results in the theory of 4-regular graphs (including, e.g., results related to PPT and delta-matroids) for the theory of gene assembly.

This paper is organized as follows. In Section 2 we briefly recall the biology of gene assembly in ciliates and its string model [19, 23] (see also [20]). In Section 3 we view gene assembly in terms of Euler circuit transformations (or, more generally, circuit partition transformations) in 4-regular graphs, and in terms of the corresponding local and edge complementation transformations on looped circle graphs. These operations of local and edge complementation turn out to be special cases of principal pivot transform defined on arbitrary square matrices, cf. Section 4. In Section 5, we find that we can view local and edge complementation in terms of a very elementary operation, called pivot, on set systems. Finally, in Section 6, we combine pivot on set systems with another operation, called loop complementation, on set systems. Together the two operations turn out to form a group, which enables us to replace the intricate graph operations by a simple algebra on set systems.

2 Gene Assembly in Ciliates

Ciliates contain two different kinds of nuclei, which differ both functionally and structurally. The relatively large *macronucleus* (MAC for short) has many copies of short chromosomes, each containing only a single or just a few genes. The *micronucleus* (MIC for short) contains a much smaller number of chromosomes, each containing numerous genes (as is usual for chromosomes in general). The germ-line MIC is only used for reproduction, while the somatic MAC is used for general cell regulation. During sexual reproduction, a newly formed MIC is transformed into a MAC. This process is called *gene assembly* and is accomplished using extensive DNA splicing and recombination operations.

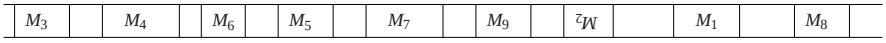

Fig. 1 The structure of the MIC gene encoding for the Actin protein in Sterkiella nova.

The genetic material in the MIC is scrambled: the genes are broken up into segments, called *macronuclear destined sequences* (or MDSs for short), which are reordered and possibly inverted with respect to the corresponding MAC genes. Moreover, the MDSs in the MIC genes are separated by *internal eliminated sequences* (IESs for short) which are not part of the genes. For example, the MIC form of the Actin I gene of the ciliate Sterkiella nova is depicted in Fig. 1 and can be described as the string $I_0 M_3 I_1 M_4 I_2 M_6 I_3 M_5 I_4 M_7 I_5 M_9 I_6 \overline{M}_2 I_7 M_1 I_8 M_8 I_9$ [39], where the M_i's are MDSs and the I_i's are IESs. Note that the inversion of the MDS M_2 is indicated by a bar. The MDSs $M_1, \ldots, M_9$ are oriented and numbered according to

the order in which they occur in the corresponding "unscrambled" MAC gene, see Fig. 2. Note that consecutive MDSs overlap (the gray segments in Fig. 2). These segments are called *pointers* in the MIC gene as they indicate the complex recombination schema that is to be performed to obtain the corresponding MAC gene.

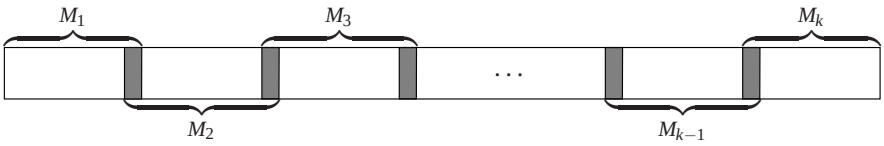

Fig. 2 The structure of a MAC gene consisting of κ MDSs.

The generic form of recombination aligns two MDSs on their common pointer, and then performs a crossover operation at that pointer, see Fig. 3. In that way the two segments are joined into a larger MDS segment. Note that the order (or the level of parallelism) in which recombination operations are applied has no influence on the outcome. This general *confluency* property of recombination ensures that the MAC gene is uniquely obtained from MIC gene by performing recombination on each pointer pair (regardless of the order in which recombination takes place) [22]. Biologically, the pointer pair no longer exists after recombination as it cannot be used for another recombination operation. Mathematically, it turns out to be worthwhile to leave the pointer pair for further consideration.

We consider the intramolecular model for gene assembly from [38, 20]. In this model three specific types of recombination operations are distinguished, see Fig. 4: (a) Two consecutive MDSs (i.e., a single IES separates the two MDSs) having the same orientation can be recombined by *loop excision*. In that process a circular molecule is removed from the segment containing that IES. (b) Two MDSs in opposite orientation can be recombined by *hairpin recombination*. This operation inverts the segment originally between the two MDSs. This segment may contain other MDSs. (c) Two interleaved pairs of consecutive MDSs, the MDSs in each pair in the same orientation, can be recombined by *double loop recombination*. During this operation two segments between the MDSs are swapped. A sequence φ of recombination operations (of these types) is called *successful* for a given MIC gene g if (i) φ is applicable to (defined on) g and (ii) applying φ on g yields the MAC gene corresponding to g. Because of the above mentioned confluence property of

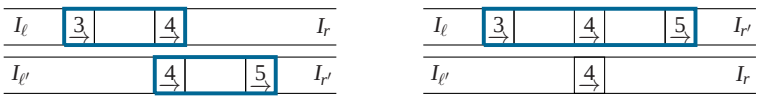

Fig. 3 Recombination on pointer 4 joins MDSs M_3 and M_4. The left and right pointers of M_3 are denoted by 3 and 4, respectively (and similarly for M_4).

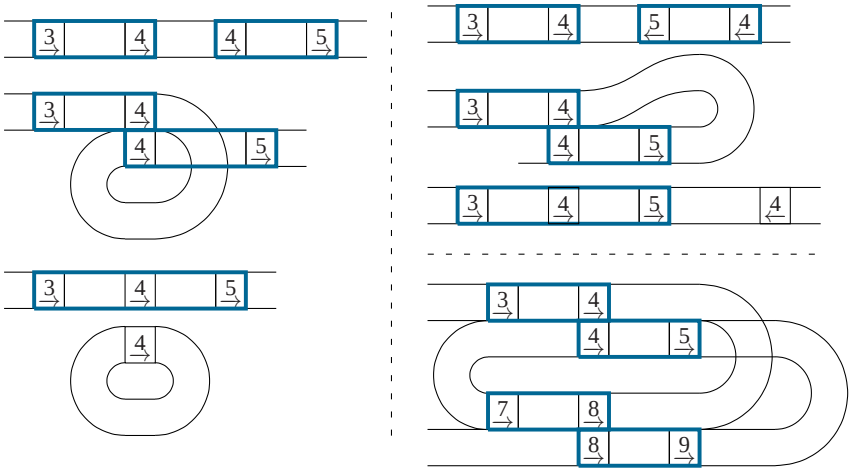

Fig. 4 Three operations: loop excision (*left-hand side*), hairpin recombination (*upper-right corner*), and double loop recombination (intermediate stage only, *lower-right corner*)

recombination in general, we have φ is successful for g iff φ is applicable to g and each pointer pair is used exactly once in φ.

The above recombination operations are formalized on strings as follows. Let us fix a positive integer κ. We denote pointers, and their orientation, by the alphabet $\Pi = \{1,2,\ldots,\kappa\} \cup \{\overline{1},\overline{2},\ldots,\overline{\kappa}\}$. The inversion of string $w = w_1 w_2 \ldots w_n \in \Pi^*$ is the string $\overline{w} = \overline{w_n} \ldots \overline{w_2}\,\overline{w_1}$, where we let $\overline{\overline{p}} = p$ for each $p \in \Pi$.

A *directed double occurrence string*, or *doc-string* for short, (called legal string in [20]) is a string w over Π that contains each pointer of w exactly twice, in either orientation (barred or unbarred). The MIC gene is then encoded by concatenating the pointers (including their orientation) in the same order as they appear in the MIC gene. Hence if there are κ MDSs, then MDS M_i (for $i \in \{2,\ldots,\kappa-1\}$) corresponds to $i\,(i+1)$, its inversion $\overline{M_i}$ corresponds to $(\overline{i+1})\,\overline{i}$, MDSs M_1 and M_κ correspond to 2 and κ, respectively, and their inversions correspond to $\overline{2}$ and $\overline{\kappa}$, respectively (recall that M_1 and M_κ have only one neighbouring MDS). Note that there is no pointer 1. Thus the MIC form of Actin I of Sterkiella nova mentioned above is written as 34 45 67 56 78 9 $\overline{32}$ 2 89, space added for clarity.

The three recombination operations can be described using doc-strings in the following straightforward manner [19, 23]. First we define the following three mappings on doc-strings. Let $u_1,\ldots,u_5 \in \Pi^*$, and let $p,q \in \{1,\ldots,\kappa\}$. Then, $u \setminus p$ deletes occurrences of p and $\overline{p}$ in u; if $u = u_1\,p\,u_2\,\overline{p}\,u_3$ then $u * p = u_1\,p\,\overline{u_2}\,\overline{p}\,u_3$; and if $u = u_1\,p\,u_2\,q\,u_3\,p\,u_4\,q\,u_5$ then $u * \{p,q\} = u_1\,p\,u_4\,q\,u_3\,p\,u_2\,q\,u_5$. In a similar way we define $u * p$ in case $u = u_1\,\overline{p}\,u_2\,p\,u_3$ (i.e., the bars on the two occurrences of p are swapped), and $u * \{p,q\}$ in case the positions of p and q are interchanged and/or in case the two occurrences in the p-pair or q-pair are barred.

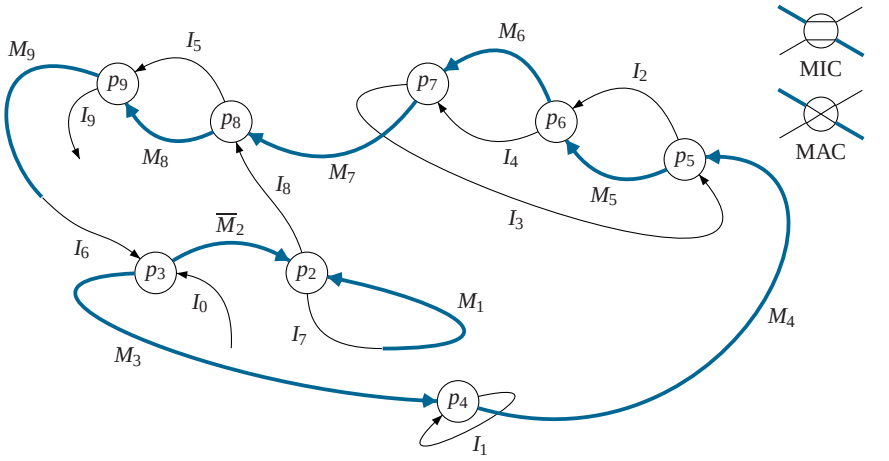

Fig. 5 Actin I gene of Sterkiella nova. Schematic diagram, based on [37]

Then (a) $u \setminus p$ models loop excision, provided u contains pp (or $\overline{p}\overline{p}$) as consecutive pointers, (b) $u * p \setminus p$ models hairpin recombination, and (c) $u * \{p,q\} \setminus p \setminus q$ models double loop recombination.

Given a doc-string, any sequence of these three operations that reduces this string into the empty string is called a *successful reduction*. Note that if a doc-string w represents a MIC gene g, then successful reductions of w correspond precisely to successful reductions of g. It is easily verified that every doc-string has a successful reduction [21]. This reduction usually is not unique.

3 Graph Models

It is not surprising that graph-theoretical concepts are important tools in modelling and understanding the process of gene assembly in ciliates. Consider for instance the diagram of the Actin I gene of Sterkiella nova as depicted by Prescott [37]. A simplified representation is given in Fig. 5 (see Example 1 below for details). The structure of the genetic material is given as a "bi-coloured" graph, with pointers as vertices, and MDS and IES segments as edges. We can read both the original MIC sequence and the target MAC sequence from the graph. If we follow the IES and MDS edges in an alternating fashion, then we obtain the MIC, and if we follow the edges according their colours, then we obtain the MAC (with flanking IESs).

Example 1. In the MDS-IES description of the MIC form of Actin I of Sterkiella nova (from Section 2), we explicitly add the pointers flanking the MDS's, to obtain the sequence $\pi = I_0\, p_3\, M_3\, p_4\, I_1\, p_4\, M_4\, p_5\, I_2\, p_6\, M_6\, p_7\, I_3\, p_5\, M_5\, p_6\, I_4\, p_7\, M_7\, p_8\, I_5\, p_9\, M_9\, I_6\, \overline{p}_3\, \overline{M}_2\, \overline{p}_2\, I_7\, M_1\, p_2\, I_8\, p_8\, M_8\, p_9\, I_9$.

One may view π as an Eulerian path in a multigraph G: the pointers p_i are the vertices of G, and the strings in-between the vertices are the (labelled) edges of G. In this way π induces Fig. 5. Apart from the MDS edges M_i (in-between p_i and p_{i+1}) and IES edges I_i, there are also "mixed" edges, like the loop $I_7 M_1$ on p_2, caused by the fact that M_1 has no initial pointer. Note that we may have parallel edges, e.g., there are two edges from p_5 to p_6.

The MAC form is obtained by recombining MDSs at each pointer. For example, at p_5 in π we have both $M_4\,p_5\,I_2$ and $I_3\,p_5\,M_5$, whereas in the MAC form M_4 and M_5 are joined at vertex (pointer) p_5, and we have both $M_4\,p_5\,M_5$ and $I_3\,p_5\,I_2$. Recall from Fig. 3 that when recombining MDSs at a pointer, IESs are at the same time joined together at that pointer.

As MDS M_2 is inverted in the MIC form, it has to be read "backwards" in the MAC form. Thus, in the MAC form we follow edge $p_3\,\overline{M}_2\,p_2$ in opposite direction. Also, when recombining M_1 and M_2 at p_2 at the same time we join I_7 and $\overline{I}_8$ at the same vertex.

In this way, the MAC form of the gene consists of three molecules. The string (the pointers are omitted) $\overline{I}_9\overline{I}_5\overline{I}_8\overline{I}_7 M_1 M_2 \ldots M_8 M_9 I_6 \overline{I}_0$ represents the strand consisting of the recombined MDS's and flanking IES's. The MAC form also has two circular IES molecules (that are excised): I_1 and $I_2\,I_4\,I_3$ (again the pointers are omitted). □

In the manner described above, every (unbarred) doc-string defines a 4-regular multigraph (every vertex has degree 4, we allow loops and parallel edges) together with an Euler cycle (that visits every edge of the graph exactly once). Start by representing every pointer pair by a vertex. Then follow the string, adding an edge as we step from pointer to pointer. Treat the string as if circular, and connect the last pointer to the first. Obviously, the multigraph is 4-regular, and the string traces an Euler cycle through the multigraph. The result is very similar to the representation of the gene in Fig. 5, if we merge the initial and final "edges" I_0 and I_9. Conversely, every 4-regular multigraph with Euler cycle induces a (unbarred) doc-string w (in fact, a set of "equivalent" doc-strings that are obtained from w by conjugation).

We briefly describe the theory of operations on cycles in 4-regular multigraphs as initiated by Kotzig [31] and continued by Bouchet. Given a 4-regular multigraph we obtain a set of cycles by pairwise "joining" the edges at each vertex. These pairings can be unambiguously described using a fixed Euler circuit as anchor, see Fig. 6 (a–c). The pairings may follow the Euler circuit, and otherwise reconnect in a way

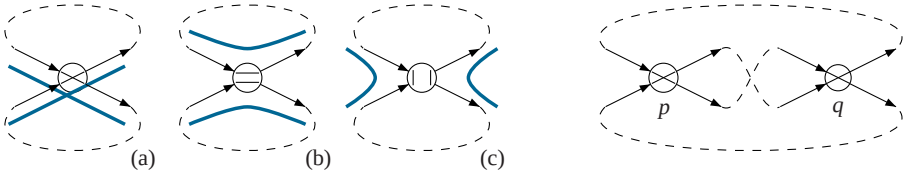

Fig. 6 Three ways to connect pairs of edges in a 4-regular graph relative to an Eulerian cycle (a) following the cycle (b) orientation consistent (c) orientation inconsistent. Two interleaved vertices in the cycle. (*rightmost diagram*)

that may or may not agree with the orientation of the Euler circuit. The pairings of Fig. 6 (a–c) are called smoothings in [3] and transitions in [6].

Example 2. (1) Consider the (unbarred) doc-string $w = 1\mathbf{2}6134563\mathbf{2}45$. Then w defines a 4-regular graph G along with an Euler circuit E_w in G, see Fig. 7(a). For convenience, the edges in G are directed according to w. (2) If we take the pairing at vertex 2 that is in an orientation-consistent way different from E_w, and if we take the same the pairings as E_w at the other vertices, then we obtain two disjoint cycles, $w_a = 1\mathbf{2}45$ and $w_b = \mathbf{2}6134563$, see Fig. 7(b). (3) If we take the pairing at vertex 2 that is in an orientation-inconsistent way different from E_w, and if we take the same the pairings as E_w at the other vertices, then we obtain the Euler circuit described by $w' = 1\mathbf{2}3654316\mathbf{2}45$, see Fig. 7(c). □

Note how an orientation-inconsistent transition induces a "reversal" of part of the original Euler circuit, which agrees with hairpin recombination. Also note that an orientation-consistent transition breaks the Euler circuit into two disconnected parts. Consider now two vertices p and q that are interleaved in the Euler circuit, occurring in the order $\cdots p \cdots q \cdots p \cdots q \cdots$, see Fig 6 (right). Then a synchronized orientation-consistent transition at both p and q again yields an Euler circuit. This Euler circuit is obtained from the original one by swapping two segments in exactly the same way as double loop recombination.

Then, to correctly model the gene assembly process, we have to keep track on which pointers (vertices) we can apply the successive transition while maintaining an Euler circuit. Both the orientation and interleavings may change during the process. The tool we use is that of a *circle graph*, which represents the intersections of the chords in a circle: each chord is represented by a vertex and two vertices are adjacent iff the corresponding chords intersect. A (barred) doc-string w defines a circle graph $C(w)$ in a natural way by writing w in a circular way and connecting the pointer pairs. We additionally encode the relative orientation of the pointers of each pointer pair by adding a loop to a vertex when the two pointers of the pointer pair have different orientation (i.e., one is with a bar, and the other is without a bar). In [20] $\pm$-signs are used instead of loops, and the corresponding graph, equivalent to a (looped) circle graph, is called a *signed graph*. The advantage of using loops instead of $\pm$-signs will become clear in Section 4. The (looped) circle graph $C(w)$ of the doc-string $w = 126\bar{1}345\bar{6}3\bar{2}45$ is given in Fig. 9 (middle, bottom row).

From now on, by *graph* we mean an undirected graph G where loops are allowed, but parallel edges are not allowed. More precisely, $G = (V, E)$ where V is a finite

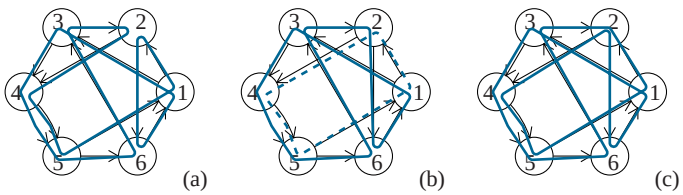

Fig. 7 Recombining edges of an Euler cycle, cf. Example 2.

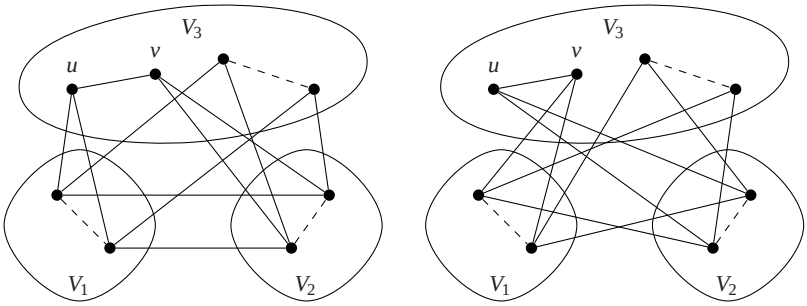

Fig. 8 Pivot on an edge $\{u,v\}$ in a graph. Adjacency between vertices x and y is toggled iff $x \in V_i$ and $y \in V_j$ with $i \neq j$. Note that u and v are adjacent to all vertices in V_3 — these edges are omitted in the diagram. The operation does not affect edges adjacent to vertices outside the sets V_1, V_2, V_3.

set of *vertices* and $E \subseteq \{\{x,y\} \mid x,y \in V\}$ a set of *edges* (we have $\{x\} \in E$ iff x is a looped vertex). If the graph G is clear from the context of considerations, then we simply denote its vertex set by V. For $X \subseteq V$, we denote the subgraph of G induced by X as $G[X]$.

We define the basic operations of local and edge complementation on graphs [31, 7]. If G is a graph with looped vertex u, then the *local complement* of G on u, denoted by $G * u$, is obtained from G by complementing the edges in the neighbourhood $N_G(u) = \{v \in V \mid \{u,v\} \in E, u \neq v\}$ of u in G. Thus, for $v,w \in N_G(u)$, $e = \{v,w\}$ is an edge of $G * u$ iff e is not an edge of $G * u$ (we allow $v = w$, i.e., that e is a loop). All other edges remain the same in G and $G * u$.

For an edge $\{u,v\}$ of G with u and v distinct unlooped vertices, one defines the *edge complement* of G on $\{u,v\}$, denoted by $G * \{u,v\}$, as follows. For vertex w its closed neighbourhood, denoted by $N_G[w]$, equals $N_G(w) \cup \{w\}$. The neighbours of u and v can be partitioned in the three sets $N_G[u] \setminus N_G[v]$, $N_G[v] \setminus N_G[u]$ and $N_G[u] \cap N_G[v]$. The graph $G * \{u,v\}$ is obtained from G by complementing all pairs $\{x,y\}$ such that x and y are each neighbours of u or v, but not in the same partition, cf. Fig. 8. This will not change any adjacencies to vertices not adjacent to u and v, nor will it change any loops.

Theorem 1 ([31]). *Let w be a doc-string and let $p,q \in \{1,\ldots,\kappa\}$. If $w * p$ is defined (i.e., w contains both p and $\overline{p}$), then $C(w) * p = C(w * p)$. If $w * \{p,q\}$ is defined, then $C(w) * \{p,q\} = C(w * \{p,q\})$.*

Of course, Theorem 1 may be reformulated using Euler cycles in a 4-regular multigraphs instead of doc-strings.

Example 3. Consider the doc-string $w = 12\overline{6}1\overline{3}4\overline{5}\overline{6}3\overline{2}45$. It defines the circle graph $C(w)$ in Fig. 9 (middle). If we complement the neighbourhood $N_{C(w)}(2) = \{1,4,5\}$ of vertex 2 in $C(w)$ we obtain the graph $C(w) * \{2\} = C(1\mathbf{2}\overline{3}\overline{6}5\overline{4}\overline{3}16\mathbf{\overline{2}}45)$, see Fig. 9 (left). Edge complement on unlooped edge $\{3,4\}$ in $C(w)$ yields $C(w) * \{3,4\} = 12\overline{6}1\mathbf{3}2\mathbf{4}5\overline{6}3\mathbf{4}5$ depicted in Fig. 9 (right). □

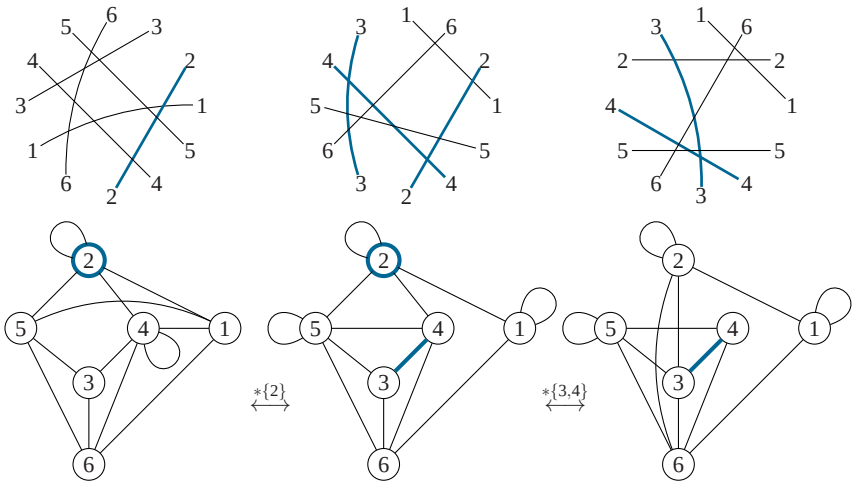

Fig. 9 Local complement on looped vertex 2 (*left-hand side*); edge complement on unlooped edge $\{3,4\}$ (*right-hand side*). The top row indicates how the pointer segments overlap in the underlying doc-strings, the bottom row contains circle graphs.

Theorem 1 suggests a generalization of the three recombination operations on doc-strings that model loop excision, hairpin recombination, and double loop recombination (from Section 2). We have that (a) removing an isolated unlooped vertex corresponds to loop excision, (b) local complementation followed by the deletion of the vertex involved corresponds to hairpin recombination, and (c) edge complementation followed by the deletion of the vertices involved corresponds to double loop recombination.

A *reduction* of a graph G is a sequence of these three operations, and a *successful reduction* of G is a reduction of G to the empty graph. Every graph has a successful reduction: indeed, apply local complement reductions until there are no more loops, then apply edge complement reductions until the graph contains only isolated unlooped vertices, and finally remove these isolated unlooped vertices.

If a doc-string w can be successfully reduced by a sequence of operations, then the associated circle graph $C(w)$ can be rewritten by the corresponding sequence of graph operations. For the converse a similar result holds, except that we may have to reorder the loop excision operations [19, 23]. For, e.g., the string $w = 2332$ there is a unique sequence of two loop excisions ("inside out"), while its circle graph $C(w)$ consists of two isolated unlooped vertices which can be removed in any order.

Since not every graph is a circle graph $C(w)$ for some doc-string w, the operations of local complementation $*\,p$ and edge complementation $*\,\{p,q\}$ are generalizations of the corresponding operations $*\,p$ and $*\,\{p,q\}$, respectively, for doc-strings.

We remark that a polynomial called the *Martin polynomial* [32] (its multivariate variant is called the *transition polynomial* [30]) has been defined w.r.t. Euler circuits C in 4-regular multigraphs. This polynomial records the number of circuits $c_T(C)$

obtained when performing on C a set T of transitions of the form described as in Fig. 6. In a similar way as described in this section, the Martin polynomial corresponds to a graph polynomial called the *interlace polynomial* [4] (or *Tutte-Martin polynomial* [8]) in which local and edge complementation play a central role. Interestingly, the well-known *Tutte polynomial* on the diagonal coincides with the interlace polynomial when restricting to bipartite graphs [2]. A number of variations of the Martin and interlace polynomials is studied in the literature as we may restrict or loosen the allowed types of transitions T. Among them is the *Penrose polynomial* [35, 1] and the *bracket polynomial* for graphs [41]. For a detailed survey on these polynomials we refer to [24, 25].

4 Matrices

With the definitions of local complementation $*u$ and edge complementation $*\{u,v\}$ in place, we are interested in *sequences* of these operations (and in particular successful reductions). Since the definition of edge complementation is already involved by itself, it seems even more difficult to reason about the effect of sequences such as $*\{u,v\}*\{v,w\}$. Fortunately, it turns out that sequences of local and edge complementation correspond to (a special case of) the so-called principal pivot transform operation on square matrices. This will lead to a different perspective in which sequences of local and edge complementations are much easier to study. Let us first recall the principal pivot transform operation.

Let V be a finite set, and let A be a $V \times V$ matrix, i.e., a matrix where the columns and rows are indexed by V. For a set $X \subseteq V$ we use $A[X]$ to denote the principal submatrix induced by X (i.e., the rows and columns are indexed by X). Moreover, we define $A \setminus X = A[V \setminus X]$. Let A be a $V \times V$-matrix (over an arbitrary field), and let $X \subseteq V$ be such that $A[X]$ is nonsingular, i.e., $\det A[X] \neq 0$. The *principal pivot transform* (*PPT* or *pivot* for short) of A on X, denoted by $A*X$, is defined as follows, see [43]. If $A = \begin{array}{c} \\ X \\ V \setminus X \end{array} \begin{array}{c} \begin{array}{cc} X & V \setminus X \end{array} \\ \begin{pmatrix} P & Q \\ R & S \end{pmatrix} \end{array}$, then

$$A*X = \begin{array}{c} \\ X \\ V \setminus X \end{array} \begin{array}{c} \begin{array}{cc} X & V \setminus X \end{array} \\ \begin{pmatrix} P^{-1} & -P^{-1}Q \\ RP^{-1} & S - RP^{-1}Q \end{pmatrix} \end{array}.$$

Hence, $A*X$ is defined iff $A[X]$ is nonsingular. Matrix $(A*X) \setminus X = S - RP^{-1}Q$ is called the *Schur complement* of X in A.

The pivot is sometimes considered a partial inverse, since A and $A*X$ are related as follows, where the vectors x_1 and x_2 correspond to the elements of X. In fact, the following relation defines $A*X$ given A and X [42].

$$A\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \text{ iff } A*X\begin{pmatrix} x_2 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_2 \end{pmatrix} \quad (1)$$

Note that if $\det A \neq 0$, then $A*V = A^{-1}$. By Equation (1) we see that a pivot operation is an involution (i.e., operation of order 2), and more generally, if $(A*X)*Y$ is defined, then $A*(X\,\Delta\,Y)$ is defined (applying the symmetric difference of X and Y) and the resulting matrices are equal. Note that in order to apply the pivot $*X$ to matrix A it is required that $A[X]$ is nonsingular.

We may apply pivot to graphs through its adjacency matrix representation. The adjacency matrix $A(G)$ of a graph $G=(V,E)$ is a $V\times V$ matrix $(a_{u,v})$ over $\mathbb{F}_2$ (the binary field) with $a_{u,v}=1$ iff $\{u,v\}\in E$. Obviously, for $X\subseteq V$, $A(G[X])=A(G)[X]$. In this paper we make no distinction between G and $A(G)$ and so we write, e.g., $\det G$ to denote $\det A(G)$, the determinant of $A(G)$ computed over $\mathbb{F}_2$. In this way, graphs correspond precisely to symmetric $V\times V$-matrices over $\mathbb{F}_2$. By convention, the determinant of the empty matrix (or graph) is 1.

For a graph G and nonempty $X\subseteq V$, X is called *elementary in* G if $G[X]$ is nonsingular and for all nonempty $Y\subsetneq X$, $G[Y]$ is singular. Hence, if X is elementary in G, then $G*X$ is defined but $G*Y$ is not defined for any nonempty proper subset of X. It is easy to see that if X is elementary in G, either (1) $X=\{u\}\in E(G)$ (i.e., X is a loop) or (2) $X=\{u,v\}\in E(G)$ and $\{u\},\{v\}\notin E(G)$ (i.e., X is an edge on unlooped vertices). Geelen [26] observed that a pivot of Case (1) is precisely local complementation and a pivot of Case (2) is precisely edge complementation.

Indeed, if vertex u has a loop in G, then the matrix $G[\{u\}]$ is equal to the 1×1 identity matrix: $\begin{matrix} u \end{matrix}\begin{pmatrix} 1 \end{pmatrix}$. Hence, $*\{u\}$ is indeed applicable to G and

$$G*\{u\} = \begin{matrix} \\ u \\ V\setminus\{u\} \end{matrix}\begin{matrix} \begin{matrix} u & V\setminus\{u\} \end{matrix} \\ \begin{pmatrix} 1 & \chi_u^T \\ \chi_u & G[V-u]-\chi_u\chi_u^T \end{pmatrix} \end{matrix},$$

where χ_u is the column vector belonging to u without element at position (u,u). One may easily verify that $G*\{u\}$ is indeed the graph obtained from G by applying local complementation on u.

Turning to edge complementation, if $\{u,v\}$ is an edge in G and u and v are not looped vertices, then the matrix $G[\{u,v\}]$ is equal to $\begin{matrix} \\ u \\ v \end{matrix}\begin{matrix} \begin{matrix} u & v \end{matrix} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix}$. Hence, $*\{u,v\}$ is indeed applicable to G and

$$G*\{u,v\} = \begin{matrix} \\ u \\ v \\ V\setminus\{u,v\} \end{matrix}\begin{matrix} \begin{matrix} u & v & V\setminus\{u,v\} \end{matrix} \\ \begin{pmatrix} 0 & 1 & \chi_v^T \\ 1 & 0 & \chi_u^T \\ \chi_v & \chi_u & G[V-u-v]-(\chi_v\chi_u^T+\chi_u\chi_v^T) \end{pmatrix} \end{matrix}$$

where χ_u is the column vector of G belonging to u without elements at positions (u,u) and (v,u) (and similarly for χ_v). One may again verify that $G*\{u,v\}$ is indeed the graph obtained from G by applying edge complementation on $\{u,v\}$.

Having thus characterized local and edge complementation in terms of pivot, we are ready to study sequences of these operations. For example, let u,v, and w be mutually distinct unlooped vertices of G. If $(G*\{u,v\})*\{v,w\}$ is defined (i.e., if $\{u,v\}$ is an edge of G, and $\{v,w\}$ is an edge of $G*\{u,v\}$), then we immediately find that $(G*\{u,v\})*\{v,w\} = G*(\{u,v\}\,\Delta\,\{v,w\}) = G*\{u,w\}$ (and that $\{u,w\}$ is an edge of G since $G*\{u,w\}$ is defined and u and w are unlooped vertices). In general we have the following *confluence* result.

Let $\varphi = *X_1*X_2\cdots*X_n$ be a sequence of pivot operations (we assume left-associativity of pivot). The *support* of φ, denoted by $\sup(\varphi)$, is defined as $\Delta_i\, X_i$, i.e., the set of vertices that occur an odd number of times in φ. If φ is applicable to a graph G, then, by the above, $G\varphi = G*(X_1\,\Delta\,X_2\,\Delta\,\ldots\,\Delta\,X_n) = G*\sup(\varphi)$. This observation may be seen as a highly generalized version of the confluence property of DNA recombination from Section 2. If we specialize this observation for the case where the $*X_i$'s are elementary (i.e., local or edge complementations), then we obtain the following.

Theorem 2 ([11]). *If φ and φ' are applicable sequences of local and edge complementations for a graph G, then* $\sup(\varphi) = \sup(\varphi')$ *implies* $G\varphi = G\varphi'$.

Special cases of this result are mentioned in the literature. The *triangle equality* stated above, $*\{u,v\}*\{v,w\} = *\{u,w\}$ for a graph with induced loopless triangle $\{u,v,w\}$, can be found as [4, Lemma 10], [27, Proposition 1.3.5], and [33, Proposition 2.5]. A "classical" proof typically involves keeping track of numerous neighbouring edges. Also commutativity of edge complementation is obtained in the context of gene assembly by Harju et al. [29]: if two disjoint edge complementations are applicable in either order, then the two results are identical. In short, $*\{u,v\}*\{w,x\} = *\{w,x\}*\{u,v\}$.

Example 4. Consider the MDS sequence $M_4M_3\bar{M}_5M_2\bar{M}_1$. It defines the pointer sequence $4534\bar{5}23\bar{2}$ which in turn has circle graph G depicted in Fig. 10. The figure illustrates that $G*\{2\}*\{3\}*\{4\} = G*\{3,4\}*\{2\}$. □

Note also that G is nonsingular iff there is a sequence φ of local and edge complementations with $\sup(\varphi) = V$ such that $G\varphi$ is defined. Moreover, if this is the case, then we may choose φ in such a way that each vertex of V appears exactly once. In

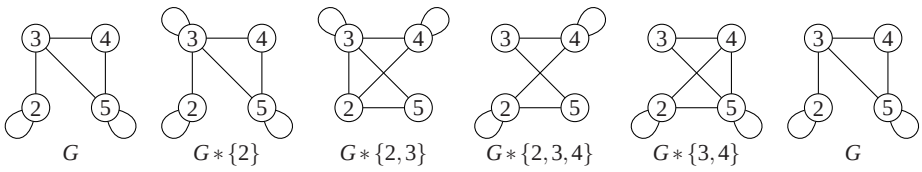

Fig. 10 Circle graph G for $4534\bar{5}23\bar{2}$ (left and right) $G*\{2,3,4\}$ is computed twice, as $G*\{2\}*\{3\}*\{4\}$, and as $G*\{3,4\}*\{2\}$ (reading from right to left).

the context of gene assembly, we thus find that there is a sequence of hairpin and double-loop recombinations to transform a MIC gene into the corresponding MAC gene iff the circle graph G corresponding to MIC gene is nonsingular. Moreover, if this is the case, then the circle graph corresponding to the MAC gene is $G*V = G^{-1}$, the inverse matrix of the adjacency matrix of G! Thus, from this point-of-view, we have the amusing fact that the construction of the MAC gene entails inverting a matrix. Intermediate products obtained during the transformation of a MIC gene into its MAC gene, using only hairpin and double-loop recombinations, correspond in this way to partially inverted matrices.

For each possible set S of operation-types (loop excision, hairpin recombination, double loop recombination) there is a characterization of the existence of a sequence of recombination operations to transform a MIC gene into the corresponding MAC gene, where each recombination operation is of a type from S, see [20, Section 13.3] when S contains loop excision, and [11, 17] for the remaining cases where only hairpin recombination and/or double loop recombination are allowed.

We remark that an extension of the interlace polynomial from graphs to arbitrary matrices (over some field), using PPT instead of local and edge complementation on graphs, has been studied in [28, 15].

5 Set Systems

In this section we provide yet another perspective on local and edge complementation. It turns out that we may define local and edge complementation (and pivot for graphs in general) in terms of a very elementary operation on set systems, essentially only involving symmetric difference.

First we recall a fundamental result on PPT due to Tucker [43] (see also [18, Theorem 4.1.1] and [34]). This result allows one to formulate the applicability of the pivot $*Y$ to the resulting matrix $A*X$ in terms of applicability of the pivot $*(X \Delta Y)$ to the original matrix A.

Proposition 1 ([43]). *Let A be a $V \times V$-matrix, and let $X \subseteq V$ be such that $A[X]$ is nonsingular. Then, for all $Y \subseteq V$, $\det(A*X)[Y] = \det A[X \Delta Y] / \det A[X]$.*

We remark here that Proposition 1 for the case $Y = V \setminus X$ is called the Schur determinant formula, $\det((A*X) \setminus X) = \det A / \det A[X]$, and was shown already in 1917 by Issai Schur, see [40].

A *set system* (over V) is an ordered pair $M = (V, D)$ with V a finite set and D a family of subsets of V. We write simply $Y \in M$ to denote $Y \in D$. Let M be a set system over V. We define, for $X \subseteq V$, the *pivot* (often called *twist* in the literature, see, e.g., [26]) $M*X = (V, D*X)$, where $D*X = \{Y \Delta X \mid Y \in D\}$.

For a $V \times V$-matrix A, we let $\mathscr{M}_A = (V, D_A)$ be the set system with $D_A = \{X \subseteq V \mid \det A[X] \neq 0\}$. As observed in [6] we have, by Proposition 1, $Z \in \mathscr{M}_{A*X}$ iff $\det((A*X)[Z]) \neq 0$ iff $\det(A[X \Delta Z]) \neq 0$ iff $X \Delta Z \in \mathscr{M}_A$ iff $Z \in \mathscr{M}_A * X$. Hence $\mathscr{M}_{A*X} = \mathscr{M}_A * X$.

Through the adjacency matrix representation of graphs, we may carry the notion of $\mathscr{M}_A$ for matrices over to graphs. Let G be a graph. Given only the set system $\mathscr{M}_G = (V, D_G)$, one can (re)construct the graph G: $\{u\}$ is a loop in G iff $\{u\} \in D_G$, and $\{u,v\}$ is an edge in G iff $(\{u,v\} \in D_G) \oplus ((\{u\} \in D_G) \wedge (\{v\} \in D_G))$ ($\oplus$ denotes exclusive or), see [9, Property 3.1]. Hence the function $\mathscr{M}_{(\cdot)}$ which assigns to each graph G its set system $\mathscr{M}_G$ is injective. In this way, the family of graphs (with set V of vertices) can be considered as a subset of the family of set systems (over set V).

As $\mathscr{M}_{G*X} = \mathscr{M}_G * X$, the pivot operation for graphs coincides with the pivot operation for set systems. Therefore, pivot on set systems forms an alternative definition of pivot on graphs. Note that while for a set system M over V, $M*X$ is defined for all $X \subseteq V$, for a graph G, $G*X$ is defined precisely when $\det G[X] = 1$, or equivalently, when $X \in D_G$, which in turn is equivalent to $\varnothing \in D_G * X$. Thus, e.g, while on a graph $*\{u\}*\{v\}$ and $*\{u,v\}$ cannot be both defined, they are on set systems, where they have the same outcome.

Example 5. Consider circle graph G from Example 4, see Fig. 10. The corresponding set system equals $\mathscr{M}_G = (V, \{\varnothing, 2, 5, 34, 23, 25, 45, 35, 234, 245, 345\})$ where $V = \{2,3,4,5\}$. Note that we abbreviate sets in this example, so, e.g., 245 denotes $\{2,4,5\}$. Then $\mathscr{M}_G * \{3\} = (V, \{3, 23, 35, 4, 2, 235, 345, 5, 24, 2345, 45\})$. This does not represent a graph (as $\varnothing$ is not a set of $\mathscr{M}_G * \{3\}$). □

It turns out that $\mathscr{M}_G$ for graphs G has a special structure, that of a *delta-matroid* [6]. A consequence of the fact that $\mathscr{M}_G$ is a delta-matroid, is that the maximal sets of $\mathscr{M}_G$ w.r.t. inclusion, denoted by $\max(\mathscr{M}_G)$, are all of cardinality equal to the rank $r(A(G))$ of matrix $A(G)$. Thus, if G is the circle graph corresponding to a MIC gene, then the nullity $n(A(G)) = |V| - r(A(G))$ of $A(G)$ (i.e., the dimension of the nullspace of $A(G)$) is equal to the number of loop recombinations in *every* successful transformation of that gene to its MAC form. Equivalently, the number of loops created during the transformation of a MIC gene to its MAC gene is equal to the nullity of the adjacency matrix of the circle graph corresponding to that MIC gene. In fact, a set $S_l \subseteq V$ is the support of the loop excision part of a successful reduction of G iff $V \setminus S_l \in \max(\mathscr{M}_G)$.

Example 6. Consider the circle graph G from Example 4 and corresponding set system $\mathscr{M}_G$, see Example 5. As G has nullity 1 the maximal sets of $\mathscr{M}_G$ are of cardinality 3. Considering the maximal sets $\{2,3,4\}$, $\{2,4,5\}$, $\{3,4,5\}$ of $\mathscr{M}_G$, we see that loop recombination can be performed on every pointer except 4. □

6 Loop Complementation

The concept of local complementation defined in this paper is only defined on looped vertices, and thus cannot be applied to simple graphs. A related concept, which is also called local complementation, is defined for each simple graph G and vertex u of G: local complementation of G on u complements the neighbourhood

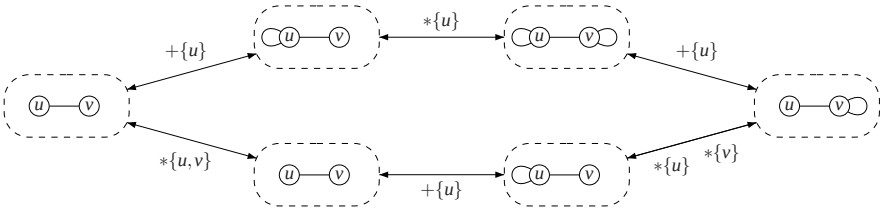

Fig. 11 Verification of applicability of $*\{u,v\}+\{u\}*\{u\}*\{v\}+\{u\}*\{u\}+\{u\}$ to any graph F having an edge $\{u,v\}$ with both u and v non-loop vertices.

of u (without introducing loops). By abuse of notation we denote also local complementation for simple graphs G by $G*\{u\}$. The operation edge complementation can be defined as for graphs with loops. In this context of simple graphs we have the "curious" identity $*\{u,v\} = *\{u\}*\{v\}*\{u\} = *\{v\}*\{u\}*\{v\}$ [7, Corollary 8.2], which is not valid for graphs with loops. In fact, in that case, the left and right side of the equation do not have the same support.

In order to deal with loops the operation loop complementation is useful. For a graph G and a set X of vertices of G, *loop complementation* of G on X, denoted by $G+X$, is the graph obtained from G by toggling the loops on the vertices in X. This operation can be faithfully represented on set systems. For set system $M = (V,D)$ and element $u \in V$, *loop complementation* of M on u, denoted by $M+u$, is the set system (V,D'), where $D' = D \,\Delta\, \{X \cup \{u\} \mid X \in D, u \notin X\}$. As the operation is commutative we extend it to $M+X$ for set $X \subseteq V$ by performing the $+u$'s, $u \in X$, in any order. We have $\mathscr{M}_{G+X} = \mathscr{M}_G + X$ for any set $X \subseteq V$.

Pivot $*X$ and loop complementation $+X$ for sets systems together form an interesting algebra. On a single common element u the operations $*u$ and $+u$ are involutions (i.e., of order 2) generating a group isomorphic to the group S_3 of permutations on 3 elements [14]. In particular, we have $+u*u+u = *u+u*u$, which is the third involution (in addition to pivot and loop complementation). On different elements $u \neq v$ the operations commute, thus, $*u+v = +v*u$, $+u+v = +v+u$, and $*u*v = *v*u$.

This algebra makes it possible to understand the relation between edge complementation and local complementation for simple graph, mentioned above. First one notes that the sequence of operations $\varphi = *\{u,v\}+\{u\}*\{u\}*\{v\}+\{u\}*\{u\}+\{u\}$ is applicable to any graph with edge $\{u,v\}$ and u and v unlooped vertices, by checking the existence of loops on u and v in successive stages, see Fig. 11. Then we observe that φ is the identity on set systems using the group structure (using the fact that, for set systems, we have $*\{u,v\} = *\{u\}*\{v\}$). This makes φ the identity on any graph where it is applicable (without having to consider the involved graph operations). We can project φ from graphs to simple graphs by skipping the loop complementation operations, and obtain the equality $*\{u,v\} = *\{u\}*\{v\}*\{u\}$.

Inspired by and motivated by the context of gene assembly, the interplay of loop complementation and pivot was implicitly (and only for the case of doc-strings) studied in [12]. This interplay led to an extension of the interlace polynomial (in-

cluding the related bracket polynomial for graphs) and an extension of the Penrose polynomial from graphs (resp. matroids) to Δ-matroids [13, 16].

7 Discussion

We have fitted the theory of gene assembly in ciliates in the theory of 4-regular graphs, and we have carried over results from the latter theory to the former. Interestingly, operations on Euler circuits in 4-regular graphs (cf. Fig. 6) occur (often implicitly) also in other topics of computational molecular biology. For instance, the monograph [36] by Pevzner has three chapters where the operations of Fig. 6 are used: Chap. 2 on restriction mapping has them under the names order exchange and order reflexion; Chap. 5 on sequencing by hybridization features rearrangements of Eulerian cycles; and Chap. 10 on genome rearrangements studies reversal in the so-called breakpoint graph. Hence we expect that these topics (and others) may benefit from a similar approach as is done in this paper; to carry over the general theory of 4-regular graphs to these topics.

References

1. Aigner, M.: The Penrose polynomial of graphs and matroids. In: Surveys in Combinatorics, 2001, *London Mathematical Society Lecture Note Series*, vol. 288, pp. 11–46. Cambridge University Press (2001). DOI 10.1017/CBO9780511721328.004
2. Aigner, M., van der Holst, H.: Interlace polynomials. Linear Algebra and its Applications **377**, 11–30 (2004). DOI 10.1016/j.laa.2003.06.010
3. Angeleska, A., Jonoska, N., Saito, M.: DNA recombination through assembly graphs. Discrete Applied Mathematics **157**(14), 3020–3037 (2009). DOI 10.1016/j.dam.2009.06.011
4. Arratia, R., Bollobás, B., Sorkin, G.: The interlace polynomial of a graph. Journal of Combinatorial Theory, Series B **92**(2), 199–233 (2004). DOI 10.1016/j.jctb.2004.03.003
5. Bouchet, A.: Isotropic systems. European Journal of Combinatorics **8**, 231–244 (1987). DOI 10.1016/S0195-6698(87)80027-6
6. Bouchet, A.: Representability of Δ-matroids. In: Proc. 6th Hungarian Colloquium of Combinatorics, Colloquia Mathematica Societatis János Bolyai, vol. 52, pp. 167–182. North-Holland (1987)
7. Bouchet, A.: Graphic presentations of isotropic systems. Journal of Combinatorial Theory, Series B **45**(1), 58–76 (1988). DOI 10.1016/0095-8956(88)90055-X
8. Bouchet, A.: Tutte-Martin polynomials and orienting vectors of isotropic systems. Graphs and Combinatorics **7**(3), 235–252 (1991). DOI 10.1007/BF01787630
9. Bouchet, A., Duchamp, A.: Representability of Δ-matroids over $GF(2)$. Linear Algebra and its Applications **146**, 67–78 (1991). DOI 10.1016/0024-3795(91)90020-W
10. Brijder, R., Daley, M., Harju, T., Jonoska, N., Petre, I., Rozenberg, G.: Computational nature of gene assembly in ciliates. In: G. Rozenberg, T. Bäck, J. Kok (eds.) Handbook of Natural Computing, vol. 3, chap. 37, pp. 1233–1280. Springer (2012). DOI 10.1007/978-3-540-92910-9_37
11. Brijder, R., Harju, T., Hoogeboom, H.: Pivots, determinants, and perfect matchings of graphs. Theoretical Computer Science **454**, 64–71 (2012). DOI 10.1016/j.tcs.2012.02.031

12. Brijder, R., Hoogeboom, H.: The fibers and range of reduction graphs in ciliates. Acta Informatica **45**, 383–402 (2008). DOI 10.1007/s00236-008-0074-3
13. Brijder, R., Hoogeboom, H.: Interlace polynomials for delta-matroids (2010). [arXiv:1010.4678]
14. Brijder, R., Hoogeboom, H.: The group structure of pivot and loop complementation on graphs and set systems. European Journal of Combinatorics **32**, 1353–1367 (2011). DOI 10.1016/j.ejc.2011.03.002
15. Brijder, R., Hoogeboom, H.: Nullity invariance for pivot and the interlace polynomial. Linear Algebra and its Applications **435**, 277–288 (2011). DOI 10.1016/j.laa.2011.01.024
16. Brijder, R., Hoogeboom, H.: Bicycle matroids and the Penrose polynomial for delta-matroids (2012). [arXiv:1210.7718]
17. Brijder, R., Hoogeboom, H.: Binary symmetric matrix inversion through local complementation. Fundamenta Informaticae **116**(1-4), 15–23 (2012). DOI 10.3233/FI-2012-664
18. Cottle, R., Pang, J.S., Stone, R.: The Linear Complementarity Problem. Academic Press, San Diego (1992)
19. Ehrenfeucht, A., Harju, T., Petre, I., Prescott, D., Rozenberg, G.: Formal systems for gene assembly in ciliates. Theoretical Computer Science **292**, 199–219 (2003). DOI 10.1016/S0304-3975(01)00223-7
20. Ehrenfeucht, A., Harju, T., Petre, I., Prescott, D., Rozenberg, G.: Computation in Living Cells – Gene Assembly in Ciliates. Springer Verlag (2004)
21. Ehrenfeucht, A., Harju, T., Petre, I., Rozenberg, G.: Characterizing the micronuclear gene patterns in ciliates. Theory of Computing Systems **35**, 501–519 (2002). DOI 10.1007/s00224-002-1043-9
22. Ehrenfeucht, A., Petre, I., Prescott, D., Rozenberg, G.: Circularity and other invariants of gene assembly in ciliates. In: M. Ito, et al. (eds.) Words, Semigroups, and Transductions, pp. 81–97. World Scientific, Singapore (2001). DOI 10.1142/9789812810908_0007
23. Ehrenfeucht, A., Petre, I., Prescott, D., Rozenberg, G.: String and graph reduction systems for gene assembly in ciliates. Mathematical Structures in Computer Science **12**, 113–134 (2002). DOI 10.1017/S0960129501003516
24. Ellis-Monaghan, J., Merino, C.: Graph polynomials and their applications I: The Tutte polynomial. In: M. Dehmer (ed.) Structural Analysis of Complex Networks, pp. 219–255. Birkhäuser Boston (2011). DOI 10.1007/978-0-8176-4789-6_9
25. Ellis-Monaghan, J., Merino, C.: Graph polynomials and their applications II: Interrelations and interpretations. In: M. Dehmer (ed.) Structural Analysis of Complex Networks, pp. 257–292. Birkhäuser Boston (2011). DOI 10.1007/978-0-8176-4789-6_10
26. Geelen, J.: A generalization of Tutte's characterization of totally unimodular matrices. Journal of Combinatorial Theory, Series B **70**, 101–117 (1997). DOI 10.1006/jctb.1997.1751
27. Genest, F.: Graphes eulériens et complémentarité locale. Ph.D. thesis, Université de Montréal (2002). Available online: `arXiv:math/0701421v1`
28. Glantz, R., Pelillo, M.: Graph polynomials from principal pivoting. Discrete Mathematics **306**(24), 3253–3266 (2006). DOI 10.1016/j.disc.2006.06.003
29. Harju, T., Li, C., Petre, I., Rozenberg, G.: Parallelism in gene assembly. Natural Computing **5**(2), 203–223 (2006). DOI 10.1007/s11047-005-4462-0
30. Jaeger, F.: On transition polynomials of 4-regular graphs. In: G. Hahn, G. Sabidussi, R. Woodrow (eds.) Cycles and Rays, *NATO ASI Series*, vol. 301, pp. 123–150. Kluwer (1990). DOI 10.1007/978-94-009-0517-7_12
31. Kotzig, A.: Eulerian lines in finite 4-valent graphs and their transformations. In: Theory of graphs, Proceedings of the Colloquium, Tihany, Hungary, 1966, pp. 219–230. Academic Press, New York (1968)
32. Martin, P.: Enumérations eulériennes dans les multigraphes et invariants de Tutte-Grothendieck. Ph.D. thesis, Institut d'Informatique et de Mathématiques Appliquées de Grenoble (IMAG) (1977). Available online: `http://tel.archives-ouvertes.fr/tel-00287330_v1/`
33. Oum, S.: Rank-width and vertex-minors. Journal of Combinatorial Theory, Series B **95**(1), 79–100 (2005). DOI 10.1016/j.jctb.2005.03.003

34. Parsons, T.: Applications of principal pivoting. In: H. Kuhn (ed.) Proceedings of the Princeton Symposium on Mathematical Programming, pp. 567–581. Princeton University Press (1970)
35. Penrose, R.: Applications of negative dimensional tensors. In: D. Welsh (ed.) Combinatorial Mathematics and its Applications, pp. 211–244. Academic Press (1971)
36. Pevzner, P.: Computational Molecular Biology: An Algorithmic Approach. MIT Press (2000)
37. Prescott, D.: Genome gymnastics: Unique modes of DNA evolution and processing in ciliates. Nature Reviews **1**, 191–199 (2000). DOI 10.1038/35042057
38. Prescott, D., Ehrenfeucht, A., Rozenberg, G.: Molecular operations for DNA processing in hypotrichous ciliates. European Journal of Protistology **37**, 241–260 (2001). DOI 10.1078/0932-4739-00807
39. Prescott, D., Greslin, A.: Scrambled actin I gene in the micronucleus of Oxytricha nova. Developmental Genetics **13**, 66–74 (1992). DOI 10.1002/dvg.1020130111
40. Schur, J.: Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind. Journal für die reine und angewandte Mathematik **147**, 205–232 (1917). URL `http://resolver.sub.uni-goettingen.de/purl?PPN243919689_0147`
41. Traldi, L., Zulli, L.: A bracket polynomial for graphs, I. Journal of Knot Theory and Its Ramifications **18**(12), 1681–1709 (2009). DOI 10.1142/S021821650900766X
42. Tsatsomeros, M.: Principal pivot transforms: properties and applications. Linear Algebra and its Applications **307**(1-3), 151–165 (2000). DOI 10.1016/S0024-3795(99)00281-5
43. Tucker, A.: A combinatorial equivalence of matrices. In: Combinatorial Analysis, Proceedings of Symposia in Applied Mathematics, vol. X, pp. 129–140. American Mathematical Society (1960). DOI 10.1090/psapm/010