COMBINATORIAL GRAY CODES—AN UPDATED SURVEY

TORSTEN MÜTZE

ABSTRACT. A combinatorial Gray code for a class of objects is a listing that contains each object from the class exactly once such that any two consecutive objects in the list differ only by a 'small change'. Such listings are known for many different combinatorial objects, including bitstrings, combinations, permutations, partitions, triangulations, but also for objects defined with respect to a fixed graph, such as spanning trees, perfect matchings or vertex colorings. This survey provides a comprehensive picture of the state-of-the-art of the research on combinatorial Gray codes. In particular, it gives an update on Savage's influential survey [C. D. Savage. A survey of combinatorial Gray codes. SIAM Rev., 39(4):605–629, 1997.], incorporating many more recent developments. We also emphasize the connections to closely related problems in graph theory, algebra, order theory, geometry and algorithms, which embeds this research area into a broader context. Lastly, we collect and propose a number of challenging research problems, thus stimulating new research endeavors.

CONTENTS

(Torsten Mütze) INSTITUT FÜR MATHEMATIK, UNIVERSITÄT KASSEL, GERMANY

E-mail address: tmuetze@mathematik.uni-kassel.de.

Version history

This is version 2 of the manuscript. Each item summarizes the changes compared to the previous version.

1. (2022-02-02) Initial version.
2. (2024-07-31) New Section 2.7 on 0/1-polytopes; new Section 8.2 on ordered trees; Figure 7 was split into two (now Figure 8); new Figure 13; problems P27, P28 and P30 were solved; added progress on problems P10, P29, P42 and P56; added new problems P61–P67; included new results and added or updated several references; added version history (added version history (added version history (added version history (... [stack overflow] ...)))).

1. Introduction

In computer science and discrete mathematics, we frequently encounter different classes of combinatorial objects. For example, we may be interested in all subsets or all permutations of some ground set, all strings over some alphabet, all binary trees on a certain number of nodes, or all spanning trees of a graph, to mention just a few examples. A *combinatorial Gray code* is a listing of the combinatorial objects in the class of interest that contains each object exactly once such that any two consecutive objects in the list differ only by a 'small change'. This concept is named after Frank Gray, a physicist and researcher at Bell Labs, whose patent [Gra53] from 1953 describes a method for listing all bitstrings of length n such that any two consecutive strings differ in exactly one bit. Combinatorial Gray codes have a large number of real-world applications, and we shall see that this research area has ramifications into graph theory, algebra, order theory, geometry and algorithms.

The starting point of this work is Savage's influential survey [Sav97] from 1997. Since its publication, the area of combinatorial Gray codes has flourished tremendously, and many new and exciting results have appeared. In particular, several of the hard open problems mentioned in her survey are now solved, and various general techniques for constructing and arguing about Gray codes have emerged, drawing on the aforementioned connections to related mathematical disciplines, which embeds this research area into a much broader context. This survey aims to provide a comprehensive overview over the state-of-the-art, and to provide pointers to closely related research strands. Another goal is to collect and propose a number of challenging research problems, thus stimulating new research endeavors. These open problems of varying difficulty are marked with $\boxed{\text{P}}$ on the page margins. Some of these problems are also mentioned in the problem collection by Jackson, Stevens and Hurlbert [JSH09], and some in Gould's survey [Gou91, Sec. 6] on Hamilton cycles in graphs. Knuth's book [Knu11] also contains several challenging research problems on Gray codes.

This survey focuses mainly on basic combinatorial objects of the types mentioned before, i.e., bitstrings, permutations, words, trees etc. We also discuss some objects defined with respect to a

fixed graph, such as spanning trees, perfect matchings, vertex colorings etc., and objects defined with respect to a fixed poset, such as linear extensions and ideals. These and other discrete structures and small change operations between them are subject of the broader research area of *combinatorial reconfiguration*; see the surveys [vdH13, Nis18, MN19]. For important classes of combinatorial objects, we often refer to the corresponding counting sequence in the OEIS [oei21] by its ID number in square brackets. As this survey is dynamic, it allows incorporating new results and developments as they arise. The reader is encouraged to contact the author with any such updates, corrections or pointers to missing references, and also with suggestions for additional open problems.

Undoubtedly, one of the main applications of Gray codes is *combinatorial generation*, i.e., the algorithmic task of efficiently producing a listing of the combinatorial objects of interest, each object exactly once. Gray codes lend themselves to fast generation, because of the 'small change' property between consecutive objects, provided that those changes can be computed efficiently. However, there are orderings of objects that are not Gray codes, and that can still be computed efficiently in an amortized sense. For example, in the counting order of all bitstrings of length n, which is the same as lexicographic order, the average number of flipped bits is $2 - o(1)$, but in some steps, many bits (up to n) are flipped. Having said that, in this survey we focus entirely on Gray code listings for combinatorial objects, and not on the broader task of efficient combinatorial generation.

There are many excellent books on combinatorial generation, which include a lot of material on Gray codes. Specifically, the author drew a great amount of inspiration from Ruskey's book draft [Rus03] and from Knuth's book [Knu11]. Moreover, the book by Nijenhuis and Wilf [NW75] is a classic, as well as Wilf's update [Wil89]. Other books that contain material on the subject are [Eve73, RND77, PW79, Wil85, Lün89, KS99]. Another useful survey on combinatorial generation is [Was16], and many of the results listed there are based on the well-known reverse search method of Avis and Fukuda [AF96].

De Bruijn sequences and *universal cycles* are closely related to Gray codes, but beyond the scope of this survey. We refer to the papers [Fre82, CDG92] and to the online resource [dB] for readers interested in this subject.

1.1. **Outline of this paper.** In Section 2 we introduce some terminology that is used throughout this paper, and we collect a few general observations that are valid for any Gray code problem. We also show how Gray codes relate to several other research areas in mathematics and computer science. Sections 3–10 describe the literature on combinatorial Gray codes, sorted by objects and sometimes by methods. Some objects generalize multiple other objects, and assigning them to one particular section is somewhat arbitrary. For example, multiset permutations generalize combinations and permutations, and they are treated in Section 5 on permutations. This survey contains numerous figures of different Gray codes, which facilitate comparison between the different constructions, thus also stimulating new research questions. Moreover, the figures convey the wealth and aesthetic appeal of the subject.

2. Preliminaries

2.1. **Flip graphs.** Nearly every Gray code problem can be described by considering the corresponding *flip graph*. The vertices of this graph are the combinatorial objects to be listed in the Gray code, and the graph has an edge connecting any two objects that differ by the desired 'small change' operation, sometimes generically referred to as a *flip*. If the flip operation is not an involution, then the edges of the flip graph are directed. In this way, we obtain a huge family
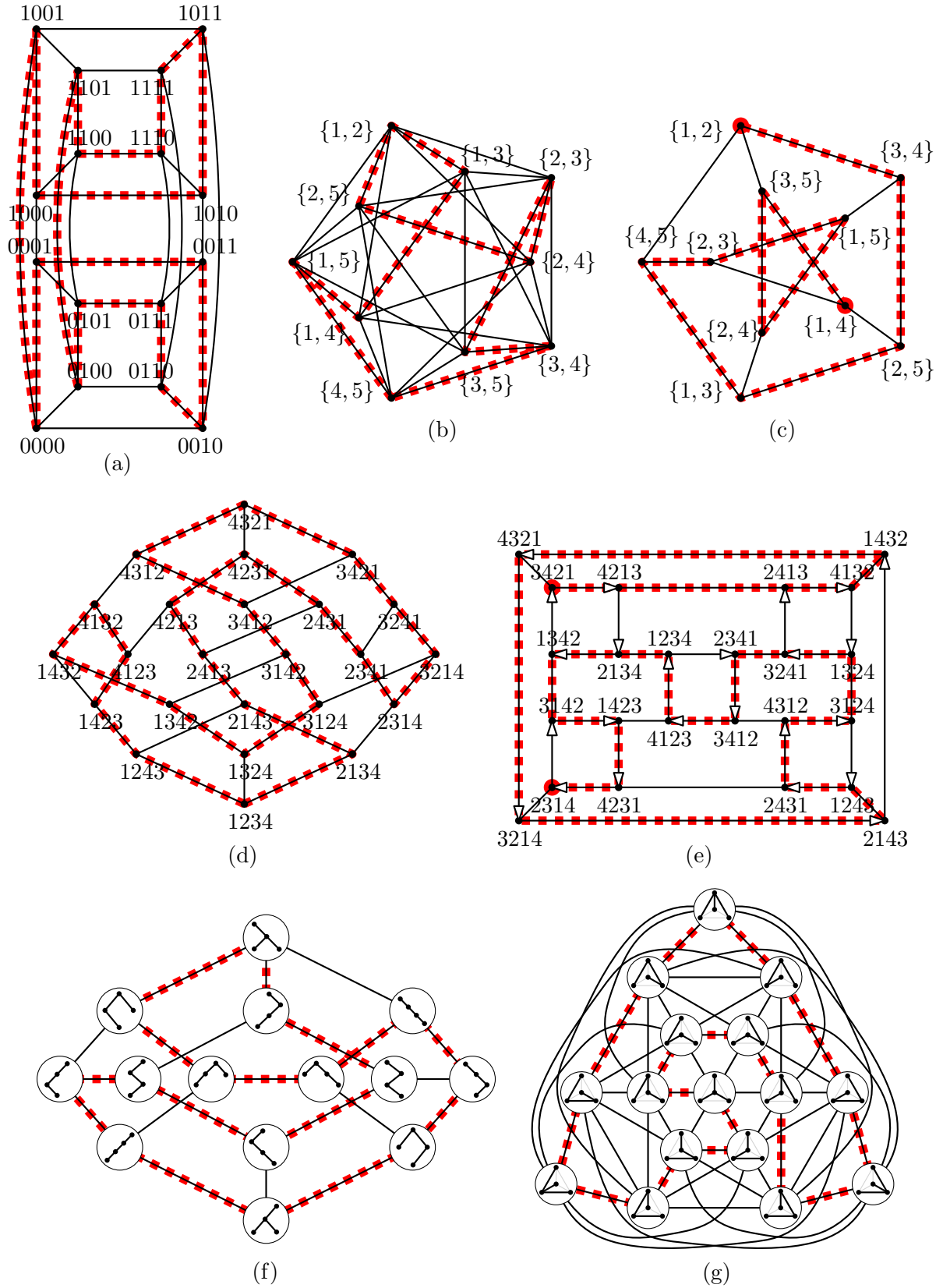
FIGURE 1. Different flip graphs with a Hamilton path or cycle: (a) 4-cube Q_4; (b) Johnson graph $J(5,2)$; (c) Kneser graph $K(5,2)$ (=Petersen graph); (d) permutahedron; (e) sigmatau graph (permutations under left shifts or swaps of the first two entries); (f) associahedron; (g) spanning trees of K_4 under edge exchanges.

of interesting graphs; see Figure 1. For example, the flip graph of bitstrings of length n under flipping a single bit is the *n-dimensional hypercube Q_n*, or *n-cube* for short (Figure 1 (a)). The flip graph of subsets of $[n] := \{1, 2, \ldots, n\}$ of fixed size k under exchanging a single element is the *Johnson graph $J(n, k)$* (Figure 1 (b)). The flip graph of subsets of $[n]$ of fixed size k under disjointness is the *Kneser graph $K(n, k)$* (Figure 1 (c)). The flip graph of permutations of $[n]$ under adjacent transpositions is the *permutahedron* (Figure 1 (d)). The flip graph of binary trees with n nodes under tree rotations is the *associahedron* (Figure 1 (f)). Clearly, finding a Gray code is equivalent to finding a Hamilton path or cycle in the corresponding flip graph. The term 'Gray code' can refer to both a Hamilton path or cycle, and we call a Gray code *cyclic* to mean a Hamilton cycle, i.e., the last and first object in the Gray code listing also differ by a flip.

In some of these examples we see that a flip operation can change many, possibly all, entries of the string representation of an object, so the question of what constitutes a 'small change' is up to debate (but can sometimes be justified via polytopes; see Sections 2.6 and 2.7). For example, the directed edges in Figure 1 (e) correspond to cyclic left shifts of the permutations, which result in a change of every entry of the permutation. Similarly, the edges of Kneser graphs as shown in Figure 1 (c) correspond to disjoint sets, i.e., a flip exchanges all elements. It turns out that such flip operations often lead to very interesting Gray codes. Insisting that a Gray code may only change a constant number of positions in the string, which is prevalent in many papers, is a misguided paradigm. One reason is that the amount of change is entirely dependent on the representation of the object. For example, we will encounter Gray codes for permutations that use cyclic shifts of substrings, and while these shifts may change many entries of the permutation, they change only a single entry in the corresponding inversion vector. Another reason is that depending on the available hardware, these 'not-so-small' changes may actually be implemented faster.

Flip graphs allow us to apply graph-theoretic tools to tackle the Gray code problem. Moreover, they call for investigating all kinds of other interesting graph-theoretic parameters beyond Hamiltonicity, such as girth, diameter, connectivity, colorability, independence number etc.

In some Gray code problems, the admissible flip operations depend on the recent history of previous flips, and in this case there is no underlying 'static' flip graph, which typically makes these problems hard. Examples of this are the order-preserving paths of Felsner and Trotter and Beckett-Gray codes (see Sections 3.3 and 3.8).

2.2. **Necessary conditions.** There are a few simple graph-theoretic conditions that a graph G has to satisfy in order to have a Hamilton path or cycle. Trivially, G needs to be connected. Furthermore, if G has vertices of degree 1, then a Hamilton cycle cannot exist, and if G has more than two vertices of degree 1, then a Hamilton path cannot exist. Also, if G has precisely two vertices of degree 1, then any Hamilton path in G must start and end at these two vertices. Moreover, if G is bipartite with partition classes of different sizes, then a Hamilton cycle cannot exist, and if the size difference between the partition classes is more than 1, than a Hamilton path cannot exist. This condition generalizes straightforwardly to k-partite graphs G for $k \geq 2$, where one compares the size of one partition class against the sum of the sizes of all others. We refer to this condition on G as *balancedness constraint*. As we shall see, for most interesting flip graphs G, the aforementioned connectivity, minimum degree and balancedness constraints are the only obstacles for Hamiltonicity, i.e., once G satisfies these three constraints, then it does have a Hamilton path or cycle. This could be considered a 'meta-conjecture' for most interesting flip graphs (a notable counterexample is mentioned in [Naa01]).

2.3. **Stronger Hamiltonicity notions.** For constructing Gray codes, it is often helpful to establish stronger Hamiltonicity properties of the corresponding flip graph. Specifically, a graph is called *Hamilton-connected*, if it admits a Hamilton path between every pair of vertices. Moreover, a bipartite graph is called *Hamilton-laceable*, if it admits a Hamilton path between every pair of vertices from the two partition classes. For example, the hypercube and the permutahedron for $n \geq 4$ are Hamilton-laceable (see [Sim78] and [Tch82], respectively), and the Johnson graph is Hamilton-connected [JR94, Kno94]. There are many more strengthened Hamiltonicity properties in addition to the aforementioned two, some of which we will encounter in Section 9; see also [XM09]. Establishing these stronger properties yields more information about the flip graph, and maybe counter-intuitively, it often makes the proof of Hamiltonicity *easier*, and *not harder*. This is because the inductive statements involving these stronger notions are more powerful and flexible.

2.4. **General Gray code results.** As mentioned before, a Gray code corresponds to a Hamilton path or cycle in the corresponding flip graph G. More generally, a *k-Gray code* is a sequence of vertices of G that contains every vertex exactly once, such that any two consecutive vertices in the sequence have distance at most k in G. Given a graph G, the *kth power of G*, denoted G^k, is the graph with the same vertex set as G, connecting any two vertices that have distance at most k in G with an edge. With this definition, a k-Gray code corresponds to a Hamilton path in G^k, and a cyclic k-Gray code corresponds to a Hamilton cycle in G^k.

Sekanina [Sek60] proved that the cube G^3 of any connected graph G has a Hamilton cycle. Equivalently, his result asserts that any connected graph G admits a cyclic 3-Gray code. If a flip graph is not just connected, but 2-connected, even more can be said. Specifically, Fleischner [Fle74] proved that the square G^2 of any 2-connected graph G has a Hamilton cycle. Equivalently, any 2-connected graph G admits a cyclic 2-Gray code. Karaganis [Kar68] strengthened Sekanina's result, by showing that for any connected graph G, the cube G^3 is Hamilton-connected. Similarly, Chartrand, Hobbs, Jung, Kapoor, and Nash-Williams [CHJ+74] strengthened Fleischner's theorem, by showing that for 2-connected G, the square G^2 is Hamilton-connected. Alstrup, Georgakopoulos, Rotenberg and Thomassen [AGRT18] developed an algorithm for computing such Hamilton cycles and paths in G^2, running in time and space $O(|E|)$, where $|E|$ is the number of edges of G. Sekanina's aforementioned result about G^3 is also algorithmic, with the same bounds, and a cycle can be obtained by traversing any spanning tree of G in prepostorder; see [Knu11, Sec. 7.2.1.6]. To complement these results, it is known that if G is not 2-connected, then the Hamiltonicity problem for G^2 is NP-complete [Und78].

These results, which deserve to be better known in the Gray code community, impose some criteria against which we should compare any Gray code results obtained for particular combinatorial objects. Put differently, as soon as the flip graph of a Gray code problem is connected or 2-connected, we are immediately guaranteed a 3-Gray code or 2-Gray code, respectively, by the results from before, so any further work on the problem should either focus on obtaining 2- or 1-Gray codes, or on conceptual simplicity or algorithmic efficiency for the particular problem.

2.5. **Algorithmic questions.** Of course, the linear bound $O(|E|)$ in the algorithms discussed in the previous section is not very satisfying in the context of Gray codes. This is because the size of a flip graph for a particular Gray code problem is often exponential in the size of the parameter that specifies the objects. For example, the n-cube has 2^n vertices and $2^{n-1}n$ edges, or the permutahedron for $[n]$ has $n!$ vertices and $n!(n-1)/2$ edges. In those situations, we will much prefer Gray code algorithms that work in time and space that are polynomial in n. The

word 'time' here refers to the time per generated object, or equivalently, to the delay between two consecutive objects. The ultimate goal from an algorithmic point of view is to develop a Gray code algorithm with constant delay, i.e., each new object is produced in constant time. Such algorithms are sometimes called *loopless* or *loop-free*, a term coined by Ehrlich [Ehr73b]. Another interesting class are *constant amortized-delay* or *constant amortized-time (CAT)* algorithms, which produce each new object in constant time on average (not necessarily in the worst case). In addition to being fast, a Gray code algorithm ideally requires only linear space to store the current object and possibly some auxiliary data structures. Information-theoretic trade-offs concerning the space- and memory-efficient computation of Gray codes are explored in the literature under the term *quasi-Gray codes*, a direction initiated by Fredman [Fre78].

Apart from generating a Gray code listing, there are two closely related algorithmic tasks of interest, namely *ranking* and *unranking*. Ranking means to compute, for a given combinatorial object, its position in the Gray code listing. On the other hand, unranking means to compute, for a given position, the object at this position in the Gray code listing. Clearly, efficient unranking can be used for efficient random generation of the objects.

In this survey, we comment on the aforementioned algorithmic issues for several Gray codes problems, but we do not delve into low-level implementation details. Knuth covers a large number of these algorithms in his book [Knu11], which also contains numerous challenging research problems and interesting background information; see also [Knu20]. Furthermore, Arndt's book [Arn11] contains a vast number of recent C++ implementations and experimental comparisons of many of the Gray code algorithms discussed here. Lastly, many of these algorithms are available on the Combinatorial Object Server [cos] for experimentation and download, an online project that was first launched by Frank Ruskey.

2.6. **Posets and polytopes.** Interestingly, many flip graphs can be equipped with a poset structure. For example, the hypercube is the cover graph of the Boolean lattice, the permutahedron is the cover graph of the weak Bruhat order of the symmetric group S_n, and the associahedron is the cover graph of the Tamari lattice (recall Figure 1 (a)+(d)+(f)). Moreover, many flip graphs are skeleta of high-dimensional polytopes, including the aforementioned hypercube, permutahedron and associahedron. These observations link the Gray code problem to order theory and geometry, enabling us to apply tools and insights from those areas. For example, Balinski's theorem yields information about the connectivity of the flip graph based on the dimension of the polytope. Also, the geometric viewpoint can be used to justify the choice of 'small change' condition used in a Gray code for a particular set of objects: The change is small if and only if it corresponds to walking along an edge of the polytope (see Section 2.7 below). Furthermore, this allows broad generalization of the classes of flip graphs under investigation, and unification of several known Gray code results through a common lens:

- The hypercube, permutahedron and associahedron arise as cover graphs of quotients of lattice congruences of the symmetric group S_n [Rea12], and the corresponding polytopes are known as *quotientopes* [PS19, PPR23]. This is a family of $2^{2^{n(1+o(1))}}$ many flip graphs, and there is a simple greedy algorithm for computing a Hamilton path in each of them [HM21], which yields a unified description of several known Gray codes for bitstrings, permutations, binary trees, set partitions, rectangulations etc. [HHMW22].
- *Graph associahedra*, introduced by Carr and Devadoss [CD06, Dev09] and Postnikov [Pos09], are another large class of polytopes that generalize hypercube, permutahedron and associahedron. They yield a family of $2^{n^2/2(1+o(1))}$ many flip graphs, most of which have a Hamilton cycle [MP16, CMM22].

- The Johnson graph and the flip graph of spanning trees of a graph under edge exchanges (recall Figure 1 (b)+(g)) arise as skeleta of *matroid base polytopes*. In fact, these polytopes are part of the much larger family of *0/1-polytopes* discussed in the next section, which are all Hamilton-connected or Hamilton-laceable [NP84], and a Hamilton path on them can be computed by a simple greedy algorithm.
- Flip graphs on acyclic orientations of a graph induced by flipping the orientation of a directed cycle or a directed cut can be equipped with a distributive lattice structure [Pro21], and they are realized by distributive polytopes that are part of the large family of *alcoved polytopes* [LP07, FK11].
- Flip graphs on acyclic orientations of a graph induced by flipping the orientation of a single arc also have a poset structure, and they can be realized as polytopes [SZ98]. These posets sometimes form a lattice and admit lattice congruences that can also be realized as polytopes [Pil22], generalizing the aforementioned quotientopes.

The question when Hamilton paths or cycles exist for the latter two classes of flip graphs is still very much open in general (see also Problem P56). $\boxed{\text{P1}}$

2.7. **0/1-polytopes.** Given any set of bitstrings $X \subseteq \{0,1\}^n$, the convex hull of X in $\mathbb{R}^n$ is a *0/1-polytope*. If X encodes a set of combinatorial objects, then the edges of this polytope 'automatically' capture small change operations between the objects. In the simplest case, if we take $X = \{0,1\}^n$, we obtain the hypercube, with edges corresponding to flipping a single bit. Furthermore, if we take X as characteristic vectors of bases of a matroid, we obtain the matroid base polytope, whose edges correspond to base exchanges. If we take X as characteristic vectors of matchings of a graph, we obtain the matching polytope [Chv75], whose edges correspond to alternating paths/cycles. In particular, perfect matchings of the complete bipartite graph $K_{n,n}$ are in bijection with permutation matrices, and their convex hull yields the Birkhoff polytope, whose edges correspond to multiplication with a permutation cycle. Furthermore, if we take X as characteristic vectors of independent sets or vertex covers of a graph, we obtain the stable set polytope and vertex cover polytope [HK78], respectively, whose edges correspond to a symmetric difference exchange on a connected subgraph. As a last example, if we take X as characteristic vectors of antichains or ideals of a poset, we obtain the chain polytope or order polytope [Sta86], respectively, whose edges correspond to a symmetric difference exchange on a connected subposet.

In fact, Naddef and Pulleyblank [NP81, NP84] proved that every 0/1-polytope is either Hamilton-connected or a hypercube, which is known to be Hamilton-laceable. An efficient algorithm for computing a Hamilton path on the skeleton of any 0/1-polytope was proposed by Merino and Mütze [MM23b]. Their algorithm greedily modifies the shortest possible prefix of the current vertex from X and moves to another vertex with minimum Hamming distance from the current one. The algorithm can start at an arbitrary vertex of the polytope, and it can be implemented efficiently by solving instances of the corresponding linear optimization problems over X. This approach yields a large number of efficient Gray code algorithms for a variety of objects based on matroids, graphs and posets, such as matroid bases, spanning trees, forests, matchings, maximum matchings, and cost-optimal matchings in general graphs, vertex covers, minimum vertex covers and cost-optimal vertex covers in bipartite graphs, antichains, maximum antichains and cost-optimal antichains in posets, as well as ideals and cost-optimal ideals of posets. In each case, the small change operations used are given by the edges of the corresponding 0/1-polytope.

2.8. **Lovász' conjecture and Cayley graphs.** Many natural Gray code problems yield flip graphs that are highly symmetric. Specifically, many are vertex-transitive, such as the

hypercube, permutahedron, Johnson graphs or Kneser graphs, and others are regular, such as the

P2 associahedron, or more generally, all graph associahedra. This connects the Gray code problem to a well-known and long-standing conjecture due to Lovász [Lov70], which asserts that every connected vertex-transitive graph has a Hamilton path. A stronger version of the conjecture asserts that such a graph admits a Hamilton cycle, unless it is one of following five exceptional graphs: a single edge, the Petersen graph, the Coxeter graph, and the graphs obtained from the latter two by replacing every vertex by a triangle. Thomassen conjectured that there are only finitely many such counterexamples (see [Ber78]). Note that by a theorem of Mader [Mad70], every vertex-transitive graph with degree k is k-connected, provided that it does not contain a clique of size 4, which can be seen as evidence for Lovász' conjecture. In stark contrast, Babai [Bab95] conjectured that for some $\varepsilon > 0$, there are infinitely many connected vertex-transitive graphs with n vertices (even Cayley graphs) without cycles of length at least $(1 - \varepsilon)n$.

There are several results establishing Lovász' conjecture subject to conditions on n, the number of vertices of the graph. For example, vertex-transitive graphs with prime n are circulant graphs [Tur67], and therefore a Hamilton cycle can be traced by following vertices with a fixed step size (modulo n) along the circulant vertex ordering. Similarly, the conjecture has been confirmed if n equals one of the following expressions, where p and q denote prime numbers: $2p$ [Als79], $3p$ [Mar88], $4p$ [KM08], $5p$ [MP82], $6p$ [KŠ09], $10p$ [KMZ12] (with some additional assumptions), p^2 and p^3 [Mar85], p^4 [Che98], p^5 [Zha15], $2p^2$ [Mar87], and pq [DKM21], in some cases in the stronger Hamilton cycle version. Christofides, Hladký and Máthé [CHM14] proved that for every $\varepsilon > 0$ there is an n_0 such that every vertex-transitive graph with $n \geq n_0$ vertices and degree at least εn has a Hamilton cycle. For more results on Lovász' conjecture, we refer to the survey [KM09] by Kutnar and Marušič.

P3 Cayley graphs are a major class of vertex-transitive graphs, and are thus also an important special case of Lovász' conjecture. The question for Hamilton cycles in Cayley graphs was first raised by Rapaport-Strasser [RS59]. For a finite group G and a set of generators X, the *Cayley graph* $\Gamma(G, X)$ has vertex set G and edges $(g, gx), (g, gx^{-1})$ for all $g \in G$ and $x \in X$. Every two vertices in this graph that are connected by an edge also have an edge in the opposite direction, so it can be considered as an undirected graph. The *Cayley digraph* $\vec{\Gamma}(G, X)$ has the vertex set G and directed edges (g, gx) for all $g \in G$ and $x \in X$ (i.e., inverse operations must not be used). Note that if $X^{-1} := \{x^{-1} \mid x \in X\} = X$, then the distinction between these two graphs is irrelevant. This happens for example if X is a set of involutions, i.e., $x^2 = 1$ for all $x \in X$. Also note that if $G = S_n$ is the symmetric group and $x = \sigma_k \in X$ is the operation that cyclically *left-shifts* the first k entries of the permutation, then for every directed edge $(\pi, \pi x) =: (\pi, \pi')$, $\pi \in S_n$, of $\vec{\Gamma}(G, X)$, we have $\pi'(i) = \pi(x(i)) = \pi(\sigma_k(i))$ for all $i = 1, \ldots, n$, i.e., the permutation π' is obtained from π by cyclically *right-shifting* the first k entries. In this case, it is maybe more intuitive to consider the Cayley graph $\vec{\Gamma}(G, X^{-1})$, which is obtained from $\vec{\Gamma}(G, X)$ by reversing the orientation of all edges; see Figure 1 (e).

There is a substantial number of cases for G and X for which $\Gamma(G, X)$ or $\vec{\Gamma}(G, X)$ are known to have a Hamilton cycle. One instance is the Chen-Quimpo theorem [CQ81], which asserts that if G is an abelian group such that the Cayley graph $\Gamma(G, X)$ is connected and has degree at least 3, then it is Hamilton-connected if it is not bipartite and Hamilton-laceable if it is bipartite. Conway, Sloane and Wilks [CSW89] showed that if G is a direct product $G = G_1 \times \cdots \times G_k$ of irreducible Coxeter groups, and if X_i is the canonical basis for G_i for all $i = 1, \ldots, k$, then the Cayley graph $\Gamma(G, X)$ for the basis $X = X_1 \times \cdots \times X_k$ has a Hamilton cycle. For more results on the Hamiltonicity of Cayley graphs, see Sections 5.2 and 5.3 and the

surveys [Als81, WG84, CG96, PR09, LPR$^+$19] (see also [GKMM12] for another relevant result on 3-regular Cayley graphs).

2.9. **Genlex ordering.** As mentioned before, this survey focuses on Gray codes and not on the broader topic of efficient combinatorial generation. For this reason, we never discuss lexicographic or colexicographic orderings. Put differently, for those orderings there is no corresponding flip operation and hence no flip graph. Nevertheless, there is a generalization of lexicographic and colexicographic orderings that is very relevant in the context of Gray codes, namely genlex ordering, a term coined by Walsh [Wal03]. A listing of strings over some alphabet is in *genlex order*, if all strings with the same prefix (or suffix) appear consecutively. Such listings are also sometimes referred to as *prefix-partitioned* (or *suffix-partitioned*) [VW06]. This generalizes *graylex order* introduced by Chase [Cha89], in which the symbols after the common prefixes (or before the common suffixes) appear in monotonic order, i.e., increasingly or decreasingly. Note that in lexicographic or colexicographic order, they always appear in increasing order. We shall see that many different classes of objects admit genlex Gray codes. This is significant, as any genlex listing can be constructed by a simple greedy algorithm; see Section 10.

3. Binary strings

In this section we discuss Gray codes for listing all 2^n bitstrings of length n, subject to various additional conditions. We also discuss Gray codes for subsets of bitstrings that arise from structural constraints in the hypercube Q_n, from forbidding certain substrings, or from considering equivalence classes of strings. Most Gray codes in this section flip a single bit in each step, i.e., they correspond to paths or cycles in Q_n or subgraphs of it. In Sections 3.16, 3.17 and 3.18 we also encounter 2- and 3-Gray codes, which flip at most 2 or 3 bits in each step, which corresponds to paths or cycles in Q_n^2 or Q_n^3. Another exception is Section 3.14, where we consider Gray codes that use cyclic shifts or transpositions of bits. Note that bitstrings of length n can be interpreted as characteristic vectors of subsets of $[n]$, so the Gray codes discussed in this section can be seen as listings of subsets of the ground set $[n]$. Flipping a single bit in a bitstring corresponds to adding or removing a single element in the corresponding subsets.

For any string x and any integer $\ell \geq 0$, we write x^ℓ for the ℓ-fold repetition of x.

3.1. **The binary reflected Gray code.** The *binary reflected Gray code (BRGC)* [Gra53] lists all bitstrings of length n such that any two consecutive strings differ in a single bit, including the last and first bitstring. The BRGC is defined inductively as $\Gamma_0 := \varepsilon$ and $\Gamma_{n+1} := 0\Gamma_n, 1\operatorname{rev}(\Gamma_n)$ for $n \geq 0$, where ε denotes the empty sequence and rev() denotes sequence reversal; see Figure 2 (a). In words, the listing for strings of length $n + 1$ is obtained by prepending 0 to all strings in the listing for n, followed by the reversed listing in which all strings are prepended 1. The idea of this code was known long before Gray, and Knuth [Knu11, Sec. 7.2.1.1] tells more about its history. In particular, the BRGC is useful for solving the Tower of Hanoi puzzle [HKMP13], the Chinese rings puzzle, and to construct Brunnian links [Rus03, Sec. 5.2.1]. The BRGC has very simple ranking/unranking functions, and many other fascinating properties. For example, the BRGC ordering is genlex by definition, and genlex Gray codes for several other interesting combinatorial objects can be obtained by considering sublists of the BRGC; see Sections 3.16–3.18. Similarly, Tang and Liu [TL73] showed that in the sublist of the BRGC induced by bitstrings with a fixed number of 1s, any two consecutive bitstrings differ in two bits (a transposition of 0 and 1); see Section 4.1. Also, it was shown by Yuen [Yue74] and Cavior [Cav75] that if any two n-bit strings differ in $t > 0$ bits, then they are at least $\lceil 2^t/3 \rceil$ steps and at most $2^n - \lceil 2^t/3 \rceil$ steps apart in the BRGC. More interesting properties of the BRGC are established in the papers by

Conder [Con99], and by Agrell, Lassing, Ström, and Ottosson [ALSO04]. Bitner, Ehrlich and
Reingold [BER76] describe a loopless algorithm for generating the BRGC. For many of the Gray
codes with additional constraints discussed in the following sections, no such efficient algorithms
are known, and it would be very interesting to devise them.

3.2. **Transition counts.** Since the discovery of the BRGC, there has been continued interest
in constructing Gray codes with additional desirable properties that the BRGC does not have.
A problem that has received a lot of attention is to construct a Gray code where the number of
flips are distributed as equally as possible over the n bit positions; see Figure 2 (b). Specifically,
let c_i denote the number of times that bit i is flipped along a given Gray cycle, where we
number bits from right to left with $i = 1, \ldots, n$. Clearly, each transition count c_i must be
an even number, and they sum up to 2^n, so we require that each c_i is approximately equal
to $2^n/n$. Formally, in a *balanced Gray code*, we require that $|c_i - 2^n/n| < 2$ for $i = 1, \ldots, n$. In
particular, if n is a power of 2, then $c_1 = \cdots = c_n = 2^n/n$. Note that in the BRGC, we have
$c_i = 2^{n-i}$ for $i = 1, \ldots, n-1$ and $c_n = 2$, i.e., it is very unbalanced. A balanced Gray code
was first constructed by Tibor Bakos around 1955, and his proof is published in Ádám's book
[Ádá68, p. 28–37]. Apparently, this proof got forgotten, and the entire problem was rediscovered
and resolved many years later. In particular, several heuristics for constructing balanced Gray
codes were suggested by Tootill [Too56], Vickers and Silverman [VS80], Ludman [Lud81], and
by Robinson and Cohn [RC81]. Eventually, Wagner and West [WW91] proved the existence of
balanced Gray codes when n is a power of 2, attributing the problem to Stanley, and subsequently
Bhat and Savage [BS96] solved the general case. A very concise general solution is described in
Knuth's book [Knu11, Sec. 7.2.1.1].

A variant of balanced Gray codes are *exponentially balanced Gray codes*, in which each
transition count c_i is one of two consecutive powers of 2. Note that if n is a power of 2, then
this is the same problem as before. Van Zanten and Suparta [vZS04] constructed exponentially
balanced Gray codes for any $n \geq 1$, answering a conjecture posed in the aforementioned paper
by Wagner and West, and a simplified construction was later given by Suparta [Sup05].

The most general version of this problem is to ask for an n-bit Gray code with arbitrary
prescribed transition counts $c_1, \ldots, c_n$. Clearly, we may assume without loss of generality that
$c_1 \leq c_2 \leq \cdots \leq c_n$, and we have to require that each c_i is even, that $\sum_{i=1}^n c_i = 2^n$, and that
$\sum_{i=1}^k c_i \geq 2^k$ for each $1 \leq k \leq n$. The problem is whether every sequence $c_1, \ldots, c_n$ of integers
satisfying those necessary conditions allows constructing a cyclic n-bit Gray code with exactly
these transition counts. Observe that solving this problem for one particular value of n also solves
it for all smaller values. On the other hand, Potapov [Pot12] positively resolved the problem
for all $n > N := \lceil 2^{2^{\mu/2}(4+\log_2 \mu)} \rceil$ where $\mu := 4(2^{35} - 2^4)/31$, *conditional* on the assumption that
it has a positive solution for $n = N$ (and therefore also for all $n < N$). Consequently, settling
the base case $n = N$ for his inductive argument is a very interesting open problem, but is
infeasible to tackle solely with computer help. Already Gilbert [Gil58, Thm. IV], using a lemma
of Shannon [Sha49], showed how to construct a Gray *path* with prescribed transition counts
of the form $1, c_2, \ldots, c_n$ (i.e., the first bit flips exactly once), if they satisfy the corresponding
obvious necessary conditions.

3.3. **Weight-monotonicity.** The *weight* of a bitstring is defined as the number of 1s in it.
Savage and Winkler [SW95] considered Gray codes for which the weight of bitstrings increases
(almost) monotonically along the code. Specifically, a *monotone* Gray path starts at 0^n and once
it has visited a bitstring of weight k, it never returns to a bitstring of weight $k - 2$. The last
bitstring on the path will either have weight n if n is odd or weight $n - 1$ if n is even. Savage
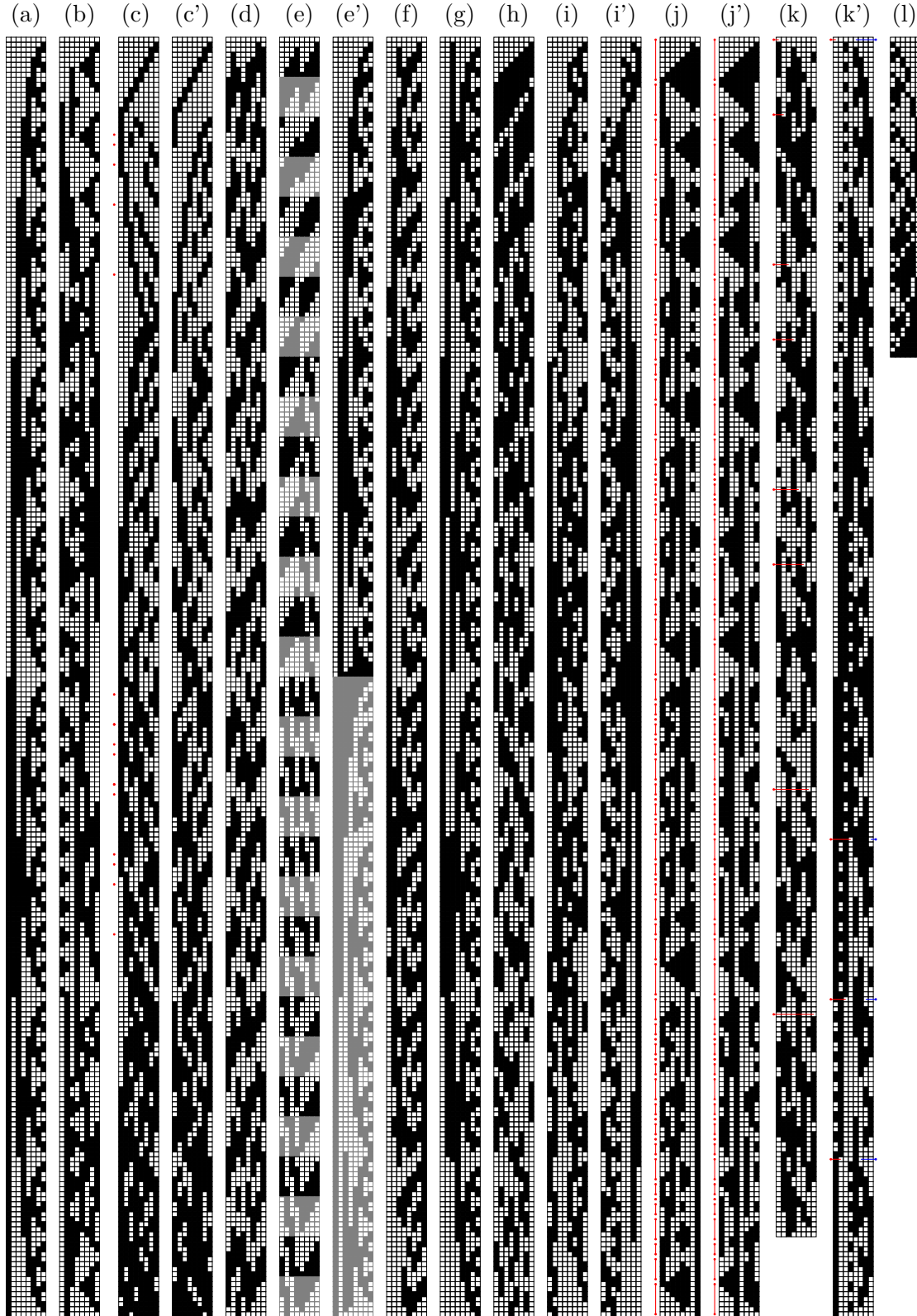
FIGURE 2. Different binary Gray codes (1-bits black, 0-bits white): (a) BRGC Γ_8; (b) balanced; (c) monotone; (c') order-preserving; (d) long-run; (e) 8-antipodal; (e') 128-antipodal; (f) non-local; (g) trend-free; (h) Beckett; (i) K_8-code; (i') Q_3-code; (j) symmetric chains; (j') Greene/Kleitman symmetric chains; (k) single-track (length 240); (k') 2-track and 8-symmetric; (l) Savage-Shields-West (here $n = 6$, unlike for the other codes where $n = 8$).

and Winkler constructed such a monotone path for any $n \geq 1$, and their solution is shown in Figure 2 (c). An even stronger requirement was proposed by Felsner and Trotter [FT95]; see also [JSH09, Problem 483]. They ask for an *order-preserving* Gray path $x_1, \ldots, x_N$ with $N = 2^n$, which starts at 0^n and if x_i is obtained from x_j by a flip $1 \to 0$, then $i \leq j + 1$. It is easy to check that any order-preserving path must also be monotone. However, the Savage-Winkler paths are not order-preserving, and for $n = 8$ the bitstrings violating the order-preserving property are marked in Figure 2 (c) with a small dot on the left. Computer experiments show that for $n = 1, \ldots, 7$, the number of order-preserving Gray paths is $1, 1, 1, 1, 10, 123, 1\,492\,723$ [OEIS A259839], and solutions have also been found for $n \in \{8, 9, 10\}$ (by Trotter and Mütze, independently), one of which is shown in Figure 2 (c'). For comparison, the number of monotone paths for $n = 1, \ldots, 6$ is $1, 1, 1, 1, 14, 46\,935$. In all those counts, paths that differ only in renaming the coordinate directions are factored out and not counted separately. It is a challenging open problem whether order-preserving paths exist for $n \geq 11$. As a partial result, Felsner and Trotter [FT95] constructed an order-preserving Gray path of length $N = 2^{n-1}(1 + o(1))$. Moreover, it was shown by Biró and Howard [BH09a] that for all $n \geq 11$, there is an order-preserving path that traverses all bitstrings with weights $0, 1, 2, 3$ (and some bitstrings with weight 4, but no other weights), so it has length $N = \Theta(n^3)$.

3.4. **Antipodal Gray codes.** In a *t-antipodal Gray code*, we require that the complement of any bitstring is visited either exactly t steps before or later. In principle, t can be any integer in the interval $[n, 2^{n-1}]$, but t and n must have the same parity. Killian and Savage [KS04] attribute this problem to Snevily, and they consider the case when $t = n$. Specifically, they proved that no solution exists for any odd $n \geq 5$ or for $n = 6$, and presented a constructive solution when n is a power of 2, which is shown in Figure 2 (e). They mention as an open problem the question whether an n-antipodal n-bit Gray code exists for even $n \geq 10$. Chang, Eu, and Yeh [CEY07] showed that a t-antipodal n-bit code for odd $n \geq 3$ exists if and only if $t = 2^{n-1} - 1$. Moreover, for even $n \geq 6$ they constructed t-antipodal codes for all $t = 2^{n-1} - 2^k$ with k odd and $1 \leq k \leq n - 3$, and for $t = 2^{n-k}$ with $1 \leq k \leq 3$. Knuth [Knu11, Ex. 49 in Sec. 7.2.1.1] describes a simple solution for even $n \geq 2$ and $t = 2^{n-1}$; see Figure 2 (e'). In Figures 2 (e) and (e'), any bitstring whose complement appears t steps earlier is drawn shaded.

3.5. **Run lengths.** Another property of interest are *run lengths* of a Gray code, i.e., the number of bitstrings visited between two flips of the same coordinate. In Figure 2, the run lengths are the heights of maximal vertical blocks of either black or white squares. We let $r(n)$ denote the maximum value r such that an n-bit Gray code with all run lengths at least r exists. The exact value of $r(n)$ for $n = 1, \ldots, 7$ is $r(n) = 1, 2, 2, 2, 4, 4, 5$, and it is unknown for $n \geq 8$ [Knu11, Sec. 7.2.1.1]. An 8-bit Gray code with all run lengths at least 5 is shown in Figure 2 (d) (it is the same as in [Knu11, Fig. 34 in Sec. 7.2.1.1]). Clearly, we have $r(n) \leq n$, and Goddyn and Gvozdjak [GG03] proved the remarkable fact that $r(n) \geq \lceil n - 2.001 \log_2(n) \rceil \geq n - 3 \log_2(n) = n - \mathcal{O}(\log n)$ for $n \geq 2$, improving upon earlier results due to Evdokimov [Evd80] and Goddyn, Lawrence and Nemeth [GLN88]. Note that in an n-antipodal n-bit Gray code, the run lengths within every pair of a block and its subsequent complemented block are equal to n, but not necessarily across the boundaries of these pairs. Codes with long run lengths have applications in image detectors [LM96] and for structured light [KRL08].

3.6. **Non-locality.** In a *non-local Gray code*, we require that among any 2^t consecutive bitstrings, strictly more than t bit positions change, for any $2 \leq t \leq n - 1$; see Figure 2 (f). This is impossible

for $n = 2, 3, 4$, but can be achieved for any $n \geq 5$, as was first shown by Mills [Mil63]. A simpler construction was later given by Ramras [Ram90].

3.7. Trend-freeness.

In a *trend-free Gray code*, the center of mass of 1-bits in each coordinate position is in the middle. Equivalently, we require that $\sum_{i=1}^{2^n} i(-1)^{x_{i,k}} = 0$ for all $k = 1, \ldots, n$, where $x_{i,k}$ denotes the kth bit in the ith bitstring. Such codes are useful for experiment design [Dic74, VS80, Che85]. Note that the BRGC satisfies the trend-freeness conditions for all bit positions except the leftmost one ($k = n$). Knuth [Knu11, Ex. 75 and 76 in Sec. 7.2.1.1] gives a simple construction of trend-free Gray codes for all $n \geq 5$; see Figure 2 (g), improving upon an earlier result due to Cheng [Che85].

3.8. Beckett-Gray codes.

Beckett-Gray codes are named after the Irish playwright Samuel Beckett, whose play 'Quad' consists of a series of arrivals and departures of characters who appear on stage together in different combinations. Whenever a character leaves, it is the one who has been on stage longest among all characters present. Formally, the problem is to find an n-bit Gray code with the property that on every transition $1 \rightarrow 0$, the 1-bit that disappears is the one that appeared first among all 1-bits. There is no constraint with regards to the other transitions $0 \rightarrow 1$. We can think of the positions of 1s as being stored in a queue that operates on a first-in first-out basis. Cooke, North, Dewar and Stevens [CNDS16] found such codes with the help of computers for $n = 2, 5, 6, 7, 8$, see Figure 2 (h), and showed that none exist for $n = 3, 4$. Sawada and Wong [SW07] report on some computational advances and additional solutions found with computer help. It is an open problem whether Beckett-Gray codes exist for $n \geq 9$. | P9 | This problem has a 'dynamic' flavor, i.e., the way that the cycle can move through the n-cube depends on the recent history, similarly to the aforementioned order-preserving path problem by Felsner and Trotter, which makes both problems hard.

3.9. Transition graphs.

We let P_n, C_n, K_n and $K_{k,n-k}$ denote the path, cycle, complete graph and complete bipartite graph on the vertex set $[n]$, where the partition classes of $K_{k,n-k}$ have size k and $n - k$. We consider the *transition sequence* of a Gray code, i.e., the cyclic sequence $\delta_1, \ldots, \delta_{2^n}$ of bit positions that change along the code. For instance, starting at 0000 the BRGC Γ_4 shown in Figure 1 (a) has the transition sequence 1213121412131214 (the rightmost bit flips in every second step, and the leftmost bit flips only twice). Such a transition sequence defines a graph G on the vertex set $[n]$, by adding all edges $\{\delta_i, \delta_{i+1}\}$ for $i = 1, \ldots, 2^n$, where indices are taken modulo 2^n. We refer to a Gray code yielding a particular graph G as a *G-code*. Note that G is always a connected graph, and the BRGC is a $K_{1,n-1}$-code. The problem of which n-vertex graphs G admit a G-code was first raised by Slater [Sla79] for the case when $G = P_n$. There is | P10 | a P_n-code for $n \leq 6$, but none for $n = 7$ [Sla89], and it is open whether there is one for $n \geq 8$. Darko Dimitrov, Petr Gregor and Borut Lužar (personal communication) ran a computer search that excludes the existence of a P_n-code also for the case $n = 8$. It is open whether there is a C_n-code for $n \geq 8$ (there is one for $n = 4$ and none for $n = 3, 6, 7$, and none is known for $n = 5$). The papers [BR96] and [WE02] catalogue all trees G on up to $n = 7$ vertices that do not admit a G-code. Moreover, Bultena and Ruskey [BR96] proved that if G is a tree with diameter 4, then there is a G-code, and if G is a tree with diameter 3, then there is no G-code. They also constructed G-codes such that G is a subgraph of any multipartite graph. Wilmer and Ernst [WE02] answered several open problems from the Bultena-Ruskey paper, by showing that there are G-codes such that G is a tree with arbitrarily large diameter, and that there are G-codes such that G is a subgraph of any two-dimensional grid. Suparta and van Zanten [SvZ08] constructed a K_n-code for any $n \geq 3$; see Figure 2 (i). Moreover, Suparta [Sup17] describes

a $K_{k,n-k}$-code for any $1 \leq k \leq n-1$. Dimitrov, Dvořák, Gregor, and Škrekovski [DDGŠ13] constructed an n-bit G-code such that G is an induced subgraph of the $\lceil \log_2(n) \rceil$-cube. In particular, if n is a power of 2, then this is a $Q_{\log_2(n)}$-code; see Figure 2 (i').

3.10. **Avoiding and including other structures.** If we consider a Gray code as a Hamilton cycle in Q_n for $n \geq 2$, we may ask for cycles that avoid or include certain other structures.

In this direction, Latifi, Zheng and Bagherzadeh [LZB92] showed that for any set F of at most $n - 2$ edges, there is a Hamilton cycle in Q_n, $n \geq 2$, that avoids F, and clearly the bound $n - 2$ is best possible. Furthermore, Chan and Lee [CL91] showed that Q_n, $n \geq 3$, has a Hamilton cycle that avoids a given set F of at most $2n - 5$ edges, provided that each vertex is incident to at least two edges not in F, and the bound $2n - 5$ is best possible. Moreover, Dimitrov, Dvořák, Gregor and Škrekovski [DDGŠ09] showed that Q_n has a Hamilton cycle that avoids a given perfect matching M if and only if $Q_n \setminus M$ is connected.

Instead of forbidding edges, one may also consider the problem of forbidding vertices. To this end, Fink and Gregor [FG12] proved that for any set F of at most $\binom{n}{2} - 2$ vertices of Q_n, $n \geq 3$, there is a cycle of length at least $2^n - 2|F|$ that avoids all vertices in F. The bound $\binom{n}{2} - 2$ is best possible. Note also that if F contains only vertices of the same parity, then any extending cycle will also skip $|F|$ vertices of the opposite parity, so $2|F|$ vertices in total.

Dvořák [Dvo05] showed that any set of at most $2n - 3$ edges that together form disjoint paths in Q_n can be extended to a Hamilton cycle, and this bound is best possible. Note that any Hamilton cycle in Q_n can be partitioned into two perfect matchings. Conversely, Kreweras [Kre96] conjectured that every perfect matching in Q_n can be extended to a Hamilton cycle (via a second perfect matching). More generally, Ruskey and Savage [RS93] asked whether any matching (not necessarily perfect) extends to a Hamilton cycle; see also [JSH09, Problem 487]. Kreweras' conjecture was answered affirmatively by Fink [Fin07], and his beautiful and concise proof was later refined by Gregor [Gre09]. More recently, Fink [Fin19] proved that every (not necessarily perfect) matching of Q_n extends to a cycle factor, i.e., to a collection of disjoint cycles that together visit all vertices. The original Ruskey-Savage conjecture is still open, and constitutes one of the most intriguing problems in this area.

A *symmetric chain* in Q_n is a path $x_k, \ldots, x_{n-k}$ where x_i has weight i for all $i = k, \ldots, n-k$, and a *symmetric chain decomposition* is a partition of all vertices of Q_n into symmetric chains. This concept was pioneered by Greene and Kleitman [GK76] (see also [dBvETK51, Kat66, Aig73, WW77]). Streib and Trotter [ST14] constructed a Hamilton cycle in Q_n that contains a symmetric chain decomposition; see Figure 2 (j), where the chains are highlighted on the side. Gregor, Mička and Mütze [GMM23b] showed that the aforementioned symmetric chain decomposition due to Greene and Kleitman can be extended to a Hamilton cycle of Q_n; see Figure 2 (j'). A Hamilton cycle that extends a symmetric chain decomposition has the minimum number of *peaks* and *valleys*, which are triples of consecutive vertices x_i, x_{i+1}, x_{i+2} where x_i and x_{i+2} have the same weight. It is interesting to contrast this with the aforementioned monotone path due to Savage and Winkler, which has the maximum number of peaks and valleys. The authors of [GMM23b] also conjectured that every symmetric chain decomposition of Q_n can be extended to a Hamilton cycle, but a counterexample in Q_6 can be constructed easily: We start with the Greene-Kleitman decomposition, and we consider the chains $C_1 = ------$, $C_2 = --01--$, $C_3 = -0011-$ and $C_4 = 000111$ of lengths 6, 4, 2, and 0, respectively, where 0s and 1s represent fixed bits, and $-$ represents a bit that changes from 0 to 1 when moving up the chain, with the flips occurring from left to right. We remove the middle vertex of C_i and reconnect the remaining two pieces of the chain C_i via the middle vertex of C_{i+1} for $i = 1, 2, 3$. This yields a modified symmetric

chain decomposition of Q_6 in which the former middle vertex of C_1, namely 111000, is a chain of length 0, with the additional property that all of its neighbors are inner vertices of another chain, i.e., this length 0 chain cannot be connected to any other chain along a Hamilton cycle.

The results of Aubert and Schneider [AS82] imply that all edges of Q_n can be partitioned into Hamilton cycles and possibly a perfect matching (see [ABS90]). Saad and Schultz [SS88] proved that Q_n contains a cycle of length ℓ for every even number ℓ from 4 to 2^n. Simmons [Sim78] first proved that Q_n has a Hamilton path between any two prescribed vertices of opposite parity, i.e., it is Hamilton-laceable. This was later generalized by Dvořák and Gregor [DG07], who showed that any two prescribed vertices u and v of opposite parity can be joined by a Hamilton path containing at most $2n - 4$ prescribed edges that together form disjoint paths, if none of the paths contains u or v internally or joins u and v. The bound $2n - 4$ is again best possible. More results about cycles in the n-cube satisfying various additional constraints are surveyed by Xu and Ma [XM09]. For some older results about the n-cube, we refer the reader to the survey [HHW88] by Harary, Hayes and Wu.

3.11. **The number of Gray codes.** Let $H(n)$ denote the number of Hamilton cycles in the n-cube, where we do not distinguish the starting vertex or direction of the cycle, but the names of the n coordinate directions matter (factoring them out one needs to divide by $n!/2$). For $n = 1, \ldots, 6$ the exact values of $H(n)$ are $1, 1, 6, 1344, 906\,545\,760, 35\,838\,213\,722\,570\,883\,870\,720$ [OEIS A066037], where the last value was computed by Haanpää and Östergård [HÖ14]. The number of Hamilton cycles is closely related to the number of perfect matchings in the n-cube, which we denote by $M(n)$ and which is known exactly for all $n \leq 7$ [OEIS A005271]. The results of Bregman [Bre73] and Clarke, George, and Porter [CGP97] imply that the quantity $m(n) := M(n)^{1/2^n}$ grows like $m(n) = (\frac{n}{e}(1 + o(1)))^{1/2}$. Clearly, we have $H(n) \leq \binom{M(n)}{2}$, which shows that the quantity $h(n) := H(n)^{1/2^n}$ satisfies $h(n) \leq \frac{n}{e}(1 + o(1))$. The best known lower bound is $h(n) \geq \frac{n}{4e} + O(\log^2 n) = \frac{n}{e}(1/4 + o(1))$, and it is due to Knuth [Knu11, Ex. 47 in Sec. 7.2.1.1]. Earlier, slightly worse lower bounds were proved by Perezhogin and Potapov [PP01], and by Feder and Subi [FS09]. These results imply that $H(n) = M(n)^{2-o(1)}$, i.e., the number of Hamilton cycles essentially grows quadratically in the number of perfect matchings. The most interesting open question here is to tighten the bounds $[\frac{1}{4}, 1]$ for the constant λ in $h(n) = \frac{n}{e}(\lambda + o(1))$. [P12]

3.12. **Single-track Gray codes.** *Single-track Gray codes* were discovered by Hiltgen, Paterson, and Brandestini [HPB96]. In such a code, any two of the n columns of bits differ only in a cyclic shift. As before, any two consecutive rows/bitstrings differ in a single bit. Clearly, the single-track property implies that the Gray code is balanced, i.e., each bit is flipped the same number of times. Single-track Gray codes are useful for measuring angular positions, using only a single column of bits to encode the different positions, and n appropriately rotated reading heads along this column; see Figure 2 (k), where the shifted columns are shaded and the shifts are indicated by horizontal bars. In this example, 16 bitstrings are missing from the code. Indeed, for $n \geq 3$ the single-track property does not allow to visit all 2^n bitstrings, but only a subset of size $N < 2^n$. In the aforementioned paper, Hiltgen, Paterson, and Brandestini show that $2n$ must divide N for such a code to exist. Moreover, they provide constructions of single-track Gray codes of length $N = kn$, for all $n \geq 4$ and every even $2 \leq k \leq 2^{n-1-\lceil\sqrt{2(n-3)}\rceil}$. Etzion and Paterson [EP96] constructed single-track Gray codes of length $N = 2^n - 2n$ for the special case when n is a power of 2. Later, Schwartz and Etzion [SE99] proved that this is the maximum length of a cycle one can achieve under these conditions. Their result shows that the single-track cycle of length $2^8 - 2 \cdot 8 = 240$ shown in Figure 2 (k) is in fact longest possible. Etzion [Etz07] [P13]

gives a short survey on the state-of-the art on single-track Gray codes, and mentions as the most interesting open problem the question whether for prime $n \geq 3$ there is a single-track Gray code of length $N = 2^n - 2$, which would be optimal. This has been tested and confirmed for all prime $n \leq 19$. A natural approach to construct such a Gray code is to find a Gray cycle through all n-bit necklaces except 0^n and 1^n, which are aperiodic as n is prime, so the number of such necklaces is N/n, selecting one representative from each necklace arbitrarily, in such a way that from the last representative a single flip leads to a shifted version of the first representative. Applying the transition sequence of such a Gray cycle n times, each time shifted appropriately, produces the desired single-track Gray code. The open problem mentioned in Section 3.17 seeks a very similar Gray path through all n-bit necklaces (starting and ending in 0^n and 1^n, respectively).

More generally, in a *t-track Gray code*, the n columns of bits can be partitioned into t groups such that the columns in each group differ only in a cyclic shift. Gregor, Merino and Mütze [GMM24] constructed a t-track Gray code of all 2^n bitstrings of length n (no bitstring is missed) for every n that is a sum of $t \geq 2$ powers of 2. In particular, if n is a sum of two powers of 2, then this is a 2-track Gray code; see Figure 2 (k'). Furthermore, these results also show that for every dimension $n \geq 5$ there is a t-track Gray code with $t \leq \lfloor \log_2(n+1) \rfloor$, i.e., at most logarithmically many tracks.

3.13. Symmetric Gray codes.
A Hamilton cycle $(x_1, \ldots, x_N)$ of Q_n, where $N = 2^n$, is called *k-symmetric*, if the mapping $x_i \rightarrow x_{i+N/k}$ for $i = 1, \ldots, N$ is an automorphism of Q_n (indices are considered modulo N). In other words, when arranging the vertices in the order of the Hamilton cycle equidistantly on a circle and drawing the edges of Q_n as straight lines, then the resulting drawing has k-fold rotational symmetry. Gregor, Merino and Mütze [GMM24] constructed a k-symmetric Hamilton cycle of Q_n for $k := 2^{\lceil \log_2 n \rceil}$, and they showed that this factor k is largest possible. Clearly, if n is a power of 2, then this optimum factor equals $k = n$. Such an 8-symmetric Hamilton cycle for $n = 8$ is shown in Figure 2 (k'); the corresponding automorphism is $f(x_1, \ldots, x_8) = (x_2, x_3, x_4, \overline{x_1}, x_6, x_7, x_8, \overline{x_5})$.

3.14. Shifts and transpositions.
Feldmann and Mysliewietz [FM96] constructed a Gray code for all bitstrings of length n which in every step either flips the leftmost bit or shifts all bits cyclically one position to the left or right.

Savage, Shields and West [SSW03] consider a Gray code for all bitstrings of length n which in every step either flips the leftmost bit or transposes two adjacent bits $01 \leftrightarrow 10$. Clearly, the listing must start and end with 0^n and 1^n, respectively, as these two only admit flipping the leftmost bit. The authors constructed such a Gray code for all $n > 5$ for which $\binom{n+1}{2}$ is odd, and showed that it does not exist otherwise due to balancedness problems. Figure 2 (l) shows their construction for $n = 6$. Observe that such a Gray code corresponds to a Hamilton path in the cover graph of the poset of all lattice paths of length n with steps $(+1, +1)$ or $(+1, -1)$ (encoded by 1 and 0, respectively) ending at the fixed point $(n, 0)$ and ordered by the pointwise-below relation, which was introduced by Stanley in a slightly different context. In this setting, the aforementioned results were considerably strengthened by Tasoulas, Manes and Sapounakis [TMS24], who showed that the cover graphs of various intervals in this poset all admit Hamilton cycles.

3.15. Coils and snakes.
A *coil-in-the-box* is an induced cycle in the n-cube, i.e., a cycle without chords. Clearly, such a cycle cannot visit all 2^n vertices, and the problem is to determine the maximum length $c(n)$ of a coil-in-the-box for each dimension n. For $n = 1, \ldots, 8$, the exact values of $c(n)$ are known to be $0, 4, 6, 8, 14, 26, 48, 96$ [OEIS A000937], and an optimal solution for $n = 8$ is shown in Figure 3 (a). In the literature one also encounters induced paths in the

n-cube, which we call *snakes-in-the-box* (in some papers, the term snake is used to refer to an induced cycle, not to an induced path). The maximum length of a snake is denoted $s(n)$ and for $n = 1, \ldots, 8$ the exact values of $s(n)$ are $1, 2, 4, 7, 13, 26, 50, 98$ [OEIS A099155]. Clearly, we have $s(n) \geq c(n) - 2$, but this leaves open whether $s(n)$ dominates $c(n)$ or vice versa for larger values of n. Kautz [Kau58] discovered coil-in-the-box and snake-in-the-box codes for single bit error detection in digital circuits, and he established simple lower and upper bounds for $c(n)$. He also determined the exact values of $c(n)$ for $n \leq 5$. Subsequently, Davies [Dav65], Kochut [Koc96], and Östergård and Pettersson [ÖP14] established the optimality of various coils for $n = 6, 7, 8$, respectively, by exhaustive search. The candidate of length 96 for $n = 8$ shown in Figure 3 (a) had already been found by Paterson and Tuliani [PT98]. There was also much effort towards proving good asymptotic bounds for $c(n)$ as n grows. The best known upper bound is due to Zémor [Zém97], who showed that

$$c(n) \leq 1 + 2^{n-1} \frac{6n}{6n + \frac{1}{6\sqrt{6}}\sqrt{n} - 7} \leq 2^{n-1}\Big(1 - \frac{\sqrt{n}}{89} + O\big(\frac{1}{n}\big)\Big).$$

for all $n \geq 1$. This bound improves upon earlier results due to Douglas [Dou69], Deimer [Dei85], and Snevily [Sne94]. As far as lower bounds are concerned, Evdokimov [Evd69] first established that $c(n) \geq \lambda 2^{n-1}$ for $\lambda := c(8)/256 = 0.375$, i.e., there are coils that miss only a constant fraction of all vertices of the n-cube, a fact that is maybe not apparent from the first few values of $c(n)$. Wojciechowski [Woj89] independently proved a similar bound, but with a slightly worse (i.e., smaller) constant. The best known lower bound is due to Abbott and Katchalski [AK91], who showed that one can take $\lambda := 77/128 > 0.601$ for $n \geq 2$, following up on their earlier paper [AK88]. Consequently, the most interesting open problem is whether $c(n) \leq \lambda 2^{n-1}$ for some constant $\lambda < 1$ (and large n). Evdokimov summarizes many of these results in his survey [Evd14] (in Russian). P15

3.16. **Factor-avoiding strings.** For any fixed bitstring τ, we consider all bitstrings of length n that *avoid τ as a factor*, i.e., that do not contain τ as a contiguous substring.

We say that a string τ is *bifix-free* if none of its proper prefixes equals one of its proper suffixes. Squire [Squ96] showed that if τ is bifix-free, or if $\tau = \sigma\sigma \cdots \sigma$ for some bifix-free string σ, then for all $n \geq 1$ there is a Gray code for all bitstrings of length n that avoid τ. If τ does not satisfy these conditions, then for infinitely many $n \geq 1$ there are balancedness problems that prevent the existence of a Gray code. Liu, Hsu and Chung [LHC94] constructed Gray codes for *Fibonacci words*, i.e., bitstrings that avoid the factor $\tau = 1^\ell$ for some fixed integer $\ell \geq 2$. Subsequently, Vajnovszki [Vaj01] presented a simple genlex Gray code for Fibonacci words; see Figure 3 (b) and also [MS05]. Chinburg, Savage and Wilf [CSW99] consider bitstrings in which any two 1s are at least $d \geq 2$ positions apart. For $d = 2$ these are precisely Fibonacci words (with $\ell = 2$), and the paper describes the same genlex Gray code as Vajnovszki for this case. For $d \geq 3$, they showed that balancedness problems rule out the existence of a (1-)Gray code, and they construct a 2-Gray code that uses single bitflips or transpositions of a 0 and 1; see also [BMdS09]. Baril, Kirgizov and Vajnovszki [BKV22] consider bitstrings in which every maximal run of b consecutive 1s that is not a suffix is followed by a run of strictly more than b/q consecutive 0s, where $q \geq 1$ is some integer parameter. They call these strings q-decreasing words, and show that they are in bijection to Fibonacci words with $\ell = q + 1$. Furthermore, they present a 3-Gray code for any $q \geq 1$, shown in Figure 3 (e11') for $n = 8$ and $q = 1$, and a 1-Gray code for $q = 1$. They also conjecture that a 1-Gray code exists for all $q \geq 1$ and $n \geq 1$. P16

Baril and Vajnovszki [BV05] constructed a genlex Gray code for *Lucas words*, i.e., bitstrings that avoid the factor $\tau = 1^\ell$ for some $\ell \geq 2$, where factors that cyclically wrap around the
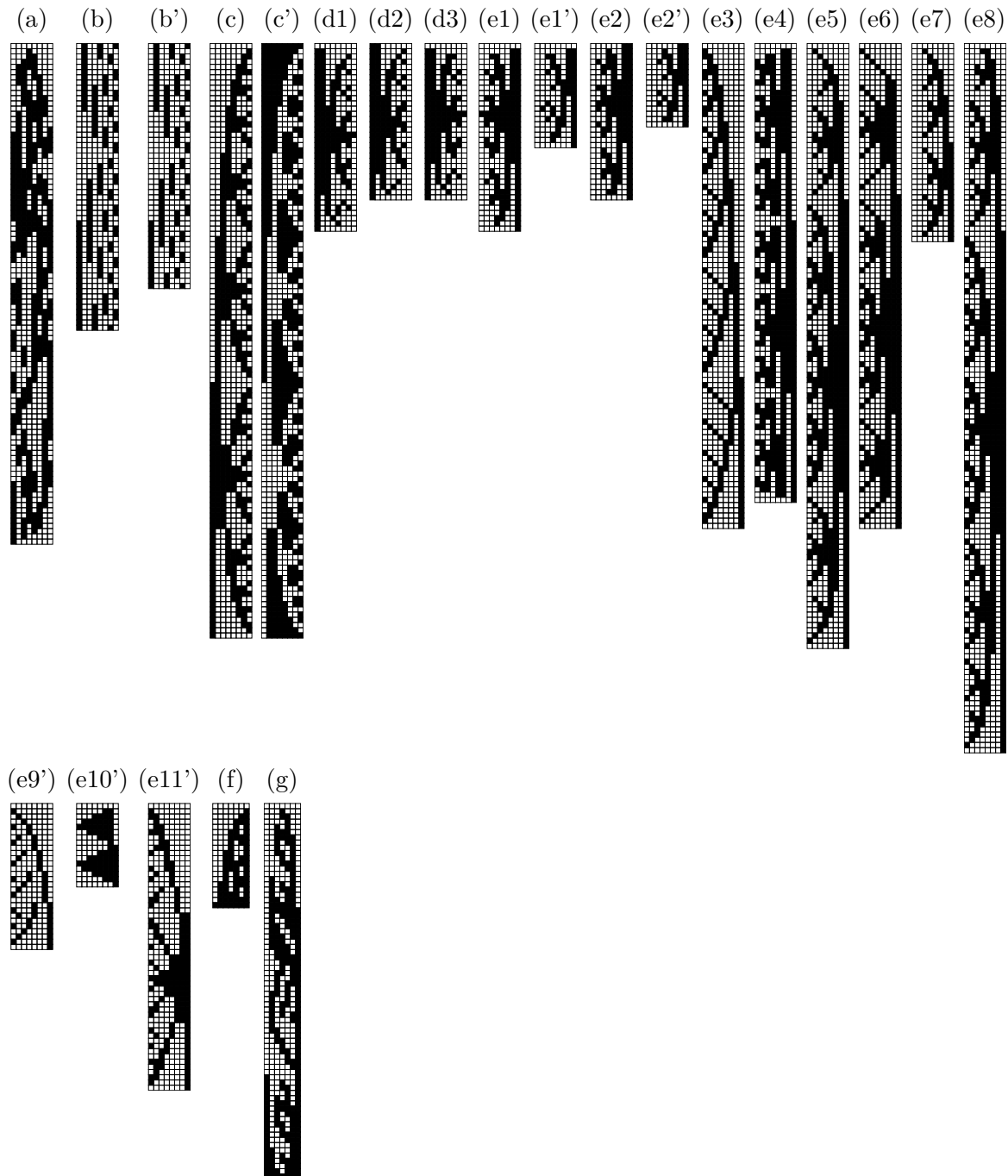
Figure 3. Restricted binary Gray codes (1-bits black, 0-bits white): (a) coil-in-the-box (length 96); (b) 11-avoiding (Fibonacci words); (b') 11-avoiding circularly (Lucas words); (c) 101-avoiding; (c') 010-avoiding; (d1) necklaces (lex. largest); (d2) aperiodic necklaces (lex. largest); (d3) bracelets (lex. largest); (e1) necklaces (lex. smallest); (e1') unlabeled necklaces; (e2) Lyndon words (=lex. smallest aperiodic necklaces); (e2') unlabeled Lyndon words; (e3) weight ≤ 3; (e4) 100-avoiding; (e5) ≤ 6 inversions w.r.t. 0^*1^*; (e6) ≤ 1 transpositions w.r.t. 0^*1^*; (e7) ≤ 00100101; (e8) $\leq$ their reversal (lex. smallest neckties); (e9') {11, 101}-avoiding; (e10') {100, 101}-avoiding; (e11') 1-decreasing; (f) necklaces (repr. in BRGC order); (g) neckties (lex. smallest).

boundary are also avoided; see Figure 3 (b'). It is a 1-Gray code if $\ell + 1$ does not divide n, and a 2-Gray code otherwise. In the latter case, a 1-Gray code does not exist because of balancedness problems.

The subgraph of the n-cube induced by Fibonacci words or Lucas words for $\ell = 2$ is known as *Fibonacci cube* [Hsu93] or *Lucas cube* [MPCZS01], respectively, and the number of those words are given by the Fibonacci numbers [OEIS A000045] or Lucas numbers [OEIS A000204].

Bernini et al. [BBP$^+$15] showed that for 'almost all' factors τ, the sublist of the BRGC induced by all τ-avoiding words is a 3-Gray code for all $n \geq 1$; see Figure 3 (c). Specifically, this holds for every τ that is not a suffix of $\cdots (110^\ell)(110^\ell)0$ with at least two 1s for some $\ell \geq 0$. This is a surprisingly general result, given that for every possible length ℓ of the forbidden factor, all but $\ell - 2$ factors $\tau \in \{0, 1\}^\ell$ satisfy this condition. These Gray codes are all cyclic, except if $\tau = 0^\ell$ for $\ell \geq 2$ or $\tau = 10^\ell$ for $\ell \geq 1$, in which case they are non-cyclic 1-Gray codes. For $\tau = 0^\ell$ these are complements of Fibonacci words, so we recover Vajnovszki's aforementioned result. Moreover, for $\tau \in \{0^\ell 1 \mid \ell \geq 1\}$ they are (cyclic) 2-Gray codes. As the BRGC is genlex, all these sublist Gray codes are also genlex.

Note that any factor τ that ends with 1 trivially satisfies the aforementioned condition of Bernini et al. On the other hand, if τ ends with 0, then we may simply consider the complemented factor $\overline{\tau}$, which does ends with 1. Consequently, we get a 3-Gray code for $\overline{\tau}$, and complementing all of its bitstrings we obtain a 3-Gray code for τ; see Figure 3 (c'). Combining these observations we see every factor τ admits a 3-Gray code for all τ-avoiding bitstrings.

Many of these results were later generalized substantially via flip languages discussed in Section 3.18. In that section, we will consider the BRGC with each bitstring reversed, which corresponds to factors starting (and not ending) with a particular symbol.

3.17. **Necklaces and relatives.** In this section we consider equivalence classes of bitstrings under shifts, reversal and/or complementation; see Figure 4. A *necklace* is an equivalence class of bitstrings under cyclic shifts. A necklace of n-bit strings is called *aperiodic* if it has size n, i.e., all n cyclic shifts are distinct strings. The lexicographically smallest representatives of aperiodic necklaces are called *Lyndon words*. A *necktie* is an equivalence class of bitstrings under reversal, and a *bracelet* is an equivalence class under cyclic shifts and reversal. The goal is to find a Gray code that visits each equivalence class exactly once, where we may also impose restrictions on the choice of representatives, such as for example the lexicographically smallest one from each class.

Vajnovszki [Vaj07] showed that the sublists of the BRGC given by lexicographically largest representatives of necklaces, aperiodic necklaces or bracelets form a cyclic 3-Gray code; see Figure 3 (d1)–(d3). If we are interested in the lexicographically smallest representatives, we can apply complementation to the strings in those Gray codes. Weston and Vajnovszki [WV06] generalized these results for necklaces and aperiodic necklaces to larger alphabets. It turns out we can do even better by reversing each string in the BRGC. While this does not change the order of the (reversed) strings among each other, we will change the choice of representatives, which results in a different sublist. Specifically, Vajnovszki [Vaj08] showed that the sublist approach applied to the BRGC with reversed strings gives a cyclic 2-Gray code for lexicographically smallest representatives of necklaces and aperiodic necklaces, in particular for Lyndon words; see Figure 3 (e1)+(e2). He also obtains a cyclic 3-Gray code for subsets of these two classes, namely for strings that are lexicographically smallest under cyclic shifts and complementation, which he calls *unlabeled necklaces* and *unlabeled Lyndon words*; see Figure 3 (e1')+(e2'). As all of these Gray codes are sublists of the BRGC, they are clearly genlex. Sawada, Williams and Wong [SWW17] gave a constant amortized-time algorithm for Vajnovszki's Gray codes.
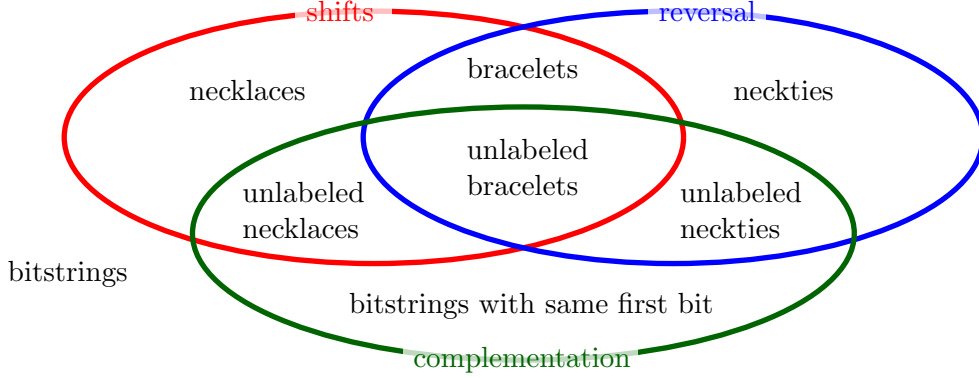
FIGURE 4. Venn diagram of combinatorial objects obtained from bitstrings under shifts, reversal and/or complementation.

The aforementioned results for necklaces, Lyndon words and their unlabeled variants can be generalized substantially via flip languages and flip-swap languages discussed in the next section.

For even values of n, a 1-Gray code for necklaces does not exist due balancedness problems (regardless of the choice of representatives); see [WS96]. However, for odd values of n this problem is still open: Is there a 1-Gray code for necklaces of length n for odd n? Note that 0^n and 1^n must be the first and last necklaces in such a Gray code, so the code cannot be cyclic. The choice of representatives in this problem is arbitrary, ideally of course they are the lexicographically smallest ones. Savage conjectured [Sav97] that the answer to this question is 'yes' (even in the more general setting of necklaces of b-ary strings for any $b \geq 2$, when the flip operation is to change one entry by ± 1). For prime values of n, Degni and Drisko [DD07] conjectured a very simple choice of representatives and ordering, which they checked works for all primes $n \leq 37$. Specifically, they consider the sublist of the BRGC obtained by choosing for each necklace the first representative in this list. Figure 3 (f) shows this 1-Gray code for $n = 7$. Note that for prime n, all necklaces except 0^n and 1^n are aperiodic. As explained in Section 3.12, certain cyclic 1-Gray codes for all necklaces except 0^n and 1^n would yield optimal single-track Gray codes for prime n.

Wang [Wan93] proved that there is a Gray code for the lexicographically smallest representatives of neckties if and only if $n \geq 3$ is odd. A simple inductive construction of a necktie Gray code N_n for odd n is given by $N_1 := 0, 1$ and $N_n := 0N_{n-2}0, 0\operatorname{rev}(M_{n-2})1, 1N_{n-2}1$, where M_n is any Gray code for all bitstrings of length n starting with 0^n and ending with 1^n, for example the aforementioned monotone Gray code due to Savage and Winkler; see Figure 3 (g).

[P18]   No 2- or 1-Gray code is known for bracelets; see [Sav97]. Balancedness problems rule out a 1-Gray code for even n.

[P19]   Is there a Gray code for unlabeled neckties, i.e., strings under reversal and complementation, or for unlabeled bracelets, i.e., strings under cyclic shifts, reversal and complementation?

3.18. **Flip-swap languages.** A *flip-swap language* is a subset S of bitstrings of length n such that S is closed under the following two operations (when applicable): (i) flipping the leftmost 1; (ii) transposing the leftmost 1 with the bit to its right. If operation (ii) is omitted, we call S a *flip language*.

Examples of flip-swap languages include all bitstrings, necklaces (lex. smallest representatives) and Lyndon words, strings with weight $\leq \ell$, strings that avoid the factor 10^ℓ for some fixed integer $\ell \geq 1$, strings with $\leq \ell$ inversions w.r.t. 0^*1^*, strings with $\leq \ell$ transpositions w.r.t. 0^*1^*, strings that are lexicographically $\leq$ some fixed string, and strings that are $<$ or $\leq$ their reversal

(in particular, lex. smallest neckties). The number of *inversions w.r.t.* 0^*1^* of a bitstring is the number of pairs of bits (x_i, x_j) where x_i is left of x_j and $(x_i, x_j) = (1, 0)$, and the number of *transpositions w.r.t.* 0^*1^* is the minimum number of exchanges of a 1 and 0 so that all 0s are left of all 1s.

Clearly, every flip-swap language is also a flip language. Examples of flip languages that are not already mentioned before are unlabeled necklaces (lex. smallest representatives) and unlabeled Lyndon words, strings that avoid any factor τ starting with 1, and q-decreasing words discussed in Section 3.16 for any $q \geq 1$.

Vajnovszki [Vaj07] proved that for the BRGC with reversed bitstrings, the sublist induced by the strings of any flip language forms a cyclic 3-Gray code; see Figure 3 (e1'), (e2') and (e9')–(e11'). Sawada, Williams and Wong [SWW22] strengthened this by showing that the sublist induced by the strings of any flip-swap language forms a cyclic 2-Gray code; see Figure 3 (e1)–(e8).

Note that complementing a flip-swap language 2-Gray code yields a 2-Gray code for the complemented flip-swap language, for example strings with weight $\geq \ell$, necklaces and aperiodic necklaces with lexicographically largest representatives, strings that avoid 01^ℓ, strings that are $\geq$ some fixed string, and strings that are $>$ or $\geq$ their reversal (in particular, lex. largest neckties). Similarly, complementing a flip language 3-Gray code yields a 3-Gray code for the complemented flip language, for example unlabeled necklaces and aperiodic necklaces with lexicographically largest representatives, and strings that avoid any factor τ starting with 0.

Observe furthermore that for any two flip(-swap) languages S and S', the intersection $S \cap S'$ and the union $S \cup S'$ are also flip(-swap) languages. For example, Lyndon words that are $\leq$ their reversal are a flip-swap language. Similarly, words avoiding any number of distinct factors all starting with 1 are a flip language. As an application, recall that Chinburg, Savage and Wilf [CSW99] considered bitstrings in which any two 1s are at least $d \geq 2$ positions apart. Those are characterized by avoiding the factors $\{11, 101, 1001, \ldots, 10^{d-2}1\}$, which is a flip language, so we immediately obtain a 3-Gray code for this problem, shown in Figure 3 (e9') for $d = 2$ (as mentioned before, this can be improved by a 2-Gray code tailored to this problem).

Taking intersections of flip languages enables us to consider avoidance of factors that contain wildcard symbols $*$, which can match both 0 or 1, also known *partial words*. For example, the factor $10*$ stands for '10 followed by another arbitrary symbol', and it can be modelled by avoiding both $\{100, 101\}$; see Figure 3 (e10').

3.19. **Larger alphabets.** The definition of BRGC can be generalized straightforwardly to larger alphabets, which yields a listings of all b-ary strings of length n where only one of the entries changes by ± 1 at a time: $\Gamma_0 := \varepsilon$ and $\Gamma_{n+1} := 0\Gamma_n, 1\operatorname{rev}(\Gamma_n), 2\Gamma_n, 3\operatorname{rev}(\Gamma_n), \ldots, (b-1)\Gamma'$ for $n \geq 0$, where $\Gamma' := \operatorname{rev}(\Gamma_n)$ if b is even and $\Gamma' := \Gamma_n$ otherwise; see [Flo56, Coh63, SK78, Er84, BB95, BBKA95]. Recently, Herter and Rote [HR18] devised loopless algorithms for generating b-ary Gray codes based on generalizations of the Tower of Hanoi puzzle. Flahive [Fla08] describes a construction of balanced b-ary Gray codes with transitions counts that differ by at most 2 from the average b^n/n, improving upon the earlier paper by Flahive and Bose [FB07]. Many other of the properties discussed before for the binary case (weight-monotonicity, runs lengths, trend-freeness, single-track etc.) are still unexplored for the general b-ary case.

P20

A *signed binary representation* (SBR) of an integer N is a string $(a_{n-1}, \ldots, a_1, a_0)$ over the alphabet $\{-1, 0, 1\}$ such that $N = \sum_{i=0}^{n-1} a_i 2^i$. An SBR is said to be *minimal* if the number of nonzero digits is minimum. Some values of N have only a single minimal SBR, whereas other values admit an exponential (in n) number of minimal SBRs. Sawada [Saw07] developed a genlex 3-Gray code for listing all minimal SBRs of an integer N.

4. COMBINATIONS

Combinations are the subsets of the ground set $[n]$ of fixed size k, where $0 \leq k \leq n$. We sometimes refer to them as (n, k)-*combinations*, and their number is $\binom{n}{k}$. In a computer an (n, k)-combination can be conveniently represented by its characteristic vector $x = (x_1, \ldots, x_n)$, i.e., $x_i = 1$ if i is in the set and $x_i = 0$ otherwise. Clearly, x contains precisely k many 1s, i.e., x has weight k.

In this section, we discuss Gray codes for combinations that use different flip operations, as well as Gray codes for restricted classes of combinations, including necklaces and Catalan objects (the latter can be seen as special $(2k, k)$-combinations). Lastly, we consider more general Gray codes that list bitstrings for certain admissible weight values and weight ranges, and also some generalizations and variants of combinations.

4.1. **Transpositions.** We first focus on transposition Gray codes or *revolving door* Gray codes for combinations, i.e., any two consecutive combinations differ by swapping a 0 and 1 in the bitstring representation. Tang and Liu [TL73] showed that restricting the binary reflected Gray code (BRGC) for bitstrings of length n to any fixed weight k yields a cyclic transposition Gray code for (n, k)-combinations; see Figure 5 (a). The resulting listing is genlex, and its ordered set representation has the interesting property that for any sublist with the same fixed prefix the next larger element is always non-decreasing or non-increasing, alternatingly depending on the parity of the prefix length, which leads to an efficient generation algorithm; see [PI79] and [Knu11, Sec. 7.2.1.3]. Bitner, Ehrlich and Reingold [BER76] devised a loopless algorithm to generate this ordering. The flip graphs of (n, k)-combinations under arbitrary transpositions are known as *Johnson graphs* $J(n, k)$, which were shown to be Hamilton-connected by Jiang and Ruskey [JR94] and by Knor [Kno94]; see Figure 1 (b) and also [Als13].

Chase [Cha70a] and Eades and McKay [EM84] constructed so-called *homogeneous* Gray codes for combinations, which use only transpositions of the form $00\cdots01 \leftrightarrow 10\cdots00$, i.e., the bits between the swapped 0 and 1 are all 0s; see Figure 5 (b). We can think of this as an ordering that plays all possible combinations of k keys out of n available keys on a piano, without ever crossing any fingers. Their homogeneous Gray code also has the genlex property. Chase [Cha89] showed that we can restrict the allowed swaps further and only allow transpositions of the form $01 \leftrightarrow 10$ or $001 \leftrightarrow 100$, and this listing is also genlex and can be implemented efficiently; see Figure 5 (c) and also [Rus93, JM95].

Buck and Wiedemann [BW84] and Eades, Hickey and Read [EHR84] independently proved that all (n, k)-combinations can be generated by using only adjacent transpositions $01 \leftrightarrow 10$ if and only if $k \in \{0, 1\}$ or $n - k \in \{0, 1\}$ or k and $n - k$ are both odd; see Figure 5 (d). Note that such a Gray code must start and end at the combinations $1^k 0^{n-k}$ and $0^{n-k} 1^k$, as these admit only a single transposition (they are degree-1 vertices in the corresponding flip graph). An efficient algorithm to compute the Eades-Hickey-Read ordering was provided by Hough and Ruskey [HR88], and an efficient algorithm for computing a distinct adjacent transposition listing was given by Ruskey [Rus88a]; see Figure 5 (d').

Extending earlier partial results of Joichi and White [JW80], Enns [Enn93] showed that (n, k)-combinations can be generated by circularly adjacent transpositions, i.e., including transpositions that swap the first and last bit, if and only if at least one of k or $n - k$ is odd.

Felsner, Kleist, Mütze and Sering [FKMS20] showed that for odd n, all $(n, 2)$-combinations can be generated by transpositions so that each of the $\binom{n}{2}$ possible transpositions is used exactly once. P21 They raised the question where such balanced Gray codes are possible for (n, k)-combinations in general, whenever n is odd and $\binom{n}{2}$ divides $\binom{n}{k}$ (it is easy to argue that for even n such Gray
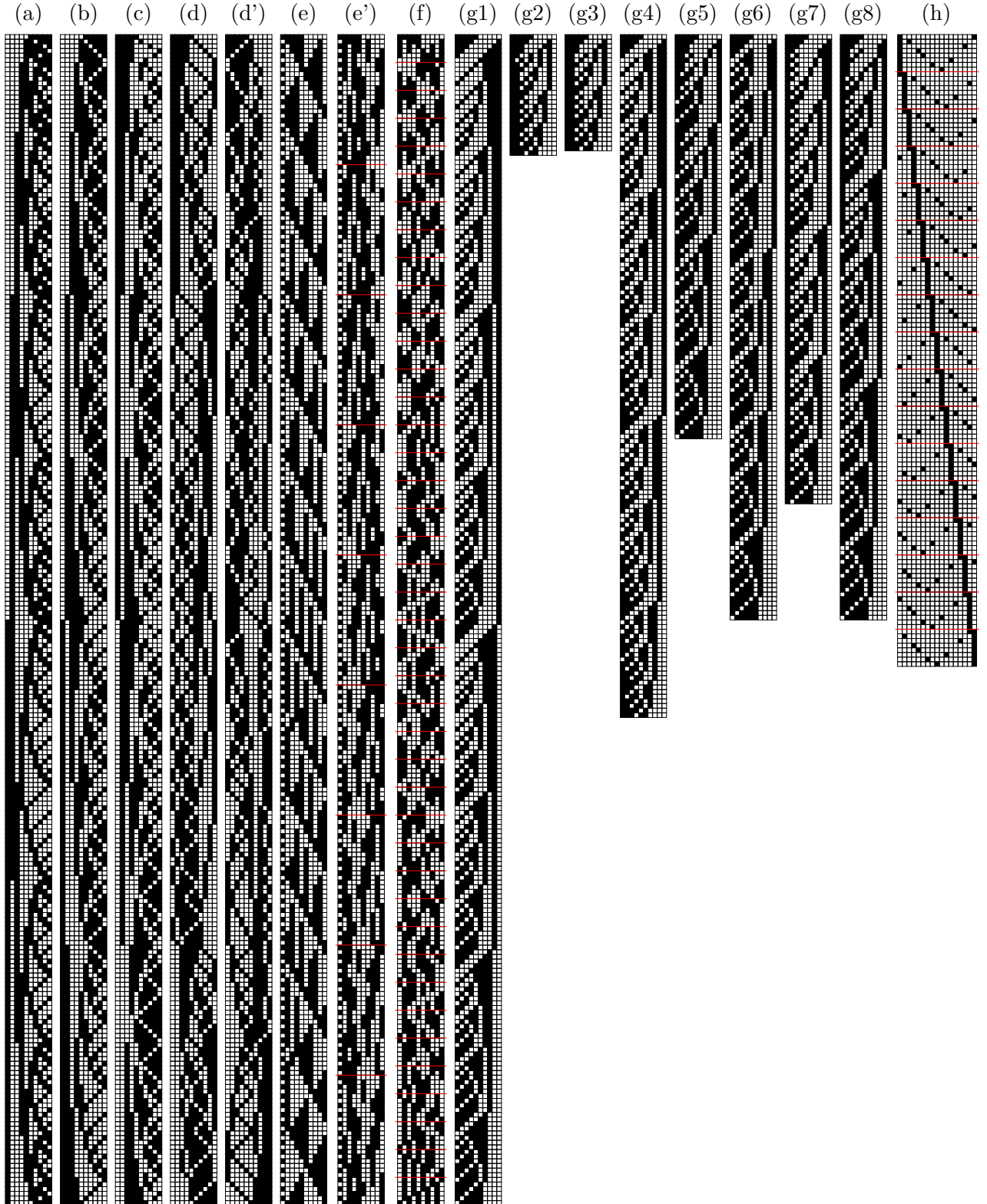
FIGURE 5. Combination Gray codes (1-bits black, 0-bits white): (a) revolving door from BRGC; (b) homogeneous transpositions $00\cdots01 \leftrightarrow 10\cdots00$; (c) transpositions $01 \leftrightarrow 10$ or $001 \leftrightarrow 100$; (d) adjacent transpositions (Eades-Hickey-Read); (d') adjacent transpositions (Ruskey); (e) star transpositions; (e') star transpositions + cyclic blocks; (f) alternating flaws; (g) prefix/substring shifts (cool-lex): (g1) all $(10, 5)$-combinations, (g2) necklaces (lex. largest), (g3) aperiodic necklaces (lex. largest), (g4) 0111-avoiding, (g5) ≤ 10 inversions w.r.t. 1^*0^*, (g6) ≤ 2 transpositions w.r.t. 1^*0^*, (g7) ≥ 1001100010, (g8) $\geq$ their reversal; (h) balanced $(17, 2)$-combinations by transpositions with cyclic blocks.

codes cannot exist). Their balanced Gray code for $(n, 2)$-combinations can be partitioned into n blocks, where each block is a cyclic right-shift of the previous one; see Figure 5 (h). Consequently, it is also a single-track Gray code in which column i is a cyclic shift of the first column by $\frac{1}{n}\binom{n}{2} \cdot (i - 1)$ for all $i = 2, \ldots, n$.

4.2. **Cool-lex order and bubble languages.** Another elegant and efficient method for generating combinations based on prefix shifts was described by Ruskey and Williams [RW09], who named their ordering *cool-lex*, because of its similarities to colex and its numerous striking features. A *prefix shift* cyclically left-shifts a prefix of the bitstring by one position to generate the next combination. The following simple successor rule generates the next combination in the cool-lex ordering: Shift the shortest prefix of length at least 3 that ends with 01, or if no such prefix exists shift the entire string; see Figure 5 (g1) and cf. [SW14]. The cool-lex ordering is cyclic, any two consecutive combinations differ by one or two transpositions, and it admits simple loopless algorithms. Moreover, the ordering is genlex in the ordered set representation (but not in the bitstring representation).

Ruskey, Sawada and Williams [RSW12] present another major application of the cool-lex approach. They consider subsets B of (n, k)-combinations that are *bubble languages*, which means that if a string x is in B and x contains 01, then the string obtained from x by flipping the first such occurrence to 10 must also be in B. Examples of bubble languages include combinations, b-ary Dyck words, necklaces and aperiodic necklaces with lexicographically largest representatives, strings that avoid the factor 01^ℓ for some fixed integer $\ell \geq 1$, strings with $\leq \ell$ inversions w.r.t. 1^*0^*, strings with $\leq \ell$ transpositions w.r.t. 1^*0^*, strings that are lexicographically $\geq$ some fixed string, and strings that are $>$ or $\geq$ their reversal (all of these families are considered for a fixed number k of 1s). The number of *inversions w.r.t.* 1^*0^* of a combination $x = (x_1, \ldots, x_n)$ is the number of pairs of bits (x_i, x_j) with $i < j$ and $(x_i, x_j) = (0, 1)$, and the number of *transpositions w.r.t.* 1^*0^* is the minimum number of exchanges of a 0 and 1 so that all 1s are left of all 0s. The main result of [RSW12] is that the words of any bubble language, when arranged in the order induced by the cool-lex order, form a cyclic Gray code in which any two consecutive strings differ by a left-shift of a substring by one position, which results in one or two transpositions; see Figure 5 (g2)–(g8) and Figure 6 (f2)+(f3). Note that complementing such a listing yields a shift Gray code for the complemented bubble language, for example necklaces and aperiodic necklaces with lexicographically smallest representatives (i.e., fixed-density Lyndon words), strings that avoid 10^ℓ, strings that are $\leq$ some fixed string, and strings that are $<$ or $\leq$ their reversal. Observe furthermore that for any two bubble languages B and B', the intersection $B \cap B'$ and the union $B \cup B'$ are also bubble languages. For example, b-ary Dyck words that avoid 01^ℓ are a bubble language. Following up on these results, Sawada and Williams [SW12a, SW13a] devised constant amortized-time algorithms for computing many of the aforementioned bubble language Gray codes.

P22   Is there a unifying principle behind flip-swap languages and bubble languages (recall Section 3.18)?

P23   Another interesting open question in this context is whether all (n, k)-combinations can be generated by *prefix reversals*, i.e., in each step, a prefix of the bitstring representation is reversed to obtain the next combination. Such orderings can be constructed easily for the cases $k \in \{1, 2\}$ or $n - k \in \{1, 2\}$, but no general construction is known.

4.3. **Fixed-density necklaces.** We now consider equivalence classes of (n, k)-combinations under cyclic shifts, so-called *necklaces with fixed density*. Wang and Savage [WS96] devised a cyclic transposition Gray code, valid for all n and k. The representatives from each equivalence

class are not the lexicographically smallest ones, but those rotated left until the first bit equals 1. Independently, Ueda [Ued00] described another transposition Gray code for necklaces with fixed density, and also a Gray code for aperiodic necklaces with fixed density. However, the chosen representatives in his two Gray codes are non-standard, in particular, they are neither the lexicographically smallest nor largest ones. He also established stronger Hamiltonicity properties about the underlying flip graphs.

Shift Gray codes for fixed density necklaces and aperiodic necklaces, either for lexicographically smallest or largest representatives, i.e., in particular for fixed density Lyndon words, can be derived from the bubble-language framework discussed in the previous section; recall Figure 5 (g2)+(g3). These Gray codes operate by substring shifts that result in one or two transpositions per step.

4.4. **Catalan objects.** *Dyck words* are $(2k, k)$-combinations with the additional property that every prefix contains at most as many 0s as 1s. By interpreting the 1s as opening brackets and the 0s as closing brackets, we see that Dyck words are in bijection with well-formed parenthesis expressions with k pairs of parenthesis. They are also in bijection to preorder sequences of binary trees with k nodes, when recording a 1 for every internal node and a 0 for every leaf (the 0 corresponding to the last leaf being implicit). It is well known that they are counted by the kth Catalan number $C_k = \frac{1}{k+1}\binom{2k}{k}$ [OEIS A000108].

Proskurowski and Ruskey [PR85] devised a transposition Gray code for Dyck words, and in a subsequent paper [RP90], they present a simpler genlex construction for this problem; see Figure 6 (a) and also [Wal98]. They also showed that an adjacent transposition Gray code exists if and only if k is even or $k < 5$; see Figure 6 (c). A homogeneous transposition Gray code for Dyck words, which is also genlex, was given by Bultena and Ruskey [BR98]; see Figure 6 (b). More generally, Roelants van Baronaigien [RvB00] developed a homogeneous transposition Gray code b-*ary Dyck words*, valid for any integer $b \geq 2$; see Figure 6 (d2)+(d3); see also [XUT00]. These are are strings with k many 1s and $k(b-1)$ many 0s such that in every prefix, the number of 0s is at most $b - 1$ times the number of 1s, which are in bijection to preorder sequences of b-ary trees. Vajnovszki and Walsh [VW06] present a genlex Gray code for this problem that is even more restrictive and uses only transpositions $01 \leftrightarrow 10$ or $001 \leftrightarrow 100$; see Figure 6 (e2)+(e3).

Ruskey and Williams [RW08] present a Gray code for Dyck words that uses prefix shifts, which cyclically left-shifts a substring that starts at the second position (the first bit clearly has to always be 1); see Figure 6 (f2). Any two consecutive words in this ordering differ by one or two transpositions, and the ordering has a very simple successor rule and loopless algorithm. Durocher, Li, Mondal, Ruskey and Williams [DLM$^+$12] generalized this construction to higher arity $b \geq 2$; see Figure 6 (f3).

Dyck words can be generalized to *words with e flaws*, where $0 \leq e \leq k$ is an integer parameter, which are $(2k, k)$-combinations with the property that precisely e prefixes ending with 0 have strictly more 0s than 1s. We denote them by $F_{k,e}$. Clearly, $F_{k,0}$ are Dyck words, and $F_{k,k}$ are complemented Dyck words. The well-known Chung-Feller theorem asserts that $|F_{k,0}| = |F_{k,1}| = \cdots = |F_{k,k}| = \frac{1}{k+1}\binom{2k}{k} = C_k$, i.e., the number of words with e flaws is the same for each possible value of e. Mütze, Standke and Wiechert [MSW18] constructed a bijection between $F_{k,e}$ and $F_{k,e+1}$, for any $e = 0, \ldots, k-1$, that uses only adjacent transpositions, and that maps any Dyck word in $F_{k,0}$ to its complement in $F_{k,k}$ when applied k times. By combining this Gray code with any of the aforementioned transposition Gray codes for Dyck words, we obtain a transposition Gray code for $(2k, k)$-combinations in which the number of flaws is alternatingly increasing and decreasing; see Figure 5 (f).
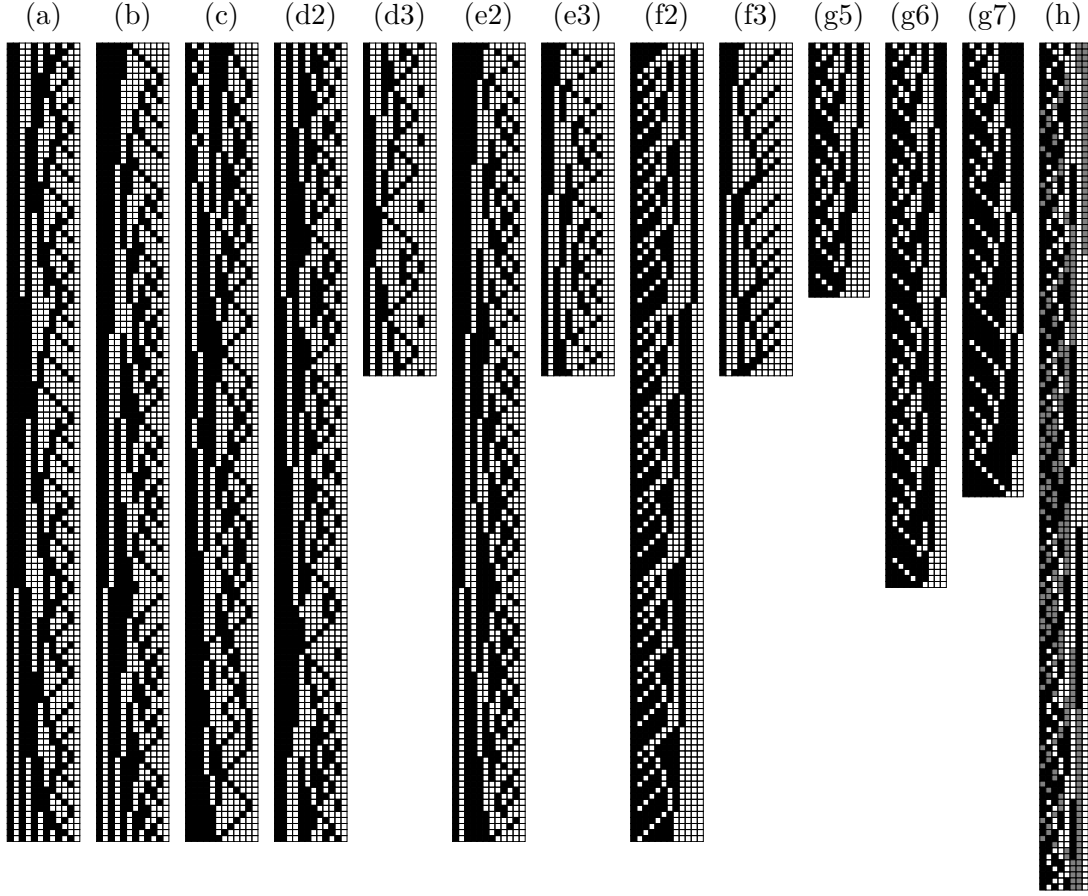
FIGURE 6. Gray codes for Dyck words and relatives (1-bits black, 0-bits white; $-$ in Motzkin words gray): (a) transpositions; (b) homogeneous transpositions; (c) adjacent transpositions; (d2)+(d3) 2/3-ary, homogeneous transpositions $00 \cdots 01 \leftrightarrow 10 \cdots 00$; (e2)+(e3) 2/3-ary, transpositions $01 \leftrightarrow 10$ or $001 \leftrightarrow 100$; (f2)+(f3) 2/3-ary, prefix shifts (cool-lex); (g5)–(g7) Dyck prefixes of length 10 with weight 5 (=Dyck words), 6 or 7; (h) Motzkin words with fixed content $\{1, 1, 1, 0, 0, 0, -, -\}$.

Alternatively, Dyck words can be generalized to *Dyck prefixes with fixed weight*, which are (n, k)-combinations for some $k \geq n/2$ such that every prefix contains at most as many 0s as 1s. A transposition Gray code for those objects, which is furthermore genlex, was derived by Vajnovszki and Wong [VW23], and it has a simple greedy description; see Figure 6 (g5)–(g7). The authors also interleave these listings into a Gray code for *ballot sequences*, which is the set of all Dyck prefixes of length n with weights $n/2 \leq k \leq n$, and this Gray code only uses transpositions and flips of single bits. In fact, the transpositions used in [VW23] are all of the form $11 \cdots 110 \leftrightarrow 01 \cdots 11$, i.e., they are homogeneous in the complemented bitstrings.

A *Motzkin word with fixed content* is a word over the alphabet $\{0, 1, -\}$ with k many 1s, k many 0s, and a fixed number m of occurrences of $-$, with the property that every prefix contains at most as many 0s as 1s. Consequently, Motzkin words for $m = 0$ are Dyck words. Vajnovszki [Vaj02] presents a genlex Gray code for Motzkin words with fixed content that either transposes two symbols or permutes three symbols in each step; see Figure 6 (h). These can be combined into a Gray code for Motzkin words of fixed length with an arbitrary number of $-$s.

Lapey and Williams [LW22b] devised a shift Gray code for Łukasiewicz words with fixed content, which generalize both Dyck words and Motzkin words with fixed content.

Further Gray codes for Catalan objects will be discussed in Sections 5.4, 5.5, 6.3, 8.1 and 8.2.

4.5. **Weight ranges.** As discussed before, combinations can be represented by bitstrings of length n with weight k. On the other hand, in Section 3 we discussed Gray codes for all bitstrings of length n, irrespective of their weight. Generalizing both types of objects, we may consider bitstrings of length n whose weight is in some interval $[k, \ell]$, where $0 \leq k \leq \ell \leq n$. The corresponding flip graph, denoted by $Q_{n,[k,\ell]}$, is the subgraph of the n-cube induced by all levels from k to ℓ, where the kth *level of* Q_n contains all bitstrings with weight k. The graph $Q_{n,[k,\ell]}$ is clearly bipartite, and it is easy to show that the partition classes have the same size if and only if n is even and $(k, \ell) = (0, n)$, or n is odd and $k + \ell = n$, i.e., the boundary levels k and ℓ are symmetric around the middle. Consequently, a Hamilton cycle in $Q_{n,[k,\ell]}$ can only exist in one of those cases.

One of most intriguing instances of this problem is the *middle levels conjecture*, which asks for a Hamilton cycle in $Q_{2k+1,[k,k+1]}$ for all $k \geq 1$, i.e., a Hamilton cycle in the middle two levels of the $(2k+1)$-cube. This question was first raised by Havel [Hav83] and Buck and Wiedemann [BW84] independently, and it received considerable attention in the literature. The conjecture was answered affirmatively by Mütze [Müt16], who showed that there are in fact double-exponentially (in k) many distinct Hamilton cycles. Subsequently, a shorter and more accessible proof of the middle levels conjecture was presented by Gregor, Mütze and Nummenpalo [GMN18], which reduces the problem to finding a spanning tree in an auxiliary graph on plane trees. This construction was turned into a loopless algorithm by Mütze and Nummenpalo [MN20]. An even shorter 2-page 'book proof' of the middle levels conjecture was presented by Mütze [Müt24a]. Consider a Hamilton cycle in the middle levels graph $Q_{2k+1,[k,k+1]}$, and prefix every bitstring of weight k with 1 and every bitstring of weight $k + 1$ with 0. Note that the resulting listing is a star transposition ordering of $(2k+2, k+1)$-combinations, i.e., in each step, the first bit is swapped with some later bit; see Figure 5 (e). Knuth [Knu11] raised a strengthened version of the middle levels conjecture, asking for a Hamilton cycle in $Q_{2k+1,[k,k+1]}$ that can be partitioned into $2k + 1$ blocks such that each block is a cyclic shift of the previous one (without the first column). Such an ordering has the property that any two of the columns $2, \ldots, 2k + 2$ of bits differ by a cyclic shift, i.e., it is a single-track Gray code, with the exception of the first column which is simply $101010\cdots$. Furthermore, such a Hamilton cycle corresponds to a Gray code for necklaces of length $2k + 1$ with weights k or $k + 1$. A general algorithmic solution for this problem was presented by Merino, Mička and Mütze [MMM22]; see Figure 5 (e'). Moreover, the middle levels graph $Q_{2k+1,[k,k+1]}$ was shown to be Hamilton-laceable by Gregor, Merino and Mütze [GMM23a]. While the algorithms described in [MN20] and [MMM22] solve the middle levels problem and its strengthened version efficiently, these algorithms are admittedly rather complicated. It remains open to find a simple algorithm for solving either of these problems, $\boxed{\text{P24}}$ ideally in the form of a successor rule. The authors of [MMM22] offer a reward of 2^5 EUR for an algorithm that can be implemented with at most 4000 ASCII characters of C++ code.

The more general *central levels problem* asks for a Hamilton cycle in $Q_{2k+1,[k-c,k+1+c]}$ for all $k \geq 1$ and $0 \leq c \leq k$. This problem was raised independently by Buck and Wiedemann [BW84], Savage [Sav93], Gregor and Škrekovski [GŠ10], and by Shen and Williams [SW19]. A solution for the case $c = k$ (all levels) is given by the BRGC. The case $c = k - 1$ (all levels except the two extreme vertices) was solved by Buck and Wiedemann [BW84] and in a more general setting by Locke and Stong [LS03]; see also [EHH01]. Gregor and Škrekovski [GŠ10] solved the
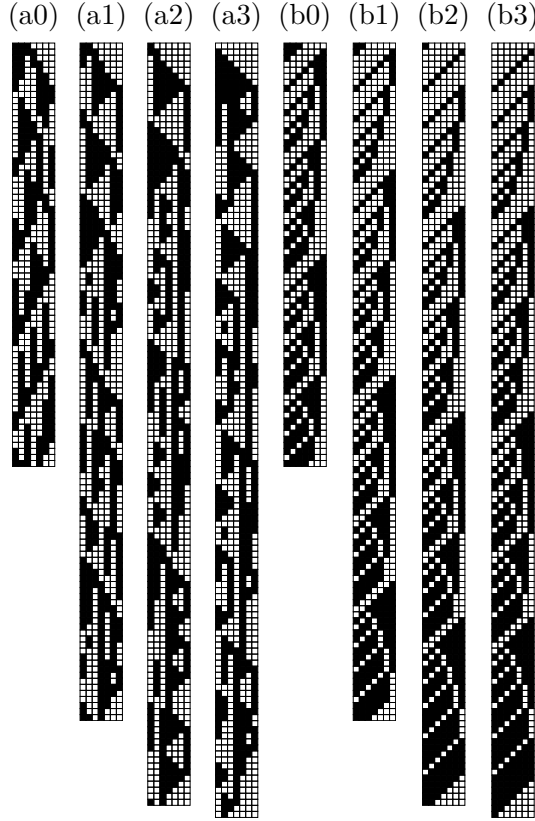
(a0) (a1) (a2) (a3) (b0) (b1) (b2) (b3)

FIGURE 7. Weight-range combination Gray codes (1-bits black, 0-bits white): (a0)–(a3)/(b0)–(b3) central levels graph $Q_{7,[3-c,4+c]}$ for $c = 0, 1, 2, 3$ by single bitflips or prefix shifts, respectively.

case $c = k - 2$ (all levels except the top two and bottom two). The case $c = 0$ (middle two levels) follows from the aforementioned solution of the middle levels conjecture. Later, Gregor, Jäger, Mütze, Sawada and Wille [GJM+22] presented a solution for the case $c = 1$ (middle four levels), and their construction is based on symmetric chain decompositions. Extending these ideas further, a full solution of the central levels problem, valid for all $k \geq 1$ and $0 \leq c \leq k$, was

P25 presented by Gregor, Mička and Mütze [GMM23b]; see Figure 7 (a0)–(a3). It is open to find an efficient algorithm for computing a Hamilton cycle in $Q_{2k+1,[k-c,k+1+c]}$ for general values of c. For $c = 0$ and $c = k$ such algorithms are known by the aforementioned results.

Coming back to the most general version of the problem about $Q_{n,[k,\ell]}$, Gregor and Mütze [GM18] showed that for all $n \geq 3$ and $0 \leq k \leq \ell \leq n$, in those cases where the partition classes of $Q_{n,[k,\ell]}$ have different sizes, the graph admits a saturating cycle and a tight enumeration, where a *saturating cycle* is a cycle in $Q_{n,[k,\ell]}$ that visits all vertices in the smaller partition class, and a *tight enumeration* is a cyclic listing of the bitstrings in both partition classes, flipping a single bit in most steps, and exchanging a 0 and 1 in d steps, where d is the difference in size between the partition classes (the steps flipping two bits are within the larger partition class). Saturating cycles and tight enumerations are natural generalizations of Hamilton cycles for bipartite graphs whose partition classes are not necessarily of the same size. Several of the weight range Gray codes devised in [GM18] are obtained by considering sublists of the BRGC within an interval $[k, \ell]$.

A *Boolean layer cake* [Sch99] is obtained from Q_n by selecting any subset of complete levels (not necessarily consecutive), connecting any two vertices in subsequent levels by an edge if they

are connected by a weight-monotone path in Q_n (equivalently, if the vertices are from subsequent levels k and ℓ, the path has length $|k - \ell|$). Obviously, the graph $Q_{n,[k,\ell]}$ is the Boolean layer cake obtained by selecting all levels $k, k+1, \dots, \ell$. It is open which Boolean layer cakes admit Hamilton cycles, or more generally, saturating cycles and tight enumerations.   P26

Stevens and Williams [SW14] modified the aforementioned cool-lex successor rule for combinations to be able to generate all bitstrings of length n as follows: Shift the shortest prefix of length at least 3 that ends with 01, or if no such prefix exists complement the first bit and shift the entire string; see Figure 7 (b3). Moreover, they generalized the rule even further so as to generate all bitstrings with weight in the interval $[k, \ell]$ by prefix shifts: Shift the shortest prefix of length at least 3 that ends with 01, or if no such prefix exists, complement the first bit if this keeps the weight in the interval $[k, \ell]$, and then shift the entire string; see Figure 7 (b0)–(b3). In the resulting cyclic Gray codes, any two consecutive bitstrings differ in at most 4 positions.

4.6. **Bipartite Kneser graphs and Kneser graphs.** For integers $k \geq 1$ and $n \geq 2k+1$, the *bipartite Kneser graph* $H(n, k)$ has as vertices all k-element and $(n-k)$-element subsets of $[n]$, and an edge between a set A of size k and a set B of size $n - k$ if $A \subseteq B$. Considering the characteristic vectors of those subsets, we see that $H(n, k)$ is isomorphic to the Boolean layer cake given by levels k and $n - k$ in the n-cube. In particular, $H(2k+1, k)$ is isomorphic to the middle levels graph $Q_{2k+1,[k,k+1]}$. Hurlbert [Hur94] and Roth (see [Gou91]) conjectured that $H(n, k)$ has a Hamilton cycle for all $k \geq 1$ and $n \geq 2k+1$. This conjecture was proved by Mütze and Su [MS17], using an inductive construction in Q_n that uses the middle levels theorem as a base case. Rephrased in terms of a Gray code, this result asserts that all k-element and $(n-k)$-element subsets of $[n]$ (i.e., (n, k)- and $(n, n-k)$-combinations) can be listed cyclically, so that we alternatingly move to a subset or a superset of the current set. The construction from [MS17] has the additional property that any two consecutive k-sets, which occur after every two steps, differ only in exchanging one element; see Figure 8 (a1)–(a4).

For integers $k \geq 1$ and $n \geq 2k+1$, the *Kneser graph* $K(n, k)$ has as vertices all k-element subsets of $[n]$, and an edge between any two disjoint sets. The Kneser graphs $K(2k+1, k)$ are also known as *odd graphs* O_k. A Hamilton cycle in the Kneser graph corresponds to a cyclic listing of all k-element subsets of $[n]$ (i.e., (n, k)-combinations), such that any two consecutive sets are disjoint. It has long been conjectured that all Kneser graphs $K(n, k)$ have a Hamilton   P27
cycle, with one notable exception, the Petersen graph $K(5, 2) = O_2$; see Figure 1 (c). For the   solved
odd graphs this conjecture was raised by Meredith and Lloyd [ML72, ML73] and Biggs [Big79]. Mütze, Nummenpalo and Walczak [MNW21] proved that the odd graphs $O_k = K(2k+1, k)$ have a Hamilton cycle for all $k \geq 3$ (even double-exponentially many distinct ones), using as a starting point the cycle factor in O_k from [MSW18] whose cycles all have the same length $k + 1$; see Figure 8 (b4). Combined with an inductive construction of Johnson [Joh11], their result implies that $K(2k + 2^a, k)$ has a Hamilton cycle for all $k \geq 3$ and $a \geq 0$, in particular for $n = 2k + 2$. A complete solution for all remaining cases $n \geq 2k + 3$ was given by Merino, Mütze and Namrata [MMN23]; see Figure 8 (b1)–(b3).

For integers $k \geq 1$, $0 \leq t \leq k - 1$, and $n \geq 2k + 1$ if $t = 0$ or $n \geq 2k - t$ if $t > 0$, the *uniform subset graph* $U(n, k, t)$ has as vertices all k-element subsets of $[n]$, and an edge between any two sets whose intersection has size t. By taking complements, we see that $U(n, k, t)$ is isomorphic to $U(n, n - k, n - 2k + t)$. For $t = 0$ this precisely gives the Kneser graphs. For $t = k - 1$ we get the flip graphs of (n, k)-combinations under transpositions, the Johnson graphs $J(n, k)$. For this reason uniform subset graphs are also known as generalized Johnson graphs in the literature. The   P28
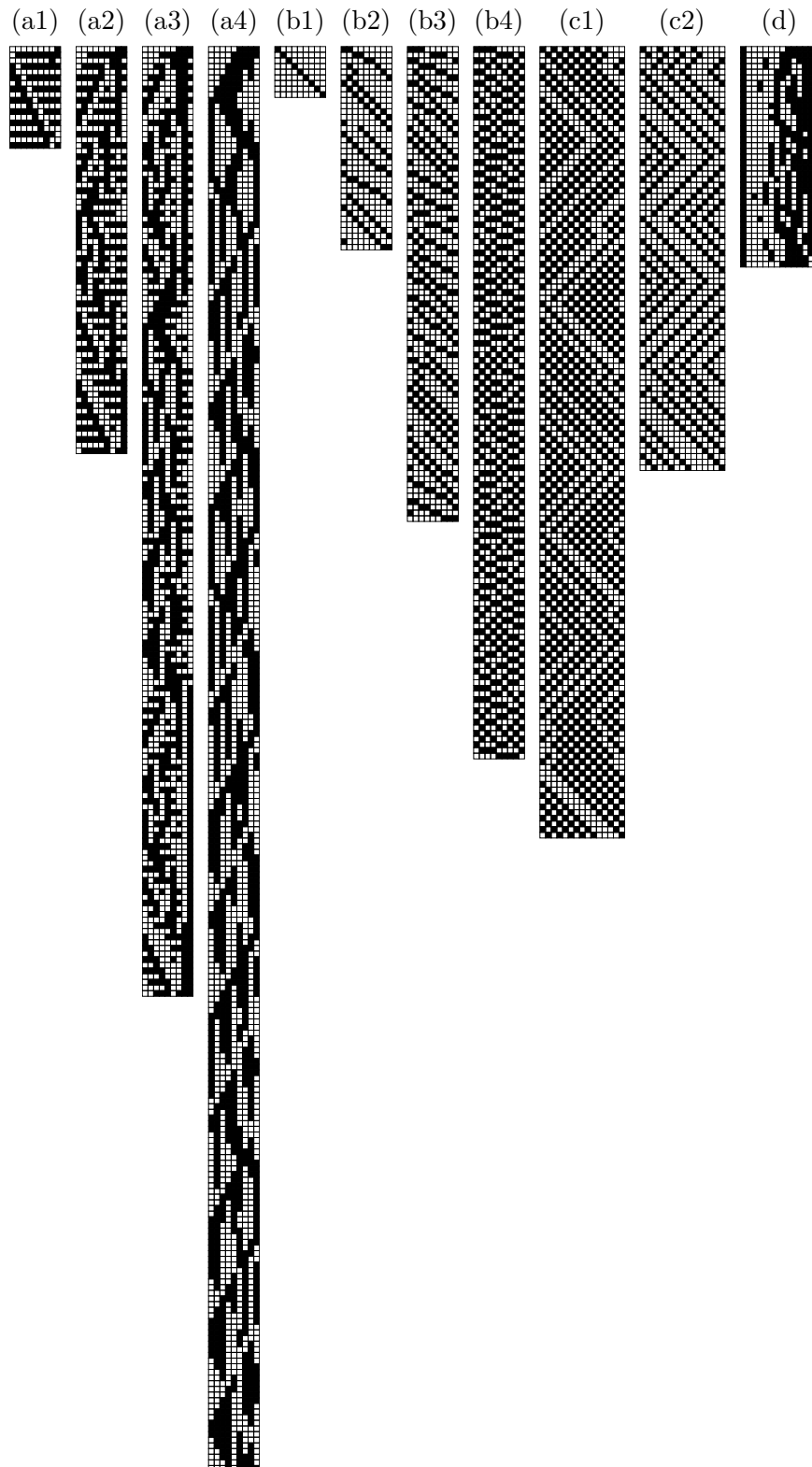graphs $U(n, k, t)$ were introduced by Chen and Lih [CL87], who conjectured that they all have a   solved

FIGURE 8. Hamilton cycles in (bipartite) Kneser graphs and relatives (1-bits black, 0-bits white): (a1)–(a4) bipartite Kneser graphs $H(9,k)$, $k = 1, 2, 3, 4$; (b1)–(b4) Kneser graphs $K(9,k)$, $k = 1, 2, 3, 4$; (c1) Schrijver graph $S(15,6) = S(15,6,2)$; (c2) s-stable Kneser graph $S(15,4,3)$; (d) numerical semigroups with genus 7.

Hamilton cycle, except the Petersen graph, which arises as $U(5, 2, 0)$ or $U(5, 3, 1)$ (cf. [Gou91]). A solution of this conjecture for the case $t = k - 1$ follows from the transposition Gray codes for (n, k)-combinations presented earlier. In addition to this, Chen and Lih settled the cases $t = k - 2$ and $t = k - 3$ in their paper. A complete solution of the problem was presented by Merino, Mütze and Namrata [MMN23]. These results settle Lovász' conjecture mentioned in Section 2.8 for all known families of vertex-transitive graphs defined by intersecting set systems (see [Müt24b]).

For integers $k \geq 1$ and $n \geq 2k + 1$, the *Schrijver graph* $S(n, k)$ is the subgraph of $K(n, k)$ induced by all subsets of $[n]$ that contain no two circularly adjacent elements. More generally, for integers $k \geq 1$, $s \geq 2$, and $n \geq sk + 1$, the *s-stable Kneser graph* $S(n, k, s)$ is the subgraph of $K(n, k)$ induced by all subsets of $[n]$ in which any two elements are in circular distance at least s. Clearly, we have $S(n, k, 2) = S(n, k)$. Schrijver graphs are named after Schrijver [Sch78], who showed that they are vertex-critical subgraphs of $K(n, k)$, i.e. they have the same chromatic number $n - 2k + 2$ as $K(n, k)$, but removing any of their vertices yields a graph with smaller chromatic number. In particular, $S(2k + 1, k)$ is an odd cycle with chromatic number 3. Schrijver graphs and more generally s-stable Kneser graphs were shown to be Hamiltonian by Ledezma and Pastine [LP24] and Mütze and Namrata [MN24], independently; see Figure 8 (c1)+(c2).

Biggs [Big79] conjectured that for $k \geq 3$ the edges of the odd graph O_k can be partitioned into $\lfloor (k+1)/2 \rfloor$ edge-disjoint Hamilton cycles. We do not even know two edge-disjoint Hamilton cycles in the odd graphs, the middle levels graphs $Q_{2k+1, [k, k+1]}$, or more generally, in Kneser graphs $K(n, k)$ or bipartite Kneser graphs $H(n, k)$. P61

4.7. Other generalizations and variants.
Combinations are generalized by bounded integer compositions, and Gray codes for those are discussed in Section 7.2.

Joichi, White and Williamson [JWW80] consider classes of combinatorial objects that satisfy the two-parameter recursion $C_{n,k} = a_{n,k} \cdot C_{n-1,k-1} + b_{n,k} \cdot C_{n-1,k}$, which includes (n, k)-combinations (for $a_{n,k} = b_{n,k} = 1$), set partitions of $[n]$ into k parts, permutations of $[n]$ with k cycles, and permutations of $[n]$ with k runs. These objects are in bijective correspondence with monotone paths in the integer grid from $(0, 0)$ to $(n - k, k)$, where the steps have edge-multiplicities as specified by the coefficients $a_{n,k}$ and $b_{n,k}$, i.e., they are described by an (n, k)-combination for the path, and an additional word of length n that describes which of the multi-edges are taken. The paper describes a general Gray code scheme for these pairs of strings, obtained from fusing a combination Gray code with a straightforward product space Gray code.

A *numerical semigroup* is a subset S of the non-negative integers $\mathbb{N}$ such that $0 \in S$, $\mathbb{N} \setminus S$ is finite, and $a, b \in S$ implies that $a + b \in S$. The numbers in $G := \mathbb{N} \setminus S$ are referred to as *gaps* of the semigroup, and the quantity $g := |G| = |\mathbb{N} \setminus S|$ is referred to as its *genus*. For any semigroup with genus g, we have $\max G < 2g$, so any such semigroup can be represented as a $(2g, g)$-combination by considering the first $2g$ entries of the characteristic vector of G. Is there a Gray code for numerical semigroups of genus g, represented as $(2g, g)$-combinations, for every $g \geq 1$? Figure 8 (d) shows a solution for $g = 7$. Jack Barron [Bar23] made some progress by showing that the corresponding flip graph is 2-connected, which implies a 2-Gray code by Fleischner's [Fle74] theorem discussed in Section 2.4. This problem is interesting, as for the counting sequence of numerical semigroups with genus g, even basic facts such as monotonicity in g are not proved; see [BA08] and [OEIS A007323]. P29

5. PERMUTATIONS

In this section we first discuss Gray codes for listing all $n!$ permutations of $[n] = \{1, 2, \ldots, n\}$ using different flip operations. We then discuss Gray code for interesting subsets of permutations,

obtained by imposing additional constraints. Lastly, we discuss Gray codes for generalizations and variants of permutations.

5.1. **Transpositions.** Possibly the most basic operation on a permutation is a *transposition*, i.e., the exchange of two entries of the permutation. Transposition Gray codes for permutations can be constructed inductively by appending an additional symbol in all possible ways to a Gray code with one symbol less, using transpositions with the newly appended symbol at the transitions between blocks, which yields a genlex ordering. Wells [Wel61] devised an explicit rule for producing such a transposition ordering of permutations; see Figure 9 (a). Another simple rule was found by Heap [Hea63], which is to perform the ith transposition involving the entry at position n in a block with the first entry if n is odd, and with the ith entry if n is even; see Figure 9 (b). Lipski [Lip79] considerably generalized the methods proposed by Wells and Heap, and came up with an entire family of transposition Gray codes for permutations, two of which are shown in Figures 9 (c)+(c'); see [Arn11].

The famous *Steinhaus-Johnson-Trotter (SJT)* algorithm generates permutations by *adjacent transpositions*, i.e., transpositions of neighboring entries of the permutation; see Figure 1 (d) and Figure 9 (d). This ordering can also be constructed inductively, but instead of appending an additional symbol in all possible ways, we insert the new symbol in all possible ways from right to left alternatingly. This algorithm was discovered independently by Trotter [Tro62] and Johnson [Joh63], and is also implicit in a book by Steinhaus [Ste64, p. 49–50], where he describes a puzzle about generating all permutations of beads on an abacus. The SJT algorithm is also known as the method of *plain changes*; see Knuth's book [Knu11, Sec. 7.2.1.2] for the historical origins of this algorithm in the art of change ringing of church bells. Ehrlich gives a loopless implementation of the SJT algorithm [Ehr73a, Ehr73b]. Sedgewick [Sed77] provides an insightful analysis and fair comparison of the computational efficiency of several permutation generation algorithms; see also Arndt's book [Arn11].

Compton and Williamson's [CW93] *doubly-adjacent* Gray code is an ordering of permutations by adjacent transpositions with the additional property that for any two consecutive transpositions $(i, i+1)$ and $(j, j+1)$, we have $|i-j| = 1$; see Figure 9 (e). The inductive construction underlying their proof is considerably more involved than for the Steinhaus-Johnson-Trotter ordering.

Another special type of transpositions are *star transpositions*, i.e., an exchange of the first entry of the permutation with some later entry. Ehrlich proposed a general method for generating permutations by star transpositions, which he communicated to Martin Gardner in 1987 [Knu11, Sec. 7.2.1.2]; see Figure 9 (f) and also [XM09, Sec. 10]. Curiously, the Ehrlich ordering is cyclic for $n = 2, 3, 4, 6, 12, 20, 40$, but for no other $n \leq 250\,000$ [Knu11, Ex. 58 in Sec. 7.2.1.2].

Shen and Williams [SW13b] constructed a *'hot potato'* Gray code for permutations, where the value 1 moves in each step, and it moves by circularly adjacent or semi-adjacent transpositions, i.e., transpositions of the form $(i, i+1)$ or $(i, i+2)$ modulo n; see Figure 9 (g). We can think of the value 1 as being the hot potato that is passed around in a circle of n children, and it is always passed to a neighbor in distance 1 or 2 along the circle. By considering the inverse permutations of this ordering, the meaning of values and positions is exchanged, so we obtain a star transposition ordering of permutations (the entry at position 1 changes in each step), with the additional property that the value at position 1 always changes by ± 1 or ± 2 modulo n; see Figure 9 (g^{-1}) and cf. Figure 9 (f). Shen and Williams' construction uses the aforementioned doubly-adjacent Gray code of Compton and Williamson as a building block.

The *parity* of a permutation π is the parity of the number of its inversions, where an *inversion* is a pair $(\pi(i), \pi(j))$ with $i < j$ and $\pi(i) > \pi(j)$. Equivalently, the parity of π is given by the
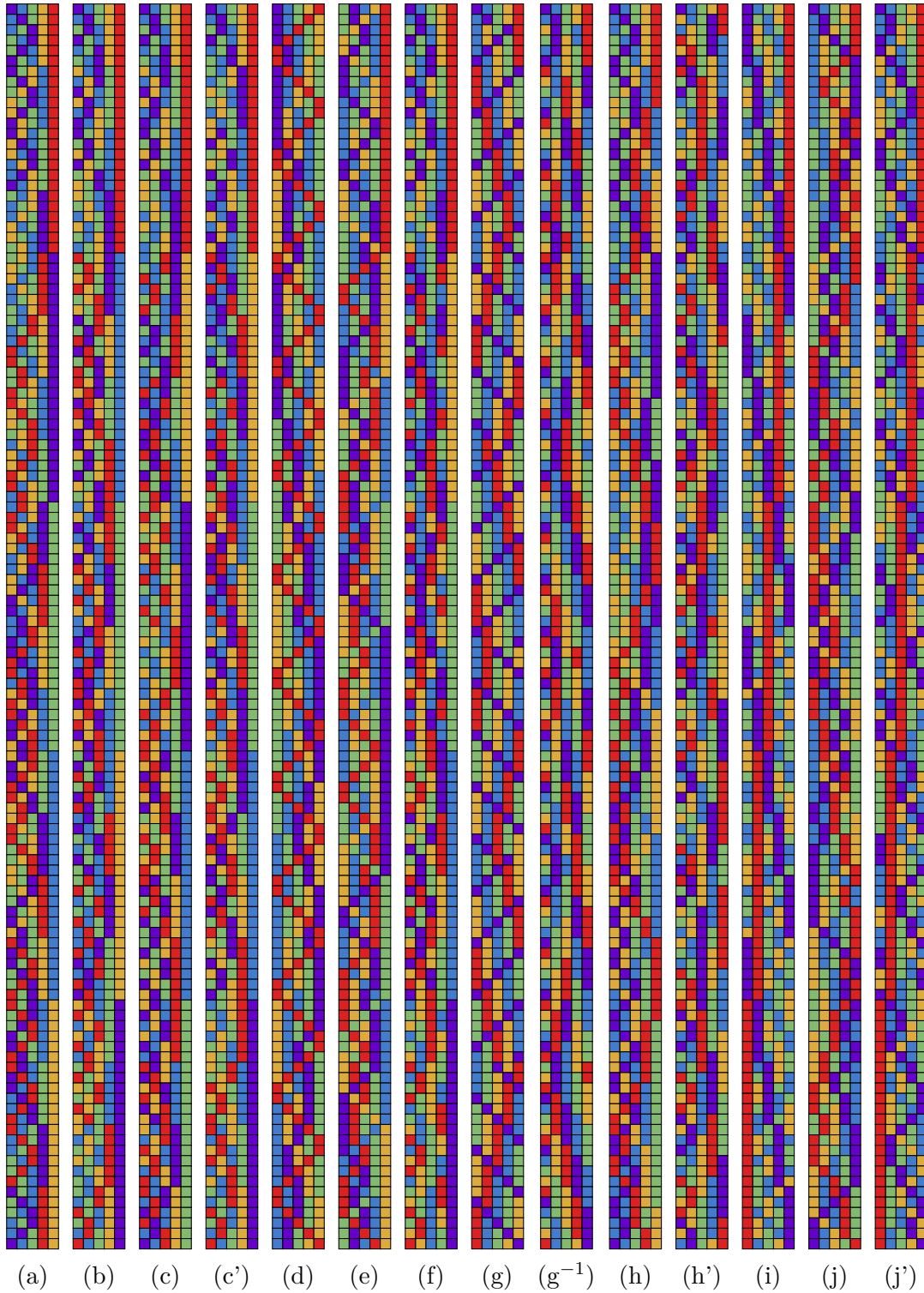
FIGURE 9. Gray codes for permutations of length 5 (1=purple, 2=blue, 3=green, 4=yellow, 5=red): (a) transpositions (Wells); (b) transpositions (Heap); (c) transpositions (Lipski 10); (c') transpositions (Lipski 16); (d) adjacent transpositions (SJT); (e) doubly-adjacent; (f) star transpositions; (g) hot potato; (g^{-1}) inverse of hot potato; (h) $(1, 2)$-alternating adjacent transpositions; (h') $(1, 2)$-alternating star transpositions; (i) balanced transpositions; (j) balanced adjacent transpositions; (j') balanced circularly adjacent transpositions; (figure continues on next page)
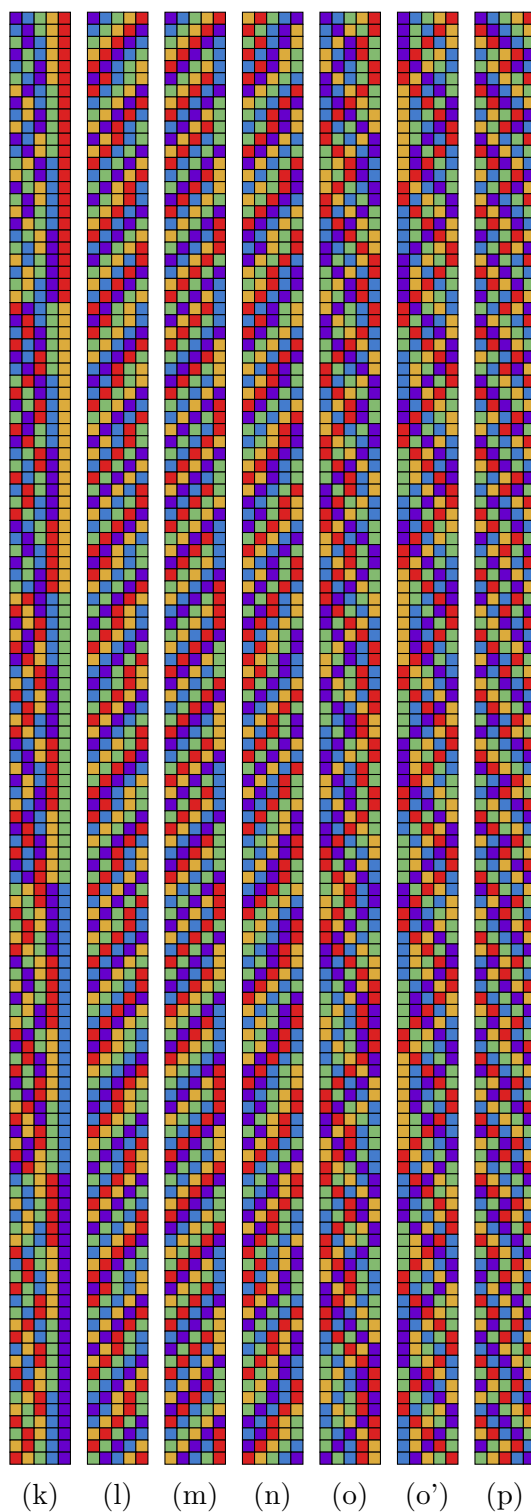
FIGURE 9. (continued) (k) prefix reversals; (l) prefix shifts; (m) $\{\sigma_5, \sigma_4\}$; (n) $\{\sigma_5, \sigma_2\}$; (o) $\{(1\,2)(3\,4), (2\,3)(4\,5), (1\,2)\}$; (o') $\{(1\,2)(3\,4), (2\,3)(4\,5), (3\,4)\}$; (p) derangement order.

parity of the number of transpositions whose product is π. Alternatively, the parity of π is the parity of n minus the number of cycles of π. Note that every transposition changes the parity of a permutation.

For a particular set of admissible transpositions, we consider the *transposition graph* on the vertex $[n]$, which contains as edges all admissible transpositions (i, j). For adjacent transpositions, the transposition graph is a path, whereas for star transpositions, the transposition graph is a star. Clearly, to be able to generate all permutations by a set of transpositions, the transposition graph must be connected. Kompel'maher and Liskovec [KL75] and Slater [Sla78] independently generalized the aforementioned results considerably by proving that all permutations of $[n]$ can be generated for any transposition tree on $[n]$. Even more generally, Tchuente [Tch82] showed that for any transposition tree on $[n]$, $n \geq 4$, and for any two permutations of opposite parity, there is a transposition Gray code starting and ending with the two chosen permutations. As every transposition changes the parity, any transposition Gray code lists permutations with even and odd parity alternatingly. In graph-theoretic language, the underlying flip graph for any tree of transpositions is bipartite, and Tchuente's result asserts that it is Hamilton-laceable; see also [XM09, Sec. 12].

Ruskey and Savage [RS93] showed that for any transposition tree on $[n]$, $n \geq 5$, and any fixed transposition from that tree, there is a transposition Gray code that uses this fixed transposition in every second step. In other words, the fixed transposition induces a perfect matching in the corresponding flip graph, and this perfect matching extends to a Hamilton cycle, i.e., the cycle alternatingly uses matching and non-matching edges. Their Gray code for a path and star on $n = 5$ vertices and for the fixed transposition $(1, 2)$ is shown in Figures 9 (h) and (h'), respectively.

Felsner, Kleist, Mütze and Sering [FKMS20] raised the question whether there is a transposition [P30] Gray code for permutations that uses each of the $\binom{n}{2}$ transpositions equally often. This is the [solved] analogue of balancing the transition counts in Gray codes for binary strings; recall Section 3.2. This problem was solved by Lucca Tiemens [Tie24b] with a beautiful inductive construction; see Figure 9 (i). Similarly, can all $n!$ permutations be listed by adjacent transpositions, using each of [P62] the $n-1$ adjacent transpositions equally often? While there is no solution for $n = 4$, there is one for $n = 5$; see Figure 9 (j). As a partial result, Gregor, Merino and Mütze [GMM24] constructed a permutation Gray code for odd n that uses each of the n circularly adjacent transpositions equally often; see Figure 9 (j'). Tiemens [Tie24a] complemented this result by providing such a Gray code also for the even values of n.

Furthermore, can all $n!$ permutations be listed by star transpositions, using each of the $n-1$ [P63] star transpositions equally often? While there is no solution for $n = 4$, Gregor, Merino and Mütze [GMM24] found one for $n = 6$.

Lastly, is there a single-track Gray code for permutations by adjacent transpositions, i.e., can [P64] all $n!$ permutations be generated by adjacent transpositions such that any two columns in the row-by-row listing of permutations only differ in a cyclic shift? For arbitrary transpositions and circularly adjacent transpositions the aforementioned balanced Gray codes also have the single-track property.

5.2. **Reversals, shifts and other generators.** Ord-Smith [Ord67] devised a method to generate all permutations by *prefix reversals*; see Figure 9 (k). His listing also has the genlex property, and the average length of reversed prefixes approaches $e = 2.71828\ldots$ (a constant, which allows for an efficient algorithm). This ordering was rediscovered by Zaks [Zak84], who was motivated by sorting problems involving stacks of pancakes; see also [XM09, Sec. 11]. A prefix reversal corresponds to flipping over some of the pancakes at the top of the stack. In the

following, we write ρ_k for the operation that reverses the first k entries of a permutation, i.e., in cycle notation we have $\rho_k := (1\ k)(2\ k-1)\cdots$.

Prefix shifts have also been investigated heavily for permutation generation. Unlike transpositions and prefix reversals discussed before, prefix shifts are in general not involutions. Formally, for any $2 \leq k \leq n$, we write σ_k for the operation that cyclically left-shifts the first k entries of a permutation by one position, i.e., $\sigma_k = (k\ k-1\cdots 2\ 1)$. Note that $\sigma_2 = \rho_2$ simply transposes the first two entries. Corbett [Cor92] showed that all permutations can be generated cyclically by prefix shifts of arbitrarily length, i.e., using the generators $\{\sigma_n, \sigma_{n-1}, \ldots, \sigma_2\}$; see Figure 9 (l). Compton and Williamson [CW93], as an application of their doubly-adjacent Gray code, showed that we can also use the generators $\{\sigma_n, \sigma_n^{-1}, \sigma_2\}$. Stevens and Williams [SW12b] showed the same result for the generators $\{\sigma_n, \sigma_3, \sigma_2\}$, and Ruskey and Williams [RW10] proved it for $\{\sigma_n, \sigma_{n-1}\}$; see Figure 9 (m). In a recent breakthrough, Sawada and Williams [SW20] give a simple and explicit rule to generate all permutations of length n using only the two generators $\{\sigma_n, \sigma_2\}$, and this listing is cyclic if and only if n is odd; see Figure 1 (e) and Figure 9 (n). Subsequently, a simple loopless algorithm for this problem was devised by Lipták, Masillo, Navarro and Williams [LMNW23]. Instead of cyclically shifting the array that holds the permutation when applying σ_n, they increment an index to what is considered the 'first' entry. This problem was known as *sigma-tau problem*, and was first mentioned in Nijenhuis and Wilf's book [NW75, Ex. 6]. All of the preceding results mentioned in this paragraph assert the existence of Hamilton paths or cycles in the Cayley digraph $\vec{\Gamma}(S_n, X)$ for various sets X of shifts as generators.

We may also consider other generators of the symmetric group S_n. In this direction, Rapaport-Strasser [RS59] proved that the Cayley graph $\Gamma(S_n, X)$ has a Hamilton cycle for the three involutions $X = \{(1\,2)(3\,4)(5\,6)\cdots, (2\,3)(4\,5)(6\,7)\cdots, (1\,2)\}$. Knuth [Knu11, Ex. 69 in Sec. 7.2.1.2] devised an efficient algorithm for computing such a cycle for the three involutions $X = \{(1\,2)(3\,4)(5\,6)\cdots, (2\,3)(4\,5)(6\,7)\cdots, (3\,4)(5\,6)(7\,8)\cdots\}$, which can be adapted to also work for Rapaport-Strasser's set of generators. The corresponding Gray codes for $n = 5$ are shown in Figures 9 (o) and (o'), respectively. In Knuth's Gray code, the largest value n moves back and forth between the first and last position, in exactly the same way as in the SJT listing (but the other values move differently). The fact that $\Gamma(S_n, X)$ has a Hamilton cycle for the two aforementioned sets X of generators follows from the general fact that the Cayley graph of any group generated by three involutions ρ, σ, τ has a Hamilton cycle if $\rho\tau = \tau\rho$; see [PR09, Lemma 1]. Note that by reordering positions, Rapaport-Strasser's and Knuth's generators are equivalent to $X = \{\rho_n, \rho_{n-1}, (1\,n)\}$ and $X = \{\rho_n, \rho_{n-1}, \rho'\}$, respectively, where ρ' reverses the middle $n - 2$ entries of a permutation.

Savage [Sav90] showed that for any $n \geq k \geq 2$ and $k \neq 3$, all permutations of length n can be listed cyclically so that consecutive permutations differ in exactly k positions. For $k = n$ such a listing is sometimes called a *derangement order*. Figure 9 (p) shows an example for this, using a construction due to Yarbrough described in [Sav97]. Rasmussen and Savage [RS94a] later generalized this result considerably, by showing that the corresponding flip graph is Hamilton-connected, even if we impose the additional constraint that the k modified positions are consecutive.

P31   Despite all of the aforementioned results, in general it is wide open for which sets of generators X the Cayley graph $\Gamma(S_n, X)$ or the Cayley digraph $\vec{\Gamma}(S_n, X)$ have a Hamilton cycle or path. Generalizing the sigma-tau problem, we may consider $X = \{\sigma_n, (1\ k)\}$ or $X = \{\sigma_n, \sigma_k\}$ for suitable values of n and $2 \leq k \leq n$. Specifically, the case $X = \{\sigma_n, \sigma_3\}$ for even n was suggested by Stevens and Williams [SW12b]. Furthermore, Savage [Sav97]

mentions $X = \{\rho_n, \rho_{n-1}, (n-1\ n)\}$ as an interesting open problem. Lastly, Bass and Sudborough [BS03] and Sawada and Williams [SW16b] suggest the instances $X = \{\rho_n, \rho_{n-1}, \rho_{n-2}\}$ and $X = \{\rho_n, \rho_{n-1}, \rho_2\}$.

5.3. **Restricted permutations.** *Even permutations* are permutations with even parity, i.e., with an even number of inversions. As mentioned before, any transposition Gray code visits permutations with even and odd parity alternatingly, so from any transposition Gray code for all permutations we can obtain a 2-transposition Gray code for even permutations (or odd permutations, or permutations with an even number of cycles, or an odd number of cycles); cf. [VV11]. Savage [Sav90] showed that they can be listed by applying a *3-rotation*, i.e., a product of two transpositions that share one element, in each step. Recall that each transposition changes the parity, so single transpositions cannot be used. Her listing is shown in Figure 10 (a), and it is genlex. She also came up with a listing that uses only 3-rotations of consecutive entries, to the left or right by one position if the middle entry is at an odd or even position, respectively (these are only $n-2$ different operations in total). This listing is described in [Knu11, Ex. 103 in Sec. 7.2.1.2], and it is shown in Figure 10 (a'). Gould and Roth [GR87] proved that for $n \geq 5$ all even permutations can be listed cyclically by 3-rotations of the form $(1\ i\ n)$ for some $1 < i < n$, i.e., in every step, the first, ith and last entry of the permutation rotate cyclically; see Figure 10 (a''). Equivalently, their result asserts that for $n \geq 5$ the Cayley digraph $\vec{\Gamma}(A_n, X)$ of the alternating group A_n with generators $X = \{(1\ i\ n) \mid 1 < i < n\}$ has a Hamilton cycle. By reordering positions, Gould and Roth's generators are equivalent to $X = \{(1\ 2\ i) \mid 3 \leq i \leq n\}$, which can be seen as 'star-like' 3-rotations, analogous to Ehrlich's star transpositions for S_n. Holroyd [Hol17] proved that for all odd $n \neq 5$, $\vec{\Gamma}(A_n, X)$ with generators $X = \{\sigma_3, \sigma_5, \ldots, \sigma_n\}$, i.e., odd prefix shifts, has a Hamilton cycle.

Walsh [Wal00] constructed a Gray code for permutations of length n with a fixed number k of inversions, $0 \leq k \leq \binom{n}{2}$, which uses a product of two adjacent transpositions in each step; see Figure 10 (b). He derived this from a genlex Gray code for integer compositions $b_1 + \cdots + b_n = k$ with bounded parts $b_i \leq i - 1$ for all $i \in [n]$, by interpreting the composition $(b_1, \ldots, b_n)$ as the inversion vector of a permutation. The *inversion vector* of a permutation π of length n is a vector $(b_1, \ldots, b_n)$, where b_i counts the number of values in π that are to the right of the value i and smaller than i. Consequently, any permutation with exactly k inversions has an inversion vector $(b_1, \ldots, b_n)$ with $b_1 + \cdots + b_n = k$. In Walsh's composition Gray code, consecutive compositions differ by changing one entry b_i by $+1$ and another entry b_j by -1, and this translates into a product of two adjacent transpositions in the permutation: Specifically, i is transposed with the first entry to its left smaller than i, and j is transposed with the first entry to its right larger than j. Vajnovszki and Vernay [VV11] generalized this Gray code to permutations for which the number of inversions is in any fixed interval.

An *alternating permutation* is a permutation π such that $\pi(1) < \pi(2) > \pi(3) < \pi(4) > \cdots$. Ruskey [Rus89] showed that for odd n, all alternating permutations can generated by transpositions (which must clearly be non-adjacent); see Figure 10 (c). He raised the question whether such a Gray code exists also for even values of n. A more general notion are *permutations with a signature*, which is sequence $s = (s_1, \ldots, s_{n-1})$ with $s_i \in \{+, -\}$, and the requirement is that $\pi(i) < \pi(i+1)$ if $s_i = +$ and $\pi(i) > \pi(i+1)$ if $s_i = -$ for all $1 \leq i < n$. Note that alternating permutations arise from the signature $s = (+, -, +, -, \ldots)$. Efficient generation algorithms for permutations with any fixed signature have been devised by Roelants van Baronaigien and Ruskey [RvBR92], and later by Korsh [Kor01]. However, none of these methods is based on a P33
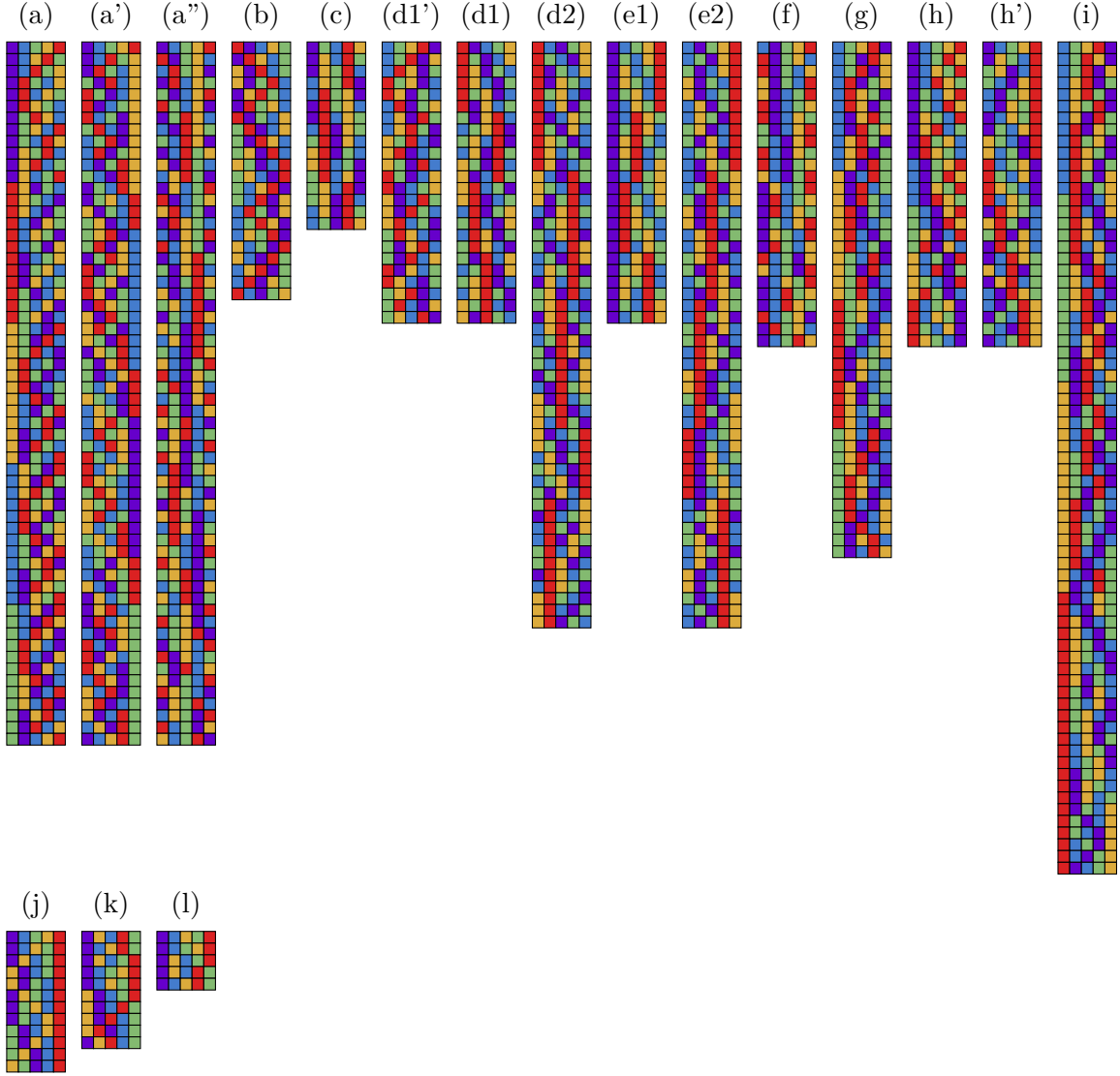
FIGURE 10. Gray codes for restricted permutations of length 5: (a)–(a") even permutations; (b) five inversions; (c) alternating; (d1')+(d1) one cycle (=cyclic permutations); (d2) two cycles; (e1)+(e2) one/two left-to-right minima; (f) one exceedance; (g) derangements; (h)+(h') involutions; (i) indecomposable; (j) rosary permutations; (k) linear extensions of ; (l) linear extensions of .

Gray code, so the question for the existence of a (transposition?) Gray code for permutations with a given signature remains open in general.

A *cycle* in a permutation π is sequence $i_1, \ldots, i_k$ such that $\pi(i_j) = i_{j+1}$ for $j = 1, \ldots, k-1$ and $\pi(i_k) = i_1$. Baril [Bar07] devised a Gray code to generate permutations of length n and any fixed number k of cycles, $1 \leq k \leq n$, with 3-rotations as transitions. Figure 10 (d1)+(d2) shows these orderings for $n = 5$ and $k = 1, 2$. Another listing for *cyclic permutations* ($k = 1$) based on the SJT algorithm was proposed by Knuth [Knu11, Ex. 96 in Sec. 7.2.1.2]; see Figure 10 (d1').

A *left-to-right minimum* in a permutation π is an entry $\pi(i)$, $i \in [n]$, for which $\pi(j) > \pi(i)$ for all $1 \leq j < i$. In particular, the first entry is always a left-to-right minimum. Baril [Bar13]

developed a Gray code for permutations of length n with a fixed number k of left-to-right minima, $1 \leq k \leq n$, which uses transpositions or 3-rotations as transitions; see Figure 10 (e1)+(e2). It is well known that for any n and k, permutations of $[n]$ with k left-to-right minima and permutations of $[n]$ with k cycles are equinumerous, and both are counted by the Stirling numbers of the first kind [OEIS A132393].

An *exceedance* in a permutation is an entry $\pi(i)$, $i \in [n]$, with $\pi(i) > i$. Baril [Bar08] describes a Gray code for permutations π that have exactly one exceedance, where consecutive permutations differ by a transposition or a 3-rotation; see Figure 10 (f). The question whether there is a Gray code for permutations with a fixed number k of exceedances is open for $k \geq 2$. $\boxed{\text{P34}}$

A *derangement* is a permutation $\pi : [n] \to [n]$ that has no fixpoints, i.e., $\pi(i) \neq i$ for all $i = 1, \ldots, n$ [OEIS A000166]. Korsh and LaFollette [KL04a] present a Gray code for derangements, using either one transposition or a 3-rotation in each step. Their listing is genlex; see Figure 10 (g). They also generalized their Gray code to visit all permutations with a given lower and upper bound on the number of fixpoints; see also [BV04].

An *involution* is a permutation $\pi : [n] \to [n]$ in which all cycles have length 1 or 2, i.e., for all $i = 1, \ldots, n$ we have $\pi(i) = i$ or $\pi(i) = j$ and $\pi(j) = i$ for some $j \in [n] \setminus \{i\}$ [OEIS A000085]. Walsh [Wal01] presents a Gray code for involutions, which uses either one or two transpositions or a 3-rotation in each step, and this listing is genlex; see Figure 10 (h). He also extended this result to involutions with a given number of fixpoints, in particular for fixpoint-free involutions (in the latter case no 3-rotations are needed). Gutierres, Mamede and Santos [GMS19] developed a Gray code for involutions that uses only one transposition or a 3-rotation in each step, i.e., consecutive permutations differ in at most three entries; see Figure 10 (h').

A permutation π of $[n]$ is *indecomposable* if no proper prefix of π is a permutation, i.e., $\pi([i]) \neq [i]$ for all $1 \leq i < n$. King [Kin06] constructed a transposition Gray code for indecomposable permutations; see Figure 10 (i). He also raised the question whether they admit an adjacent transposition Gray code. $\boxed{\text{P35}}$

A *rosary permutation* is bracelet of permutations, i.e., an equivalence class of permutations under cyclic shifts and reversal. We can pick as representatives the permutations that have n at their last position and the value 1 before the value 2, and then these representatives can be listed in adjacent transposition order by considering the first $(n-1)!/2$ permutations in the SJT listing for n; see Figure 10 (j) and [Roy73].

5.4. **Pattern-avoiding permutations.** Given a permutation π of length n and a permutation τ of length k, where $k \leq n$, we say that π *contains the pattern* τ, if there is a subsequence of indices $i_1 < \cdots < i_k$ such that the elements of $\pi(i_1), \ldots, \pi(i_k)$ appear in the same relative order as in τ. Otherwise, we say that π *avoids* τ. In a *vincular* pattern τ, two consecutive entries are underlined, and for π to contain τ, these entries must match consecutive entries of π. For example, $\pi = 531462$ contains $\tau = 4231$, as evidenced by the shaded entries of π, whereas π avoids $\tau' = 4\underline{23}1$ and also $\tau'' = 2413$. The study of pattern-avoidance in permutations is a major branch of combinatorics, and the reason is that a vast number of combinatorial objects are bijectively equivalent to pattern-avoiding permutations. For example, 321-avoiding permutations and 231-avoiding permutations are both counted by the Catalan numbers, and are thus bijectively equivalent to many other Catalan objects [Sta15]. For any pattern τ, we write $S_n(\tau)$ for the set of permutations of $[n]$ that avoid τ. Moreover, for any set X of permutation patterns (possibly of different lengths), we write $S_n(X)$ for the permutations of $[n]$ that avoid each of the patterns from X, i.e., $S_n(X) := \bigcap_{\tau \in X} S_n(\tau)$.

FIGURE 11. Gray codes for pattern-avoiding permutations of length 5 ((a)–(f) are for regular patterns X, and (g)–(m) are for tame patterns X): (a) $X = \{321\}$; (b) $X = \{321, 312\}$; (c) $X = \{321, 3412\}$; (d) $X = \{4321, 4312\}$; (e) $X = \{321, 3412, 4123\}$; (f) $X = \{4321, 4312, 4132, 4231\}$; (g) $X = \{231\}$; (h) $X = \underline{231}$; (i) $X = \{231, 132\}$ (permutations without peaks); (j) $X = \{2143\}$ (vexillary permutations); (k) $X = \{2143, 3412\}$ (skew-merged permutations); (l) $X = \{2413, 3142\}$ (separable permutations); (m) $X = \{2\underline{41}3, 3\underline{14}2\}$ (Baxter permutations); (n) stamp foldings; (n') semi-meanders.

Dukes, Flanagan, Mansour and Vajnovszki [DFMV08] developed Gray codes for pattern-avoiding permutations. To state their result, we call a set X of permutation patterns *regular*, if every $\tau \in X$ of length k satisfies the following two conditions: (i) two entries smaller than k

are to the right of k in τ; (ii) if the entry k is not at the leftmost position of τ, then X contains at least one subpattern of τ', where τ' is obtained from τ by transposing k with the entry to its left. For example, if $\tau = 436512 \in X$, then by (ii) one subpattern of $\tau' := 463512$ must be in X, for example $\sigma := 4632 = 3421 \in X$. Consequently, if $\sigma = 3421 \in X$, then by (ii) one subpattern of $\sigma' := 4321$ must be in X, for example $\sigma' \in X$ itself. Overall, $X = \{\tau, \sigma, \sigma'\} = \{436512, 3421, 4321\}$ is a regular set of patterns. Note that if all patterns in X have the largest entry at the leftmost position, then condition (ii) is void, and (i) requires that all patterns have length at least 3. The paper [DFMV08] shows how to list, for any regular set of patterns X, all pattern-avoiding permutations $S_n(X)$ such that at most five entries change in each step. This construction was refined by Baril [Bar09] to most three changes in each step (which must be a 3-rotation). Examples of regular patterns to which this result applies are $X = \{321\}$ or $X = \{312\}$ (counted by the Catalan numbers), $X = \{321, 312\}$ (counted by 2^{n-1}), $X = \{321, 3412\}$ or $X = \{321, 4123\}$ or $X = \{321, 4321\}$ (counted by the even-index Fibonacci numbers), $X = \{4321, 4312\}$ or $X = \{4213, 4123\}$ or $X = \{4132, 4231\}$ (counted by the Schröder numbers [OEIS A006318]), $X = \{321, 3412, 4123\}$ or $X = \{312, 3421, 4321\}$ (counted by the Pell numbers [OEIS A000129]), and $X = \{4321, 4312, 4132, 4231\}$ or $X = \{4213, 4123, 4132, 4231\}$ (counted by the central binomial coefficients $\binom{2n-2}{n-1}$); see Figure 11 (a)–(f).

Hartung, Hoang, Mütze and Williams [HHMW22] propose a different Gray code for generating pattern-avoiding permutations, using cyclic shifts of substrings. Specifically, a *right jump of the value* a_i in a permutation $\pi = a_1, \ldots, a_n$ cyclically shifts a substring $a_i, \ldots, a_j$ with $a_i > a_{i+1}, \ldots, a_j$ one position to the left, yielding $a_{i+1}, \ldots, a_j, a_i$, whereas a *left jump of the value* a_j shifts a substring $a_i, \ldots, a_j$ with $a_j > a_i, \ldots, a_{j-1}$ one position to the right, yielding $a_j, a_i, \ldots, a_{j-1}$, and the *distance* of the jump is the number $j - i$. A permutation pattern τ of length k is called *tame*, if it does not have the largest value k at the leftmost or rightmost position. A vincular pattern is tame if in addition to this condition the largest value k is part of the vincular pair. For example, $\tau = 231$ and $\tau = 2\underline{41}3$ are tame, whereas $\tau = 321$ and $\tau = 24\underline{13}$ are not. The paper shows how to list, for any set of tame patterns X, all pattern-avoiding permutations $S_n(X)$ by jumps. Examples of pattern-avoiding permutations to which this result applies are permutations without peaks ($X = \{231, 132\}$), vexillary permutations ($X = \{2143\}$), skew-merged permutations ($X = \{2143, 3412\}$), separable permutations ($X = \{2413, 3142\}$), and Baxter permutations ($X = \{2\underline{41}3, 3\underline{14}2\}$); see Figure 11 (g)–(m). The listings are obtained from a simple greedy algorithm, which repeatedly performs a minimal distance jump with the largest possible value, so that a new permutation in the list is generated. The main feature of this approach is that applying standard bijections to these jump orderings of permutations yields Gray codes for many different combinatorial objects; see Figure 12. This generation approach therefore generalizes several known Gray codes, such as the BRGC (for $X = \{231, 132\}$), the SJT algorithm (for $X = \emptyset$ minimal jumps are adjacent transpositions), the Gray code for binary trees by rotations due to Lucas, Roelants van Baronaigien and Ruskey [LRvBR93] (for $X = \{231\}$), and the Gray code for set partitions by element exchanges described by Kaye [Kay76] (for $X = \{\underline{231}\}$). It also yields Gray codes for many different classes of rectangulations [MM23a], and it produces Hamilton paths and cycles on large classes of polytopes [HM21, CMM22, CHM$^+$23]. The reason why minimal jumps are natural becomes clear when considering the inversion vector of the permutations. Observe that a right jump of a_i as indicated in the definition above decreases the a_ith entry of the inversion vector of π by $j - i$, whereas a left jump of a_j as indicated before increases the a_jth entry by $j - i$ (all values that are jumped across are smaller than the value that jumps). Consequently, minimal jumps change the inversion vector in a single entry by a
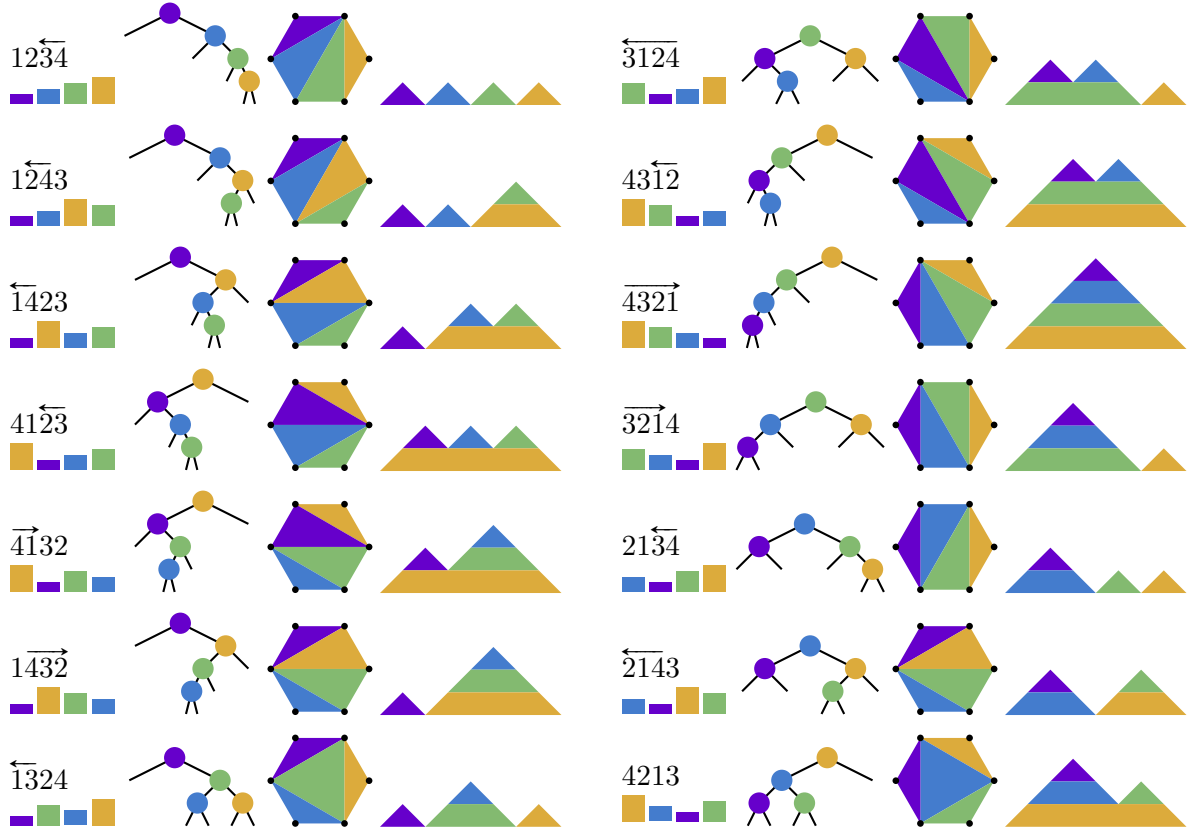
FIGURE 12. 231-avoiding permutations of length $n = 4$ generated by minimal jumps and the resulting Gray codes for Catalan objects (binary trees, triangulations, Dyck paths), which coincide with the ordering due to Lucas, Roelants van Baronaigien, and Ruskey [LRvBR93]. Minimal jumps in the permutations correspond to rotations in the binary trees, to flipping a diagonal in the triangulations, and to flipping a hill in the Dyck paths.

minimal amount, and a jump Gray code of permutations corresponds to a genlex ordering of the inversion vectors.

5.5. **Linear extensions and ideals of posets.** A *poset P* is a reflexive, antisymmetric and transitive binary relation. A *linear extension* of P is a total order (=chain) that extends P. For further terminology regarding posets used below, see for example [Tro92]. Linear extensions are sometimes also referred to as *topological sorts*, and they generalize many interesting combinatorial objects.

For example, if P is an antichain on $[n]$, then the linear extensions are all $n!$ permutations of $[n]$. More generally, if P is a set of k disjoint chains of sizes $c_1, \ldots, c_k$, then the linear extensions are multiset permutations with frequencies $c_1, \ldots, c_k$ (see Section 5.6), obtained by identifying the elements of each chain in the linear extension; see Figure 10 (k). Furthermore, linear extensions of a $2 \times k$ grid tilted by 45 degrees correspond to Catalan objects. Indeed, if we regard the k elements in the lower chain as opening brackets and the k elements of the upper chain as closing brackets, then every linear extension corresponds to a well-formed parenthesis expression with k pairs of parentheses. By subdividing the lower chain and appending additional minima, this construction can be generalized to encode b-ary Catalan objects for any $b \geq 2$.

Moreover, note that if the poset P is a fence, i.e., $1 < 2 > 3 < 4 > \cdots$, then its linear extensions are the inverses of alternating permutations. Similarly, all Young tableaux of a given shape arise as the inverse permutations of linear extensions of the tableau turned so that its corner is the minimum; see Figure 10 (l). Lastly, partitions of the ground set $[k \cdot c]$ into k sets of size c are obtained as inverse permutations of the linear extensions of the comb poset, obtained from k chains of size c by combining their minima into a chain of size k. This idea can be generalized to partitions of $[n]$ into subsets of arbitrary prescribed sizes $c_1, \ldots, c_k$, by forming one comb for every group of identical integers c_i as before, and taking the disjoint union of those combs.

Clearly, the flip graph of linear extensions under transpositions is bipartite, with the partition classes given by the parity of the permutations. We say that a poset is *balanced* if the two partition classes have the same size. Ruskey [Rus88b] conjectured that all balanced posets admit a transposition Gray code, and he also asked whether an adjacent transposition Gray code exists; see also [JSH09, Problem 485]. This conjecture still stands as one of the most challenging open problems in this area, and has been confirmed for several special cases. In [Rus92], Ruskey also raised a seemingly easier version of his conjecture, asking for a perfect matching in the flip graph of balanced posets. Ruskey's original conjecture is known to be true for the $2 \times k$ grid [RP90], which is balanced if and only if k is even. It is also known for k disjoint chains of sizes $c_1, \ldots, c_k$, $k \geq 2$, which form a balanced poset if and only if at least two of the c_i are odd [Rus92], and this positive result also holds for the more restrictive requirement of adjacent transpositions [Sta92]. Ruskey [Rus92] also showed that if P is a forest, i.e., the Hasse diagram of P is a collection of rooted trees, then P is balanced if and only if the forest does not admit a matching in which all but at most one of the trees' roots are matched. To state the next result, we say that a poset is *ranked*, if it admits a rank function that assigns rank 0 to all maximal elements and the rank changes by -1 when moving down any cover relation. Pruesse and Ruskey [PR91] confirmed the conjecture for ranked posets P in which every non-minimal element has at least two lower covers or every non-maximal element has at least two upper covers. Using the results from [RS93], this statement even holds for adjacent transpositions. This confirmed case of the conjecture generalizes Ruskey's [Rus89] earlier transposition Gray code for alternating permutations for odd n, obtained when P is an odd fence. It also applies to forest posets in which no vertex has exactly one child, i.e., leaves have none, and all other vertices have at least two. West [Wes93] gives another proof for such forest posets, by showing that if the linear extensions of P admit an adjacent transposition Gray code, then the poset $P \wedge Q$ also has this property, which is obtained by making a single element of P cover an antichain Q of size at least 2. In a similar vein, Stachowiak [Sta92] showed that if a poset P has an even number of linear extensions that can be listed in adjacent transposition order, then for any poset Q, the linear extensions of the disjoint union $P \cup Q$ also admit an adjacent transposition Gray code. Another relevant result due to Pruesse and Ruskey [PR94] is that the linear extensions of any poset P can be generated by one or a product of two adjacent transpositions in each step, and this scheme can be implemented efficiently (cf. [CW95, KL02]); see Figure 10 (k)+(l). They proved this by showing that the Cartesian product of the linear extension flip graph $G(P)$ with a single edge, i.e., the *prism* of $G(P)$, has a Hamilton cycle. As $G(P)$ is bipartite, this easily implies that the square $G^2(P)$ has a Hamilton cycle.

An *ideal* of a poset P is a downward-closed subset of P. Koda and Ruskey [KR93] constructed a Gray code for the ideals of a forest poset, where consecutive ideals differ by exactly one element. Pruesse and Ruskey [PR93] showed that the ideals of any poset can be listed by changing (i.e., adding or removing) at most two elements in each step, by proving that the prism of the ideal flip graph admits a Hamilton cycle. They also proved that such a listing exists for the ideals of fixed

size of a series-parallel poset. Habib, Nourine and Steiner [HNS97] generalized this result about ideals of fixed size to interval orders, and they showed that all ideals of a poset can be listed in non-decreasing order of cardinality by changing at most two elements in each step. Pruesse and Ruskey [PR93] also conjectured that for any poset, its ideals of any fixed size can be listed by element exchanges. Furthermore, they conjectured that the ideals of size k or $k+1$ admit a Gray code by element addition/removal, provided that there is the same number of ideals of each of the two sizes. This generalizes the middle levels conjecture discussed in Section 4.5, obtained when the poset is an antichain of size $2k+1$. It turns out that there are easy counterexamples for both of these conjectures; see Figure 13 (a) and (b), respectively. For the first one, take the 6-element crown poset and its 3-element ideals. The flip graph formed by element exchanges is a star with 3 rays, so it does not have a Hamilton path. For the second one, consider the 7-element poset formed by four chains of sizes $3, 2, 1, 1$, which has 11 ideals of size 3 and 4 each, but the bipartite graph formed by them through element addition/removal does not admit a Hamilton path. Nevertheless, we may still ask, for which posets can the ideals of fixed size be listed by element exchanges, or the ideals of sizes k or $k+1$ be listed by element addition/removal?

(a)   (b)
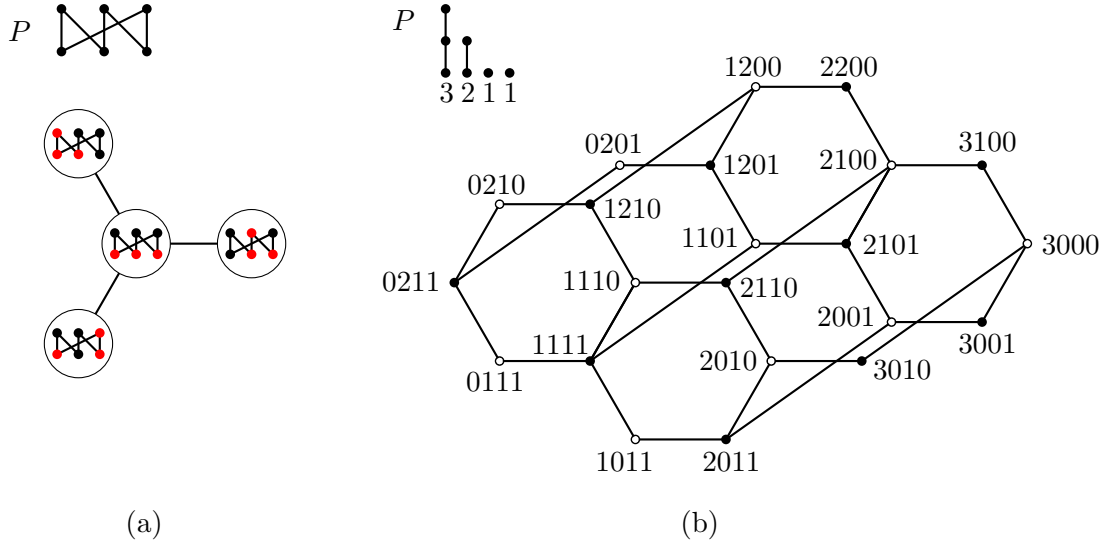
FIGURE 13. Counterexamples to two conjectures of Pruesse and Ruskey [PR93]. In (b), the ideals of the poset P are encoded by specifying the number of elements taken from each of the four chains of P.

Also, it is a long-standing open problem whether the ideals of any poset can be generated in constant amortized-time per ideal, either in a Gray code fashion or not.

5.6. **Multiset permutations.** A useful generalization of permutations are multiset permutation, which allow repetition of symbols. Formally, a multiset permutation over the alphabet $[k]$ with frequencies $c_1, \ldots, c_k \geq 1$ is a string that contains exactly c_i entries i for all $i \in [k]$. Permutations are the special case where $c_1 = \cdots = c_k = 1$. Multiset permutations also generalize combinations, which are obtained for $k = 2$.

Ko and Ruskey [KR92] describe a genlex Gray code for multiset permutations that uses transpositions, and that is constructed from a generation tree. Takaoka [Tak99a] used a similar idea to construct another transposition-based genlex Gray code, which moreover yields a loopless algorithm. Ruskey [Rus03] also devised a Gray code that operates by homogeneous transpositions, i.e., whenever entries a_i and a_j are transposed, all entries in between satisfy

$a_{i+1}, \ldots, a_{j-1} \leq \min\{a_i, a_j\}$. This construction is a generalization of the Eades-McKay homogeneous transposition Gray code for combinations; see also [Vaj03, KL04b]. Already much earlier, Chase [Cha70b] generalized his Gray code for combinations to multiset permutations, such that it only uses transpositions of the form $a_i, a_{i+1} \leftrightarrow a_{i+1}, a_i$ or $a_{i-1}, a_i, a_{i+1} \leftrightarrow a_{i+1}, a_i, a_{i-1}$ with $a_i = \min\{a_{i-1}, a_{i+1}\}$.

Lehmer [Leh65] raised the question under which conditions multiset permutations can be generated by adjacent transpositions. He phrased the problem in terms of a motel, which hosts k families with c_i members for each $i \in [k]$, and the task is to create all possible assignments of these guests to $n := \sum_{i \in [k]} c_i$ rooms, where changes take place only between adjacent rooms. Stachowiak [Sta92] answered this question completely, proving that an adjacent transposition Gray code exists if and only if at least two of the c_i are odd, or in the trivial cases $k = 1$, or $k = 2$ and $\min\{c_1, c_2\} = 1$; see also [Rus92].

Williams [Wil09a] showed that multiset permutations can be generated by prefix shifts, and the average length of the shifts is less than 3 when $c_1 = \cdots = c_k = 1$. He obtained this result by generalizing his cool-lex order to multiset permutations. In his thesis [Wil09b] (results also published in [SW23]), he also shows that prefix shifts of length $n - 1$ or n are sufficient to generate all multiset permutations, which generalizes the aforementioned result for permutations published in [RW10].

Shen and Williams [SW21] conjectured that multiset permutations with equal frequencies $\boxed{\text{P38}}$
$c_1 = \cdots = c_k =: c$ can be generated by star transpositions. This generalizes Ehrlich's star transposition Gray code for permutations ($c = 1$), and the middle levels conjecture ($k = 2$). Such star transposition Gray codes were constructed by Gregor, Merino and Mütze [GMM23a] for all frequencies satisfying $c_i \leq 4$ for $i \in [k]$ and $2 \cdot \max_{i \in [k]} c_i \leq \sum_{i \in [k]} c_i$, settling in particular the cases $c = 2, 3, 4$ of Shen and Williams' conjecture. All other cases are still open.

5.7. Other generalizations and variants.

A *signed permutation* is a permutation in which every entry has a sign $\{+, -\}$. Korsh, LaFollette and Lipschutz [KLL11] combined the BRGC and the SJT algorithm to a Gray code for signed permutations that operates by adjacent transpositions or inverting the sign of the last entry.

In a complemented prefix reversal we reverse a prefix of a signed permutation and complement all of its signs. In terms of the pancake flipping analogy, we can imagine pancakes with one side burnt and the other side unburnt, so flipping them around brings up the other side. Suzuki, N. Sawada and Kaneko [SSK05] showed that the flip graph on signed permutations under those operations is Hamilton-connected. Moreover, J. Sawada and Williams [SW16a] describe two simple algorithms for computing a Hamilton cycle in this graph. The first algorithm greedily applies complemented prefix reversals to the *shortest* possible prefix, so that a new signed permutation is generated. The average length of the reversed prefixes approaches $\sqrt{e} = 1.64872\ldots$. Interestingly, the Ord-Smith/Zaks ordering of (unsigned) permutations is also obtained by such a greedy rule. Moreover, it is shown that greedily reversing (and complementing) the *longest* possible prefix also yields a Gray code listing of permutations (and signed permutations, respectively). In the follow-up paper [SW16b], Sawada and Williams conjectured that signed permutations $\boxed{\text{P39}}$
can be generated using only complemented prefix reversals of lengths n, $n - 1$ or $n - 2$.

Cameron, Sawada and Williams [CSW21] generalized these results to *colored permutations*, where each entry of the permutation has one of k available colors. For $k = 1$ these are (uncolored) permutations and for $k = 2$ these are signed permutations. This paper establishes that the greedy strategy that modifies the shortest possible prefix by reversing and cyclically shifting colors works for any number k of colors, and the average length of the reversed prefixes approaches $e^{1/k}$.

Gutierres, Mamede and Santos [GMS18] present a Gray code for signed involutions that uses a transposition, a 3-rotation and/or a sign change in each step.

A *weak order* is a ranking of n objects where ties are allowed. For example, all thirteen weak orders on three elements $\{1, 2, 3\}$ are $1 \equiv 2 \equiv 3$, $1 \equiv 2 \prec 3$, $1 \prec 2 \equiv 3$, $1 \prec 2 \prec 3$, $1 \equiv 3 \prec 2$, $1 \prec 3 \prec 2$, $2 \prec 1 \equiv 3$, $2 \prec 1 \prec 3$, $2 \equiv 3 \prec 1$, $2 \prec 3 \prec 1$, $3 \prec 1 \equiv 2$, $3 \prec 1 \prec 2$, $3 \prec 2 \prec 1$, where $\prec$ denotes a strict precedence and $\equiv$ denotes a tie. The number of weak orders is given by the Fubini numbers [OEIS 000670]. Any weak order can be described by a string $a_1, \ldots, a_n$, where a_i denotes the number of $\prec$ symbols that precede i in the weak order (these strings are called *Cayley permutations* in [MF84]). The corresponding strings for the weak orders for $n = 3$ listed before are $000, 001, 011, 012, 010, 021, 101, 102, 100, 201, 110, 120, 210$. Baril [Bar03] and Jacques, Wong and Woo [JWW20] describe 2-Gray codes for weak orders in this string representation, i.e., any two consecutive strings differ in at most two positions. The following two

$\boxed{\text{P40}}$ challenging open problems are mentioned in [JSH09, Problem 482]: Savage raised the question whether there is a 1-Gray code for weak orders. For $n = 3$ such a (non-cyclic) Gray code is $000, 001, 201, 101, 102, 100, 120, 110, 210, 010, 012, 011, 021$, and computer experiments show that

$\boxed{\text{P41}}$ they exist also for $n = 4, 5, 6$ (and these are even cyclic). Moreover, Knuth asked for a 2-Gray code that uses only adjacent transpositions $a_i \leftrightarrow a_{i+1}$ and adjacent left-assignments $a_i := a_{i+1}$; see also [Knu11, Ex. 106 in Sec. 7.2.1.2].

Cardinal, Merino and Mütze [CMM22] consider *partial permutations*, which are all linearly ordered subsets of $[n]$ of arbitrary size [OEIS A000522]. Their Gray code uses either an adjacent transposition, deletion or insertion of a trailing element in each step.

Stamp foldings, *semi-meanders* and *open meanders* are permutations that are characterized by avoiding certain geometric patterns (self intersections), which can be captured by permutation

$\boxed{\text{P42}}$ patterns with additional parity constraints. Sawada and Li [SL12] raised the question which of these families of objects admit Gray codes. Liu and Wong [LW23] made progress towards this problem, by providing Gray codes for stamp foldings and semi-meanders; see Figure 11 (n)+(n'). Their code operates by cyclic substring shifts (by more than one position) in the permutation representation.

6. Set partitions

A *set partition* is a partition of $[n]$ into a disjoint union of nonempty sets. We consider every set partition in its canonic representation, where every subset is sorted increasingly, and the subsets are sorted increasingly by their smallest elements. A *restricted growth string* (RGS) is a string $a_1, \ldots, a_n$ such that $a_1 = 0$ and $a_i \leq 1 + \max\{a_1, \ldots, a_{i-1}\}$ for all $i = 2, \ldots, n$. It is well known that set partitions are in bijection with RGSs. Indeed, given a set partition in canonic representation, we give labels $j = 0, 1, \ldots$ to each subset according to the canonic ordering, and we then define $a_i := j$ if the element i is in the subset labeled j. For example, the set partition $\{1, 7, 8\}, \{2, 3, 5, 6\}, \{4\}, \{9\}$ of $[9]$ corresponds to the RGS 011211003. In this section we consider Gray codes for all set partitions (unrestricted) and some restricted variants obtained by fixing the number of parts or forbidding certain patterns, as well generalizations of RGSs. Unrestricted set partitions are counted by the Bell numbers [OEIS A000110], whereas set partitions with a fixed number of parts are counted by the Stirling numbers of the second kind [OEIS A008277]. Furthermore, non-crossing and non-nesting partitions are counted by the Catalan numbers, and non-crossing and non-nesting partitions with a fixed number of parts are counted by the Narayana numbers [OEIS A001263].
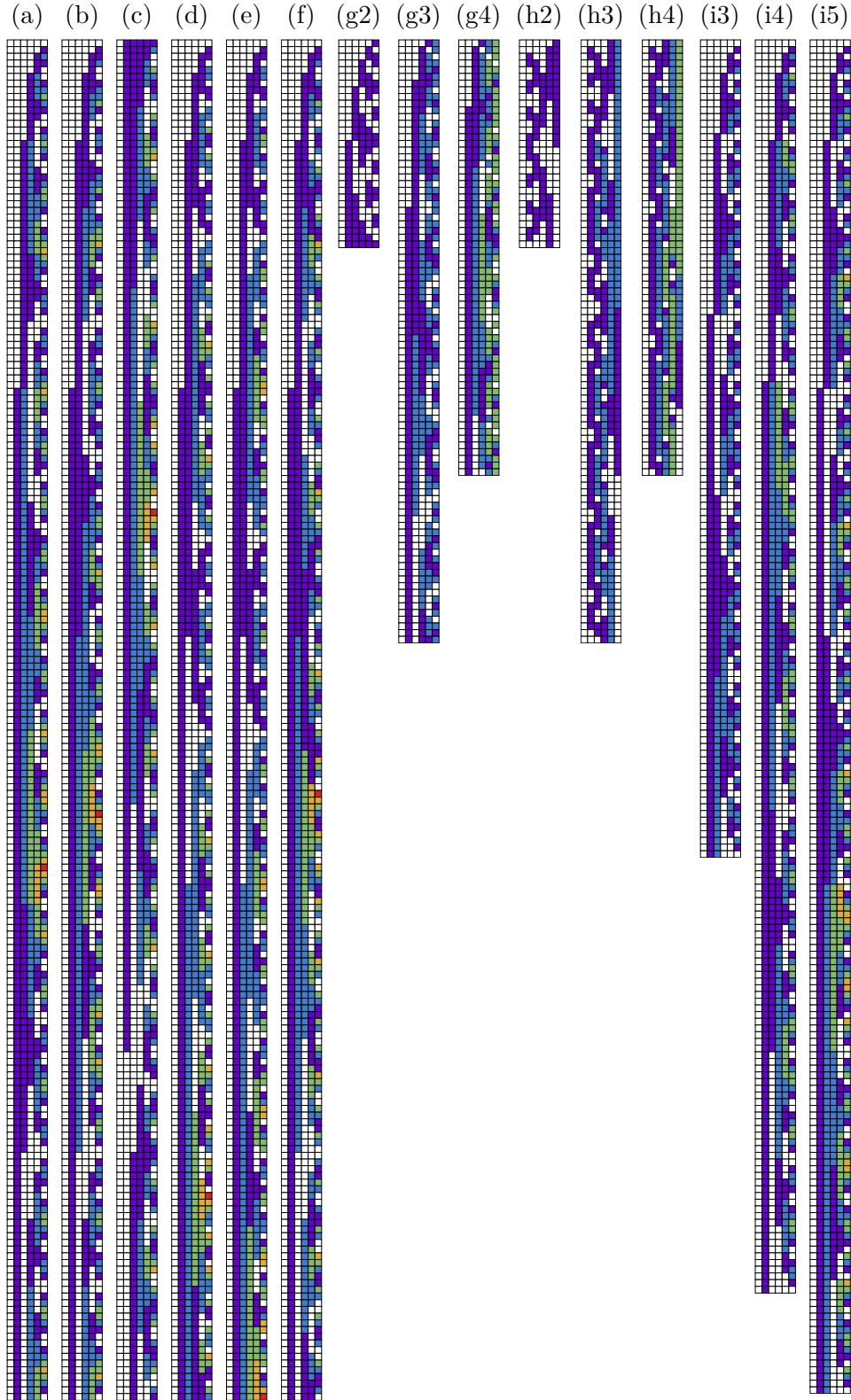
FIGURE 14. Gray codes for set partitions of [6] or [5] represented by restricted growth strings (0=white, 1=purple, 2=blue, 3=green, 4=yellow, 5=red, 6=brown): (a) Kaye; (b) Ruskey; (c) Ehrlich; (d) Mansour-Nassar; (e) greedy (Williams); (f) greedy (Kerr-Sawada); (g2)–(g4) with 2, 3 or 4 parts (Ehrlich); (h2)–(h4) with 2, 3 or 4 parts (Ruskey); (i3)–(i5) with $\leq$ 3, 4 or 5 parts (Sabri-Vajnovszki); (figure continues on next page)
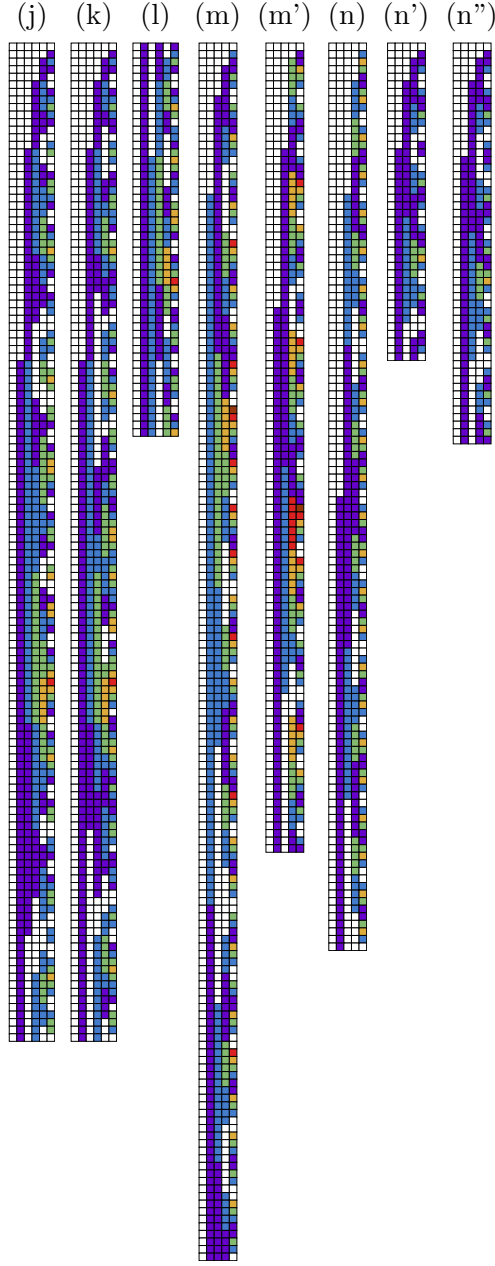
FIGURE 14. (continued) (j) non-crossing; (k) non-nesting; (l) sparse; (m)+(m') restricted growth functions: $b = (b_2, b_3, b_4, b_5) = (2, 1, 1, 2)$ and $b = (b_2, b_3, b_4, b_5) = (1, 1, 3, 1)$; (n) subexcedant sequences: $f(a_1, \ldots, a_{i-1}) = i - 2$; (n') staircase strings: $f(a_1, \ldots, a_{i-1}) = a_{i-1}$; (n") ascent sequences: $f(a_1, \ldots, a_{i-1}) = |\{1 \leq j < i - 1 \mid a_j < a_{j+1}\}|$.

6.1. **Unrestricted set partitions.** Kaye [Kay76] describes a Gray code for set partitions where in each step, one element moves from one subset to an adjacent subset (in the canonic representation); see Figure 14 (a). This Gray code for the partitions of $[n]$ can be constructed inductively by considering the listing for partitions of $[n-1]$ and adding n to every subset and as a singleton, alternately from left to right or vice versa. In Kaye's paper and in [Wil89], this construction idea is attributed to Knuth, who solved the corresponding problem stated in [NW75, p. 292, Problem 35]. The corresponding RGS listing is genlex, but consecutive RGSs may differ

in multiple positions. Ruskey [Rus03] proposed a simple modification of this ordering which maintains the genlex property and ensures that consecutive RGSs differ in exactly one position, and by ± 1 or ± 2 in that position; see Figure 14 (b). Already earlier, Ehrlich [Ehr73b] presented a Gray code for set partitions which is genlex and consecutive RGSs differ in exactly one position, either by ± 1 or the value in that position changes from 0 to the maximum value or vice versa; see Figure 14 (c). In the canonic set representation, this means that an element moves to a circularly adjacent subset, i.e., the first and last subset are also considered adjacent. Ehrlich also proved that under the stricter requirement that two consecutive RGSs differ only by ± 1, Gray codes do not exist for infinitely many values of n because of balancedness problems. Mansour and Nassar [MN08] present another genlex Gray code for set partitions in which consecutive RGSs differ in exactly one position.

Williams [Wil13] describes a genlex Gray code for set partitions that is produced by starting at the partition $\{1, 2, \ldots, n\}$, and then repeatedly and greedily moving the largest possible element to the leftmost possible set so that a new set partition is created; see Figure 14 (e). In the corresponding RGS listing, some strings differ in more than one position (in the listing for $n = 6$ shown in the figure, there is precisely one such transition from the 73rd to the 74th string, which differ in the last two positions). Williams' paper claims that this listing is the same as Kaye's, but this is not the case. In particular, Williams' listing always ends with the partition $\{1\}, \{2\}, \ldots, \{n\}$, unlike Kaye's. Yet another genlex greedy Gray code was communicated to the author by Joe Sawada and his student Kevin Kerr in 2021 [Ker21]. It starts with 0^n and then repeatedly and greedily changes the rightmost possible entry (and only that entry) to the largest possible value so that a new RGS is created; see Figure 14 (f).

6.2. **Fixed number of parts.** Ehrlich's paper [Ehr73b] also describes a Gray code for set partitions with a fixed number of parts; see Figure 14 (g2)–(g4). The sets differ in moving one or two elements to another subset, and the corresponding RGS listings are genlex and differ in at most two positions in each step. His construction works more generally for set partitions whose number of parts is in any fixed interval. Ruskey [Rus93] presents another Gray code for this problem in which the RGSs differ in exactly one position in each step, but this listing is not genlex; see Figure 14 (h2)–(h4). Ruskey and Savage [RS94b] showed that a Gray code with only single ± 1 transitions in the RGS representation does not exist in general.

Sabri and Vajnovszki [SV17] consider Gray codes for set partitions with a fixed number of parts or an upper bound on the number of parts. They constructed a 5-Gray (in the RGS representation) for the first family and a 3-Gray code for the second family, based on sublists of generalizations of the BRGC, implying that both of these Gray codes are genlex; see Figure 14 (i3)–(i5).

6.3. **Non-crossing, non-nesting and sparse partitions.** Huemer, Hurtado, Noy and Omaña-Pulido [HHNOP09] developed Gray codes for non-crossing set partitions. A set partition is *non-crossing* if for any two sets A and B of the partition and any four elements $a, a' \in A$ and $b, b' \in B$ we do not have $a < b < a' < b'$. The authors construct a cyclic Gray code for non-crossing partitions, valid for all $n \geq 3$, that moves one element of a subset to another in each step. A non-cyclic Gray code for this problem can be obtained easily, by following Kaye's construction described in Section 6.1 and skipping set partitions that have crossings, which yields a genlex listing; see Figure 14 (j). This paper also construct a cyclic Gray code for non-crossing partitions of fixed size k, for all $n > k \geq 3$, allowing however one additional flip operation, namely to exchange two elements between two subsets, i.e., $A \cup \{a\}, B \cup \{b\} \longrightarrow A \cup \{b\}, B \cup \{a\}$. The authors conjectured that the latter problem can be solved for all $n > k \geq 3$ without using this extra type of flip operation.

P43

Complementing the aforementioned results, Conflitti and Mamede [CM15] consider *non-nesting* set partitions, which are set partitions such that for any two sets A and B of the partition and any four elements $a, a' \in A$ and $b, b' \in B$ with $[b, b'] \cap A = \emptyset$ we do not have $a < b < b' < a'$. They present a Gray code for non-nesting partitions that moves one element of a subset to another in each step; see Figure 14 (k). The construction is a straightforward variation of Kaye's construction, and it also yields a genlex listing. The authors used the same idea to construct a genlex Gray code for *sparse* set partitions, which are partitions such that no subset contains two consecutive elements; see Figure 14 (l). Interestingly, there is a simple bijection between sparse set partitions of $[n]$ and unrestricted set partitions of $[n-1]$. In particular, sparse set partitions are also counted by the Bell numbers. In addition to these simple non-cyclic constructions, Conflitti and Mamede also provide constructions of cyclic Gray codes for both problems. In a later paper, Conflitti and Mamede [CM17] present similar Gray codes for non-crossing and non-nesting partitions in Weyl groups of type B and D.

P44   For all the classes of restricted set partitions discussed in this section, it is open whether they admit Gray codes that change only few entries in the RGS representation in each step.

6.4. **Generalizations.** For any integer $k \geq 0$, a *restricted growth tail* is a string $a_1, \ldots, a_n$ such that $a_1 \leq k$ and $a_i \leq 1 + \max\{a_1, \ldots, a_{i-1}, k-1\}$ for all $i = 2, \ldots, n$. Obviously, for $k = 0$ these are the same as RGSs. Ruskey and Savage [RS94b] present Gray codes for restricted growth tails, valid for any $n \geq 1$ and $k \geq 0$, which satisfy Ehrlich's conditions mentioned in Section 6.1, i.e., a single position changes in each step, either by ± 1 or the value in that position changes from 0 to the maximum value of at least k or vice versa. These Gray codes are moreover cyclic.

Mansour, Nassar and Vajnovszki [MNV11] consider another generalization of RGSs, called *restricted growth functions*, for which there is a vector $b = (b_2, \ldots, b_n)$ with $b_i \geq 1$ for all $i = 2, \ldots, n$, and the requirement for the string $a_1, \ldots, a_n$ is that $a_1 = 0$ and $a_i \leq b_i + \max\{a_1, \ldots, a_{i-1}\}$ for all $i = 2, \ldots, n$. Clearly, for $b = (1, \ldots, 1)$ we recover standard RGSs. They present genlex Gray codes that are valid for any vector b, and those Gray codes change only one entry of the RGS representation at a time; see Figure 14 (m)+(m'). For $b = (1, \ldots, 1)$ the obtained listings coincide with the Gray code of Kerr-Sawada mentioned before, and they can also be constructed greedily, by starting with 0^n and then repeatedly and greedily changing the rightmost possible entry to the largest possible symbol.

Subsequently, Mansour and Vajnovszki [MV13] introduced an even more general concept, called *restricted growth words*. They consider strings $a_1, \ldots, a_n$ such that $a_1 = 0$ and $a_i \leq 1 + f(a_1, \ldots, a_{i-1})$, where f is any non-negative function of $a_1, \ldots, a_{i-1}$. Their paper describes a genlex Gray code for any such family of strings, which changes only one entry at a time, either by ± 1 or ± 2. For $f = \max$ we obtain RGSs. For $f = i - 2$ we obtain *subexcedant sequences*, which are counted by the factorial numbers; see Figure 14 (n). For $f = a_{i-1}$ we obtain *staircase words*, which are another Catalan family; see Figure 14 (n'). For $f = 0$ we obtain bitstrings with first bit 0, which are counted by powers of 2, and the resulting listing is the (first half of the) BRGC. For $f = |\{1 \leq j < i - 1 \mid a_j < a_{j+1}\}|$ we obtain *ascent sequences*, which are counted by the Fishburn numbers [OEIS A022493]; see Figure 14 (n"). Interestingly, the Kerr-Sawada greedy approach also works in this more general setting, yielding different listings, however.

Sabri and Vajnovszki [SV15] showed that restricting generalizations of the BRGC to RGSs, subexcedant sequences, staircase words or ascent sequences yields genlex 3-Gray codes.

7. Integer partitions and compositions

In this section we discuss Gray codes for integer partitions and integer compositions.

7.1. **Integer partitions.** A *partition* of n is a sequence $a_1, \ldots, a_k$ of integers such that $a_1 \geq a_2 \geq \cdots \geq a_k \geq 1$ and $a_1 + \cdots + a_k = n$. Integer partitions are counted by the partition numbers [OEIS A000041].

Wilf asked whether integer partitions of n admit a Gray code with transitions of the form $a_i, a_j \leftrightarrow a_i - 1, a_j + 1$ in each step, which includes transitions $a_i(, 0) \leftrightarrow a_i - 1, 1$ that change the number of parts. This problem was solved by Savage [Sav89], who showed more generally that for all $n \geq k \geq 1$, all partitions of n with parts of size at most k admit such a Gray code. Her results also yield Gray codes for all partitions of n with largest part equal to k, all partitions of n into at most k parts, and all partitions of n into exactly k parts.

Rasmussen, Savage and West [RSW95] constructed a Gray code for partitions of n into parts of size at most k and congruent to 1 modulo d, where $d \geq 1$ is some integer parameter. For $d = 1$ this is the same problem as before. For $d = 2$ these are partitions into odd parts of size at most k. Their Gray code uses transitions $a_i, a_j \leftrightarrow a_i - d, a_j + d$ and $a_i(, 0^d) \leftrightarrow a_i - d, 1^d$. Furthermore, the authors present a Gray code for partitions of n into distinct parts of size at most k, using transitions $a_i, a_j \leftrightarrow a_i - 1, a_j + 1$ as before. It is a well-known result of Euler that the number of partitions of n into odd parts is the same as the number of partitions of n into distinct parts [OEIS A000009].

The paper [RSW95] also raised the problem to construct Gray codes for the following five families of restricted partitions of n: exactly k distinct parts; distinct odd parts; distinct parts congruent to 1 modulo d; at most c copies of each part; parts congruent to 1 modulo d and at most c copies of each part. Savage [Sav97] conjectured that such codes exist for all $n \geq k \geq 1$ and $c, d \geq 1$. $\boxed{\text{P45}}$

A compact representation of an integer partition $a_1, \ldots, a_k$ is obtained by grouping together runs of a_i that have the same value, and recording their multiplicities, i.e., we obtain a list of pairs $(a'_1, b_1), \ldots, (a'_\ell, b_\ell)$ such that $a'_1 > \cdots > a'_\ell \geq 1$, $b_i \geq 1$ for $i = 1, \ldots, \ell$ and $a'_1 \cdot b_1 + \cdots + a'_\ell \cdot b_\ell = n$. Ruskey [Rus03] observed that ordering partitions lexicographically, the corresponding compact representations differ at most in the last three pairs of integers.

7.2. **Integer compositions.** A *composition* of n is a sequence of integers $a_1, \ldots, a_k$ such that $a_1, \ldots, a_k \geq 1$ and $a_1 + \cdots + a_k = n$, i.e., in contrast to partitions, the order of the summands matters. The number of compositions of n is 2^{n-1}, and there is a simple bijection between bitstrings of length $n - 1$ and compositions: Specifically, for any bitstring x, we consider the lengths of all maximal non-overlapping substrings of $x1$ that end with 1. For example, for the string $x = 1101000110000$ of length 13 we have $x1 = 1|1|01|0001|1|00001$, where the vertical bars indicate the boundaries between maximal non-overlapping substrings ending with 1, and therefore x is mapped to the composition $1, 1, 2, 4, 1, 5$ of 14. Observe that the BRGC has the interesting property that for every flip $1 \leftrightarrow 0$, the flipped bit is the last one or at least one of the bits next to it is also a 1. Applying the aforementioned bijection to the BRGC, we thus obtain a cyclic genlex Gray code for compositions that uses only transitions $a_i, 1 \leftrightarrow a_i + 1$, where the 1 is at positions $i + 1$ or $i - 1$, respectively.

If we consider compositions of n with a fixed number k of parts, then the above bijection shows that they correspond to $(n - 1, k - 1)$-combinations. Using the Gray codes for combinations discussed in Section 4.1 that use transitions $01 \leftrightarrow 10$ or $001 \leftrightarrow 100$, we obtain Gray codes for compositions with transitions $a_i, a_{i+1} \leftrightarrow a_i - d, a_{i+1} + d$ with $d = \pm 1$ or $d = \pm 2$, respectively (recall that using only the former requires some conditions on n and k).

Klingsberg [Kli82] describes a genlex Gray code for compositions into a fixed number of parts that uses only transitions $a_i, a_j \leftrightarrow a_i - 1, a_j + 1$ (cf. also [MN10]). In Klingsberg's paper and

in [Wil89], the idea for this construction is attributed to Knuth, who solved the corresponding problem stated in [NW75, p. 292, Problem 34].

Ruskey and Savage [RS96] consider compositions $a_1, \ldots, a_k$ of n with a fixed number k of parts, with the additional requirement that the parts are bounded, i.e., there is a vector $b = (b_1, \ldots, b_k)$ such that $a_i \leq b_i$ for all $i = 1, \ldots, k$. For $b_i \geq n$ this restriction becomes irrelevant, so we obtain compositions of n into k parts. Moreover, for $b_1 = \cdots = b_k = 2$ these are $(k, n-k)$-combinations (by changing the alphabet $1, 2 \to 0, 1$). The authors construct a Gray code that uses transitions $a_i, a_j \leftrightarrow a_i - 1, a_j + 1$. They raise the problem to determine when an adjacent transition Gray code for this problem is possible, i.e., if we also require that $|j - i| = 1$ (in some cases, there are balancedness problems). Earlier, Ehrlich [Ehr73a, Ehr73b] presented an adjacent transition Gray code, but with changes that can be more than ± 1.

Walsh [Wal00] considered the same setting of bounded compositions as Ruskey and Savage, and he showed that for any vector b, the sublist of Klingsberg's ordering obtained by selecting only the compositions that satisfy the upper bounds given by b forms a genlex Gray code with transitions $a_i, a_j \leftrightarrow a_i - 1, a_j + 1$. By choosing an appropriate b and interpreting every composition as an inversion vector of a permutation, he used this to derive a Gray code for permutations with a fixed number of inversions; see Section 5.3.

Vajnovszki and Vernay [VV11] showed more generally that we may start with a straightforward generalization of the BRGC for tuples $a_1, \ldots, a_k$ with $1 \leq a_i \leq b_i$ (recall Section 3.19), and consider the sublist of all tuples that satisfy $s \leq a_1 + \cdots + a_k \leq t$, and this sublist will be a genlex Gray code with transitions $a_i \leftrightarrow a_i \pm 1$ or $a_i, a_j \leftrightarrow a_i - 1, a_j + 1$. In particular, Walsh's result for compositions follows from this by setting $s = t = n$. Furthermore, this also generalizes Tang and Liu's result about combinations discussed in Section 4.1 (for $b_1 = \cdots = b_k = 2$ and $s = t$), who obtained their Gray code for combinations by restricting the ordinary BRGC to bitstrings with fixed weight.

8. Geometric objects

In this section we consider Gray codes for geometric objects equipped with flip operations that are constrained by geometry, e.g., from a planarity requirement, and this makes the topic particularly interesting. For results on flips in planar graphs beyond Gray codes, we refer to the survey of Bose and Hurtado [BH09b]. In addition to the material in this section, non-crossing and non-nesting set partitions are discussed in Section 6.3.

8.1. **Triangulations and matchings.** We first consider the flip graph G_n that has as vertices all triangulations of a convex n-gon, and edges connecting triangulations that differ by a *flip*, i.e., a diagonal in a quadrilateral formed by two triangles is replaced by the other diagonal. The number of such triangulations is the $(n-2)$nd Catalan number. It is well known that G_n is the cover graph of a lattice, namely the Tamari lattice, and the skeleton of a polytope, namely the $(n-3)$-dimensional associahedron. By considering the dual graphs of the triangulations, we see that triangulations are isomorphic to binary trees with $n-2$ nodes, and edge flips correspond to tree rotations; see Figure 1 (f). Lucas [Luc87] proved that G_n has a Hamilton cycle, and her proof yields an efficient generation algorithm. Subsequently, Lucas, Roelants van Baronaigien and Ruskey [LRvBR93] provided a simpler proof for the existence of a Hamilton path in G_n, which yields a loopless algorithm. Possibly the simplest description of their path was given by Williams [Wil13] via a greedy algorithm, which is to repeatedly rotate the edge with the largest inorder label so that a new tree is generated; see also [CMM22] and Figure 12. Another proof for the existence of a Hamilton cycle in G_n was obtained by Hurtado and Noy [HN99] via a

generation tree. Huemer, Hurtado and Pfeifle [HHP08] later generalized this result to a flip graph of dissections of a convex polygon into $(b+1)$-gons under edge flips, whose dual graphs are b-ary trees under tree rotations.

Hernando, Hurtado and Noy [HHN02] consider the set of all non-crossing perfect matchings on a set of $n = 2m$ points in convex position, with the flip operation that replaces edges and non-edges along an alternating 4-cycle. They proved that the corresponding flip graph has a Hamilton cycle for every even $m \geq 4$, and no Hamilton path for any odd $m \geq 5$. Note that non-crossing matchings are also counted by the Catalan numbers.

Aichholzer, Aurenhammer, Huemer and Vogtenhuber [AAHV07] exhibit a set of six points in non-convex position for which the flip graph of triangulations is a star with three rays, so it does not admit a Hamilton path. They speculate whether Hamiltonicity holds for triangulations of every sufficiently large point set. Moreover, they conjectured that the flip graph of pseudo-triangulations under edge removals/insertions and exchanges has a Hamilton cycle, where a pseudo-triangulation has faces that are bounded by three concave chains. $\boxed{\text{P47}}$

Certain flip graphs for triangulations and non-crossing matchings on general point sets were considered by Houle, Hurtado, Noy and Rivera-Campo [HHNRC05], and they established that these graphs are connected, which implies 3-Gray codes by Sekanina's [Sek60] theorem discussed in Section 2.4.

Gregor, Merino and Mütze [GMM24] asked for the largest integer k for which there is a k-symmetric Hamilton cycle as defined in Section 3.13 in the associahedron. As a stepping stone, are there Gray codes for necklaces and bracelets of triangulations on a convex point set, i.e., for equivalence classes of these objects under cyclic rotation and/or reflection? The same two questions can naturally be asked for the flip graph of non-crossing matchings on a convex point set. $\boxed{\text{P66}}$ $\boxed{\text{P67}}$

8.2. Ordered trees. An *ordered* tree with n nodes has a root and a specified left-to-right ordering of the children of each node. These objects are counted by the $(n-1)$st Catalan number. Lapey and Williams [LW22a] provide a loopless Gray code algorithm for ordered trees that uses so-called pull operations, at most 2 in each step, and their method is derived from cool-lex order. Nakano [Nak24] gives a Gray code for ordered trees that uses single leaf moves, i.e., in each step a leaf of the current tree is removed and reattached elsewhere.

8.3. Dissections, plane graphs and spanning trees. Huemer, Hurtado, Noy and Omaña-Pulido [HHNOP09] consider Gray codes for dissections of a convex polygon. A *dissection* of a convex polygon with n vertices is a subdivision of the polygon by k non-crossing diagonals, where $0 \leq k \leq n-3$. We let G_n and $G_{n,k}$ be the graphs on the set of all dissections and those with exactly k diagonals, respectively. In $G_{n,k}$ two dissections are connected by an edge if they differ by removing one diagonal and adding another diagonal in the subpolygon that contains the end vertices of the removed diagonal. The graph G_n is the union over all $G_{n,k}$ for $k = 0, \ldots, n-3$ plus edges that correspond to either removing or adding one diagonal in the dissection. The authors prove that G_n has a Hamilton cycle for all $n \geq 4$, and $G_{n,k}$ has a Hamilton cycle for all $k \geq 1$ and $n \geq 4$. Note that the vertices of $G_{n,k}$ are precisely the $(n-3-k)$-dimensional faces of the associahedron (as a polytope), so this result generalizes the aforementioned result about the Hamiltonicity of the associahedron (the special case $k = n-3$).

Hernando, Hurtado, Marquez, Mora and Noy [HHM+99] proved that all plane spanning trees on a set of $n \geq 3$ points in convex position can be listed cyclically so that consecutive trees differ by exchanging one edge. Rivera-Campo and Urrutia-Galicia [RCUG01] proved the same result for plane spanning paths.
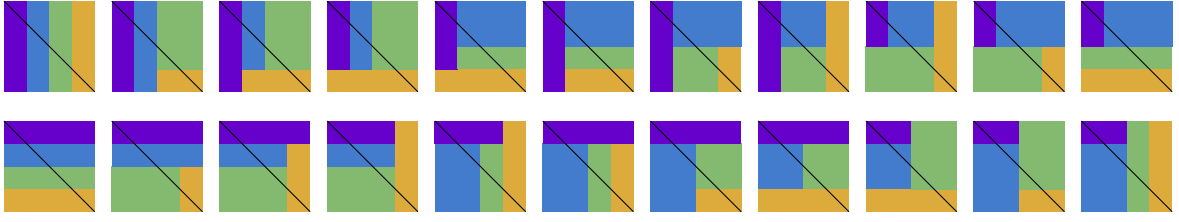
FIGURE 15. Gray code for diagonal rectangulations with $n = 4$ rectangles.

Aichholzer, Aurenhammer, Huemer and Vogtenhuber [AAHV07] consider point sets in general position, with plane graphs whose edges connect certain pairs of points by a straight line. In this setting, they obtained Gray codes via generation trees for three different families of graphs: all plane graphs by edge removals/insertions, all plane spanning trees by edge exchanges (generalizing the result from [HHM$^+$99] for convex position), and all connected plane graphs by edge removals/insertions or exchanges.

8.4. **Rectangulations and rainbow cycles.** Merino and Mütze [MM23a] present Gray codes for various families of *generic rectangulations*, which are subdivisions of a rectangle into n smaller rectangles, subject to the constraint that no four rectangles meet in a point. In particular, they constructed a Gray code for all generic rectangulations with n rectangles by flip operations called wall slides, simple flips and T-flips. They also obtained a Gray code for diagonal rectangulations, which are those generic rectangulations for which every rectangle intersects the diagonal from the top-left to the bottom-right corner of the outer rectangle, and this Gray code uses only simple and T-flips (diagonal rectangulations are counted by the Baxter numbers [OEIS A001181]); see Figure 15. More generally, their approach yields Gray codes for many subfamilies of generic rectangulations that are characterized by pattern avoidance, such as guillotine rectangulations or 1-sided rectangulations, by so-called jumps, which generalize wall slides, simple flips and T-flips. They raised the problem of constructing Gray codes for rectangulations of point sets as introduced in [ABP06] (see also [YRN18]). Downing, Einstein, Hartung and Williams [DEHW23] provide Gray codes for Catalan squares and staircases, which are a Catalan family of geometric objects that can be viewed as T-avoiding rectangulations.

Felsner, Kleist, Mütze and Sering [FKMS20] constructed cycles in the flip graphs of different geometric configurations that are balanced in the sense that each flip operation occurs equally often along the cycle. They refer to these cycles as *rainbow cycles*, and establish existence and non-existence results for the flip graphs of triangulations of a convex polygon, non-crossing perfect matchings on a convex set of points, and plane spanning trees on a general point set. They conjectured that the flip graph of non-crossing perfect matchings on a convex $2m$-gon under flips along alternating 4-cycles does not contain a cycle along which every possible edge on the point set appears and disappears exactly once (such a cycle has length $m^2/2$), whenever $m \geq 6$ is even, which is confirmed for $m \in \{6, 8, 10, 12\}$ (see also [MMP21]).

9. OBJECTS ON GRAPHS

In this section we discuss Gray codes for combinatorial objects that are defined with respect to a fixed graph H. We write K_n, P_n and C_n for the complete graph, path or cycle on n vertices, respectively, and $K_{a_1, a_2, \ldots, a_r}$ for the complete r-partite graph with partition classes of sizes $a_1, \ldots, a_r$.

9.1. **Spanning trees, cycles, and generalizations to matroids.** Cummins [Cum66] showed that the spanning trees of any fixed graph H can be listed cyclically by edge exchanges, i.e., in each step one edge is removed from the current spanning tree and another edge is inserted; see Figure 1 (g). Equivalently, his result asserts that the flip graph of the spanning trees of H under edge exchanges has a Hamilton cycle. Cummins proved more strongly that this flip graph is *edge-Hamiltonian*, i.e., for each of its edges, there is a Hamilton cycle containing this edge. A short proof of this result was given by Shank [Sha68]. Smith [Smi97] devised an algorithm for listing all spanning trees of a graph by edge exchanges, which requires only constant time on average per generated spanning tree. Details of his algorithm are described in [Knu11, Sec. 7.2.1.6].

Holzmann and Harary [HH72] generalized the aforementioned results to matroids, proving that the flip graph of the bases of a matroid under element exchanges has a Hamilton cycle. They showed more generally that if the flip graph has at least two cycles, then it is *Hamilton-uniform*, i.e., for every edge of the flip graph, there is a Hamilton cycle containing this edge, and a Hamilton cycle avoiding this edge. Furthermore, by the results from [NP81] discussed in Section 2.7, the matroid base flip graph is either Hamilton-connected or a hypercube, which is known to be Hamilton-laceable. Alspach and Liu [AL89] showed that for simple matroids, this flip graph is also *edge-pancyclic*, i.e., for every edge and every integer ℓ at least 3 and at most the number of vertices of the flip graph, there is a cycle of length ℓ through this edge; see also [BI76]. Note that for the uniform matroid, the matroid base flip graph is the Johnson graph, and in this case a Hamilton cycle corresponds to a transposition Gray code for combinations.

Many of the aforementioned Gray codes for spanning trees of a graph H are based on a straightforward recursion, already observed by Feussner [Feu02] (see also [Min65]), namely to consider the spanning trees that do not contain a fixed edge e, which are the spanning trees of $H - e$, and those that do contain e, which are the spanning trees of H/e. The graph $H - e$ is obtained from H by removing e, but none of its end vertices, whereas H/e is obtained from H by contracting the end vertices of e to a single vertex. Based on this recursion, Merino, Mütze and Williams [MMW22] discovered a strikingly simple greedy algorithm to list the spanning trees of H by edge exchanges, by considering their characteristic vectors: An edge exchange corresponds to a transposition of a 0 and 1 in the characteristic vector, and the rule is to perform a transposition that changes the shortest possible prefix so as to generate a previously unseen spanning tree. This amazing rule works for any graph H, for any edge ordering that determines the indices of the characteristic vector, for any initial spanning tree, and for any tie-breaking rule, it always yields a genlex listing, and it also generalizes straightforwardly to bases of a matroid (however, this Gray code is in general not cyclic).

Cameron, Grubb and Sawada [CGS21] asked whether there is a Gray code for spanning trees of a graph H that uses edge exchanges, with the additional requirement that the removed and added edge have a vertex of H in common. They refer to a listing satisfying this stronger requirement as a *pivot Gray code*, and they constructed such a Gray code for the family of fan graphs, which are obtained by joining a single vertex to all vertices of a path. Knuth [Knu11, Ex. 102 in Sec. 7.2.1.6] considers a directed graph and all of its spanning trees in which all arcs are oriented away from a fixed root vertex, also known as *out-trees* or *arborescences*. He asked for a Gray code to list those directed trees in Gray code order, exchanging one arc at a time. Note that the removed and added arc must point to the same vertex. We conclude that the existence of Knuth's Gray code would imply a pivot Gray code for undirected graphs (simply replace each undirected edge with two arcs in opposite directions).

Furthermore, Knuth [Knu11, Ex. 101 in Sec. 7.2.1.6] asked for a simple listing of the spanning trees of the complete graph K_n by edge exchanges. The precise definition of 'simple' is not

P50

P51

P52

specified, but it could mean fast ranking and unranking algorithms, and possibly also a more fine-grained explanation of Cayley's formula n^{n-2} for the number of all spanning trees.

Li and Liu [LL08a] consider the flip graph of circuits of a matroid, whose vertices are the circuits and two of them are connected by an edge if their intersection is nonempty. They proved that the circuit flip graph of a connected matroid is Hamilton-uniform, provided that there are at least four circuits; see also [FL13]. In the case of graphical matroids, a Hamilton cycle in the flip graph corresponds to a cyclic listing of the cycles of a graph so that any two consecutive cycles share at least one edge. Li and Liu [LL07] proved that the circuit flip graph of a connected matroid is also edge-pancyclic; see also [LL08b, LL12].

9.2. Perfect matchings.
For a graph H, we consider the flip graph $G(H)$ that has as vertices all perfect matchings of H, and two of them are connected by an edge in $G(H)$ if they differ by an alternating cycle. By the results from [NP81] discussed in Section 2.7, $G(H)$ is either Hamilton-connected or a hypercube, which is known to be Hamilton-laceable; see also [NP84] and [ZG86b]. Naddef [Nad84] showed that the flip graph $G(H)$ is also edge-pancyclic, provided that the set of edges of H that are contained in some perfect matching form a connected subgraph of H.

9.3. Hamilton cycles and Eulerian tours.
Fleischner, Horák and Širáň [FHŠ93] consider the flip graph $G(H)$ of Hamilton cycles of H, where any two cycles C and C' are connected by an edge in $G(H)$ if the symmetric difference of their edge sets is a 4-cycle in H, and the vertices of this 4-cycle appear consecutively on C and C'. They showed that $G(K_n)$ for $n \geq 4$ is edge-Hamiltonian (see also Sierksma's papers [Sie93, Sie98]). Note that a Hamilton cycle in $G(K_n)$ corresponds to a cyclic listing of rosary permutations of $[n]$ by circularly adjacent transpositions. It would be very interesting to investigate the Hamiltonicity of $G(H)$ for other graphs H, for example $H = K_{a,b}$.

Let H be a graph that admits a Eulerian tour, i.e., H is connected and each vertex has even degree. The graph H may also have multiple edges, but no loops. Zhang and Guo [ZG86a] consider the flip graph $G(H)$ that has as vertices all Eulerian tours of H without considering their orientation, i.e., a forward or backward traversal of the same tour is represented as a single vertex in $G(H)$. Moreover, two Eulerian tours are connected by an edge in $G(H)$ if they differ by reversing the orientation of traversing a closed proper subtour. This flip operation is known as *Kotzig transformation*. They proved that if H has at least three Eulerian tours, then $G(H)$ is edge-Hamiltonian. In [ZG87], the same authors also consider an analogous question for directed graphs. Let H be a digraph that admits a Eulerian tour, i.e., H is connected and for each vertex, the in-degree equals the out-degree. As before, multiple edges in H are allowed, but no loops. The flip graph $G(H)$ has as vertices all directed Eulerian tours of H, and two of them are connected if they differ by exchanging two directed subtrails between the same pair of vertices. The authors showed that if H has at least three directed Eulerian tours, then $G(H)$ is edge-Hamiltonian.

Guan and Pulleyblank [GP85] consider orientations of an undirected graph H that admit a directed Eulerian tour. The corresponding flip graph $G(H)$ has as vertices all Eulerian orientations of H, and two of them are connected in $G(H)$ if they differ by reversing an oriented (simple) cycle. They proved that $G(H)$ is either Hamilton-connected or a hypercube, which is known to be Hamilton-laceable.

9.4. Elimination orders and trees.
A *perfect elimination ordering* (PEO) of a graph is an ordering $v_1, \ldots, v_n$ of its vertices such that for every $i = 1, \ldots, n$, the neighbors of v_i among $v_1, \ldots, v_{i-1}$ form a clique. It is well known that a graph H admits a PEO if and only if H is chordal. Chandran, Ibarra, Ruskey and Sawada [CIRS03] present a Gray code for the PEOs of

any chordal graph that performs one or a product of two adjacent transpositions in each step. Their construction is based on the antimatroid Gray code from [PR93] (see also [Saw05]).

An *elimination tree* of a connected graph H is a rooted unordered tree, obtained by removing a vertex v of H, which becomes the root of the tree, and recursing on the connected components of $H - v$ that become the children of v in the tree. An *elimination forest* of a disconnected graph H is a collection of elimination trees, one for each connected component of H. Elimination trees generalize many combinatorial objects, such as binary trees (if H is a path), permutations (if H is a complete graph) and bitstrings (if H is a disjoint union of edges). Manneville and Pilaud [MP16] showed that if H has at least two edges, then its elimination trees can be generated cyclically by tree rotations, which naturally generalize classical rotations in binary trees. The corresponding flip graphs arise as skeleta of *graph associahedra*, which are a class of polytopes introduced by Carr and Devadoss [CD06, Dev09] and Postnikov [Pos09]. Cardinal, Merino and Mütze [CMM22] present a simple and efficient algorithm to compute a Gray code for the elimination forests of H by tree rotations for the case when H is a chordal graph, and this Gray code is cyclic if H is also 2-connected. This Gray code generalizes the Gray code for binary trees due to Lucas, Roelants van Baronaigien and Ruskey [LRvBR93], the SJT algorithm for permutations, and the BRGC for bitstrings.

9.5. **Acyclic orientations.** Savage, Squire and West [SSW93] consider a graph H and all of its acyclic orientations. The flip graph $G(H)$ has these acyclic orientations as vertices, and two of them are adjacent if the differ by the reversal of a single arc. Observe that for a graph H with m edges, $G(H)$ is a subgraph of the m-cube, in particular, $G(H)$ is bipartite. Moreover, if H is a forest, then $G(H)$ is the full m-cube, i.e., acyclic orientations are in 1-to-1 correspondence to bitstrings of length m, and reversing an arc corresponds to flipping a single bit. Observe also that if $H = K_n$, then acyclic orientations are in 1-to-1 correspondence to permutations of $[n]$, and reversing an arc corresponds to an adjacent transposition. For these two cases we immediately obtain Hamilton cycles in $G(H)$ by the BRGC and SJT algorithm, respectively. The aforementioned paper of Savage, Squire and West showed more generally that $G(H)$ has a Hamilton cycle for any chordal graph H, and the resulting orderings generalize the BRGC and SJT. To describe the other results, we introduce two more families of graphs: The *ladder* L_n is defined as the Cartesian product $L_n := P_n \times P_2$, and the *wheel* W_n is obtained from C_n by adding a new vertex adjacent to all others. The authors constructed Hamilton cycles in $G(C_n)$ and $G(L_n)$ for odd n, and a Hamilton path in $G(W_n)$ for odd n. On the negative side, they proved that the following flip graphs have no Hamilton path because of balancedness problems: $G(C_n)$, $G(L_n)$ and $G(W_n)$ for even n, $G(K_{a,b})$ for $a, b \geq 2$ if a or b is even. They conjectured that $G(W_n)$ ┃P54┃ has a Hamilton cycle for odd n (not just a Hamilton path), and they also raise the problem of finding a Hamilton path or cycle in $G(K_{a,b})$ when a and b are both odd. More generally, they conjectured that $G(H)$ has a Hamilton cycle if and only if its partition classes have the same size. This question was reiterated by Savage and Zhang [SZ98] and by Edelman (see [Sav97]). Towards this question, Squire [Squ94] showed that the square $G^2(H)$ has a Hamilton cycle for any graph H. Pruesse and Ruskey [PR95] strengthened this result and proved that the prism $G(H) \times P_2$, i.e., the Cartesian product of $G(H)$ with a single edge, admits a Hamilton cycle. As $G(H)$ is bipartite, this easily implies that $G^2(H)$ has a Hamilton cycle. Squire [Squ98] gives an efficient algorithm for computing a Hamilton cycle in the square $G^2(H)$, i.e., for listing all acyclic orientations of H by reversing at most two arcs in each step.

The problem of listing the acyclic orientations of H by arc reversals can be generalized so that it simultaneously generalizes the problem of listing the linear extensions of a poset by

adjacent transpositions discussed in Section 5.5. Specifically, for a graph H, we consider a subset F of edges of H to which we assign a fixed acyclic orientation, and we let $G(H, F)$ be the subgraph of $G(H)$ induced by acyclic orientations that preserve the orientation of the edges in F. Clearly, we have $G(H, \emptyset) = G(H)$, so this captures acyclic orientations as before. On the other hand, if F are the cover edges of a poset P on $[n]$, oriented upwards, then the acyclic orientations of $G(K_n, F)$ (i.e., we take H as the complete graph on $[n]$) are in 1-to-1 correspondence with the linear extensions of P, and reversing an arc corresponds to an adjacent transposition. Savage [Sav97] asked whether $G(H, F)$ admits a Hamilton cycle whenever its partition classes have the same size, but a counterexample was constructed by Naatz [Naa01]. Pruesse and Ruskey [PR95] also give an example showing that the prism of $G(H, F)$ need not be Hamiltonian, and Squire [Squ94] strengthened this by an example showing that the square $\boxed{\text{P55}}$ $G^2(H, F)$ need not be Hamiltonian. In general it is open under which conditions $G(H, F)$ or its prism or square admit a Hamilton path or cycle. As this generalizes Ruskey's conjecture discussed in Section 5.5, this is arguably a hard problem.

Pilaud [Pil22] equipped $G(H)$ with a poset structure, and found conditions on H for when $\boxed{\text{P56}}$ this poset is a lattice with various additional properties. He also describes the corresponding lattice congruences and realizations of their quotients as polytopes, and he raises the question whether they admit a Hamilton cycle. Substantial progress towards this problem was made by Cardinal, Hung, Merino, Mička and Mütze [CHM$^+$23], who answered a slightly weaker version of this question, namely for Hamilton paths instead of cycles, and those can be computed with a simple greedy algorithm.

9.6. **Vertex colorings.** Choo and MacGillivray [CM11] consider the flip graph $G(H, k)$ of k-colorings of a fixed graph H, where two colorings are adjacent if they differ in the color of a single vertex of H. They showed that for every H there is a minimum integer $k_0(H)$, such that for all $k \geq k_0(H)$, the flip graph $G(H, k)$ admits a Hamilton cycle. They also proved that $k_0(K_n) = n+1$ for $n \geq 2$, $k_0(C_n) = 4$ for $n \geq 3$, and for any tree H we have $k_0(H) = 3$, unless H is a star with an odd number of vertices, in which case $k_0(H) = 4$. Celaya, Choo, MacGillivray and Seyffarth [CCMS16] proved that $k_0(K_{a,b}) = 3$ if a and b are both odd and $k_0(K_{a,b}) = 4$ otherwise. Bard [Bar14] established the upper bound $k_0(K_{a_1,a_2,\ldots,a_r}) \leq 2r$. He also asked whether $G(H, k)$ having a Hamilton cycle implies that $G(H, k+1)$ also has one. Stijn Cambie showed that the answer is 'No', by taking H as $K_{5,5}$ minus a perfect matching (cf. [CvdHJ08]). The resulting graph $G(H, 3)$ has a Hamilton cycle, whereas $G(H, 5)$ is disconnected and hence has none. Cavers and Seyffarth [CS19] showed that any 2-tree H satisfies $k_0(H) \in \{4, 5\}$, and they characterized precisely which 2-trees attain which of these two possible values. A *2-tree* is obtained from a triangle by repeatedly adding a new vertex that is joined to the end vertices of an existing edge. McDonald [McD15] considers a variation that allows recoloring more than one vertex at a time, namely at most ℓ vertices at a time, and he asked for the minimum ℓ for which the resulting flip graph has a Hamilton cycle.

Haas [Haa12] introduced another kind of flip graph for colorings. Specifically, we consider isomorphism classes of k-colorings of H that differ only in permuting the colors. For a fixed ordering π of the vertices of H, the lexicographically smallest coloring is chosen as the representative of each isomorphism class. The flip graph $G(H, \pi, k)$ has those representative colorings as vertices, and an edge between any two that differ in the color of a single vertex of H. She showed that if H is a tree on at least 4 vertices and $k \geq 3$, then there is an ordering π such that $G(H, \pi, k)$ has $\boxed{\text{P57}}$ a Hamilton cycle. She also proved that $G(C_n, \pi, k)$ is connected for all $n \geq 3$, $k \geq 4$ and some π, and she mentions computer experiments suggesting that this flip graph has a Hamilton cycle for

$n > 5$. Haas and MacGillivray [HM18] showed that $G(K_{a,b}, \pi, k)$ has a Hamilton path for all $a, b \geq 2$, $k \geq 3$, and some π. They also report some positive results for graphs H obtained via unions and joins, and some negative results about $H = K_{a_1, a_2, \ldots, a_r}$ for $r \geq 3$, $a_i \geq 2$, $k \geq r + 1$.

Finbow and MacGillivray [FM14] consider the flip graph $G(H, k)$ of vertex partitions of H into at most k independent sets, where two of them are adjacent if they differ by moving a single vertex from one class to another. Note that if H is an independent set of size n (=complement of K_n), then $G(H, k)$ is the flip graph of set partitions with at most k parts under element exchanges. The authors showed that $G(H, n)$ has a Hamilton cycle for every n-vertex graph H except for K_n and $K_n - e$, and that $G(H, k)$ has a Hamilton cycle for any tree H with at least 4 vertices and $k \geq 3$.

9.7. Vertex labelings.

Knor [Kno94] considers all non-isomorphic vertex labelings of an n-vertex graph H with integers $1, \ldots, n$, and two of these vertex-labeled graphs are adjacent if one can remove and reinsert k edges to transform one graph into the other. For example, if $H = P_n$, then there are $n!/2$ non-isomorphic labelings of H, and if $k = 1$ then the flip operation is the well-known Pósa rotation. In general, we write $G(H, k)$ for the corresponding flip graph. It is easy to see that $G(H, k)$ is vertex-transitive, as the neighbors of some vertex labeling of H only depend on H and not on the labeling, which makes this problem an interesting test case for Lovász' conjecture (provided that $G(H, k)$ is connected, which is not always the case). Knor proved that $G(P_n, 1)$ has Hamilton cycle for $n \geq 3$, $G(C_n, 2)$ has a Hamilton cycle for $n \geq 4$, and $G(F_n, 1)$ is Hamilton-connected for $n \geq 7$, where F_n is the *fork*, i.e., the n-vertex graph obtained by appending two vertices to the same end vertex of the path P_{n-2}. He further showed that $G(K_{a,b}, a + b - 2)$ for $a \neq b$ has the Johnson graph $J(a + b, a)$ as a spanning subgraph, so it is Hamilton-connected by the results discussed in Section 4.1 (for $a = b$ an analogous statement holds for $J(2a - 1, a)$). Another interesting special case is $G(K_{a+1,a}, a)$, which is isomorphic to the odd graph $K(2a + 1, a) = O_a$.

9.8. Dominating sets.

Adaricheva et al. [ABC$^+$21] consider the flip graph $G(H)$ of dominating sets of H, where two dominating sets are connected by an edge if they differ by adding or removing a single vertex of H from the set. As the number of dominating sets of any graph H is odd [BCS09], the bipartite graph $G(H)$ has two partition classes of different sizes and therefore no Hamilton cycle, so we only need to to focus on finding Hamilton paths. Note that every nonempty subset of vertices of K_n is a dominating set, so $G(K_n)$ is isomorphic to the n-cube Q_n minus a single vertex. Similarly, $G(K_{1,n})$ is isomorphic to Q_n plus a pending edge attached to one vertex. As Q_n has a Hamilton cycle, it follows immediately that $G(K_n)$ and $G(K_{1,n})$ have a Hamilton path. The aforementioned paper shows moreover that $G(P_n)$ has a Hamilton path, and $G(K_{a,b})$ has a Hamilton path if and only if a or b is odd (if both are even there are balancedness problems).

Adaricheva et al. [ABB$^+$22] proved that $G(H)$ has a Hamilton path for any tree H. They also consider the case of cycles $H = C_n$. Note that characteristic vectors of dominating sets of C_n are precisely bitstrings of length n that cyclically avoid the factor 000 (three consecutive vertices on C_n that are not in the dominating set leave the middle vertex undominated), i.e., they are complements of Lucas words. Therefore, by the results of Baril and Vajnovszki [BV05] discussed in Section 3.16, there is a Hamilton path in $G(C_n)$ if and only if n is not a multiple of 4. Similarly, characteristic vectors of dominating sets of P_n are precisely bitstrings of length n that avoid the factor 000 and also 00 at the first or last two positions, which are a proper subset of complemented Fibonacci words.

Haas and Seyffarth [HS14] asked whether the flip graph $G(H, k)$ of dominating sets of size at most k in H has a Hamilton path or cycle. More results and open problems about flip graphs on colorings and dominating sets of graphs are surveyed by Mynhardt and Nasserasr [MN19].

P58

9.9. **Degree sequences and 0/1-matrices.** Arikati and Peled [AP99] consider degree sequences of graphs, which are integer sequences $d = (d_1, \ldots, d_n)$ for which there exists a simple graph with vertex set $[n]$ such that the degree of the vertex i is d_i for all $i = 1, \ldots, n$. The flip graph $G(d)$ has as vertices all graphs that realize the degree sequence d, and two of them are connected by an edge in $G(d)$ if they differ by removing two edges and adding two edges that together form a 4-cycle on $[n]$, an operation that clearly preserves all degrees. The authors proved that $G(d)$ has a Hamilton cycle provided that d has majorization gap 1 (see the paper for definitions). In general, the question whether $G(d)$ admits a Hamilton path or cycle is open.

$\boxed{\text{P59}}$

Brualdi [Bru80] considers $m \times n$ matrices of 0s and 1s, with fixed row sum vector $r = (r_1, \ldots, r_m)$ and column sum vector $s = (s_1, \ldots, s_n)$, where $0 < r_i < n$ for $i = 1, \ldots, m$ and $0 < s_j < m$ for $j = 1, \ldots, n$. The flip graph $G(r, s)$ has as vertices all 0/1-matrices whose row and column sums are r and s, respectively, and two of them are connected by an edge in $G(r, s)$ if the matrices agree in all but four entries that form submatrices $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, respectively. Brualdi raised the question whether the flip graph $G(r, s)$ admits a Hamilton path or cycle. Li and Zhang [LZ94] showed that the answer is 'yes' if $r = (1, \ldots, 1)$ or $s = (1, \ldots, 1)$, proving more strongly that $G(r, s)$ is edge-Hamiltonian. By taking complements, the same result follows if $r = (n-1, \ldots, n-1)$ or $s = (m-1, \ldots, m-1)$.

$\boxed{\text{P60}}$

In the aforementioned paper [AP99], Arkati and Peled point out that Brualdi's problem can be rephrased equivalently using degree sequences as follows: We can interpret an $m \times n$ matrix with row and column sums r and s, respectively, as the biadjacency matrix of a bipartite graph with partition classes of sizes m and n. We then turn the bipartite graph into a split graph by adding all edges in one of the partition classes, making this partition class a clique. Note that the degree sequence d of the resulting split graph yields a flip graph $G(d)$ which is isomorphic to $G(r, s)$. This is because in a split graph, every flip operation removes and adds two edges that go between the independent set and the clique. Therefore, Brualdi's question is equivalent to asking whether the flip graph $G(d)$ for the degree sequence d of any split graph admits a Hamilton path or cycle.

10. GREEDY GRAY CODES

Throughout this survey, we encountered a number of versatile methods for constructing Gray code listings for several different combinatorial objects, by a suitable general encoding of the objects. These include flip-swap languages (Section 3.18), bubble languages (Section 4.2), pattern-avoiding permutations (Section 5.4), linear extensions of posets (Section 5.5), elimination trees (Section 9.4) and acyclic orientations (Section 9.5).

Williams [Wil13] discovered another powerful general paradigm for constructing Gray codes, which he dubbed the *greedy approach*. The idea is to describe the listing implicitly by an algorithm, in which the next object in the list is defined by applying a flip operation to the preceding object, and this operation is selected greedily from a set of applicable flip operations so as to generate an object which has not been encountered in the list before. Williams showed in his paper that many classical Gray codes can be described by such a greedy rule:

- the BRGC for bitstrings (Figure 2 (a)): flip the rightmost possible bit;
- the homogeneous transposition Gray code for combinations due to Eades and McKay (Figure 5 (b)): transpose the rightmost possible 1 homogeneously with the rightmost possible 0;
- the SJT algorithm for permutations (Figure 9 (d)): transpose the largest possible entry with its neighbor;
- the Ord-Smith/Zaks ordering of permutations (Figure 9 (k)): reverse the shortest possible prefix;

- the Lucas, Roelants van Baronaigien and Ruskey Gray code for binary trees (Figure 12): rotate the edge with the largest possible inorder label.

Besides describing known Gray codes in a unified way, the greedy approach has proven to be very useful for discovering new Gray codes. One major application of the greedy method are the Gray codes for zigzag languages of permutations introduced by Hartung, Hoang, Mütze and Williams [HHMW22], which include many classes of pattern-avoiding permutations (recall Section 5.4), and which encode several classes of rectangulations [MM23a] (recall Section 8.4), elimination trees of chordal graphs [CMM22] (recall Section 9.4), and acyclic orientations of chordal graphs (recall Section 9.5).

Another major application area where the greedy method thrives are 0/1-polytopes. As was shown by Merino and Mütze [MM23b], the greedy approach can be used to compute a Hamilton path on the skeleton of any 0/1-polytope (recall Section 2.7). This vastly extends the method to all families of combinatorial objects that can be naturally encoded by 0/1-strings, including spanning trees of graphs (recall Section 9.1 and [MMW22]), matchings of graphs (recall Section 9.2), etc.

The Gray codes derived from the greedy approach all have an underlying hierarchical generation tree, instances of which can be found in several predecessor papers, including Hurtado and Noy's tree of triangulations [HN99], Takaoka's twisted lexico tree [Tak99b], the production trees of the ECO method [BBGP04], and in Li and Sawada's reflectable languages [LS09] (see also [XCU10]).

The key to understanding the success of the greedy method was recently revealed in [MM23b], by exhibiting an underlying connection to Gray code listings that are genlex. Specifically, if there is a genlex listing of combinatorial objects using a particular flip operation, where objects with the same prefix (or suffix) appear consecutively, then the greedy rule to perform a flip that modifies the shortest possible suffix (or prefix, respectively) can generate this listing, when using a suitable tie-breaking rule. This explains why the greedy approach works in all the aforementioned cases. More importantly, it extends the applicability of the greedy method to any of the other genlex Gray codes mentioned in this survey.

Acknowledgements

References

[AAHV07]   O. Aichholzer, F. Aurenhammer, C. Huemer, and B. Vogtenhuber. Gray code enumeration of plane straight-line graphs. *Graphs Combin.*, 23(5):467–479, 2007.

[ABB+22]   K. Adaricheva, H. S. Blake, C. Bozeman, N. E. Clarke, R. Haas, M.-E. Messinger, and K. Seyffarth. Hamilton paths in dominating graphs of trees and cycles. *Graphs Combin.*, 38(6):Paper No. 174, 9 pp., 2022.

[ABC+21]   K. Adaricheva, C. Bozeman, N. E. Clarke, R. Haas, M.-E. Messinger, K. Seyffarth, and H. C. Smith. Reconfiguration graphs for dominating sets. In D. Ferrero, L. Hogben, S. R. Kingan, and G. L. Matthews, editors, *Research Trends in Graph Theory and Applications*, pages 119–135. Springer International Publishing, Cham, 2021.

[ABP06]   E. Ackerman, G. Barequet, and R. Y. Pinter. On the number of rectangulations of a planar point set. *J. Combin. Theory Ser. A*, 113(6):1072–1091, 2006.

[ABS90]   B. Alspach, J.-C. Bermond, and D. Sotteau. Decomposition into cycles. I. Hamilton decompositions. In *Cycles and rays (Montreal, PQ, 1987)*, volume 301 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 9–18. Kluwer Acad. Publ., Dordrecht, 1990.

[Ádá68]   A. Ádám. *Truth functions and the problem of their realization by two-terminal graphs*. Akadémiai Kiadó, Budapest, 1968.

[AF96]   D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65(1-3):21–46, 1996. First International Colloquium on Graphs and Optimization (GOI), 1992 (Grimentz).

[AGRT18]   S. Alstrup, A. Georgakopoulos, E. Rotenberg, and C. Thomassen. A Hamiltonian cycle in the square of a 2-connected graph in linear time. In *Proceedings of the Twenty-ninth ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*, pages 1645–1649. SIAM, Philadelphia, PA, 2018.

[Aig73]   M. Aigner. Lexicographic matching in Boolean algebras. *J. Combin. Theory Ser. B*, 14:187–194, 1973.

[AK88]   H. L. Abbott and M. Katchalski. On the snake in the box problem. *J. Combin. Theory Ser. B*, 45(1):13–24, 1988.

[AK91]   H. L. Abbott and M. Katchalski. On the construction of snake in the box codes. *Utilitas Math.*, 40:97–116, 1991.

[AL89]   B. Alspach and G. Liu. Paths and cycles in matroid base graphs. *Graphs Combin.*, 5(3):207–211, 1989.

[Als79]   B. Alspach. Hamiltonian cycles in vertex-transitive graphs of order $2p$. *Congr. Numer.*, XXIII–XIV:131–139, 1979. Proceedings of the Tenth Southeastern Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1979).

[Als81]   B. Alspach. The search for long paths and cycles in vertex-transitive graphs and digraphs. In *Combinatorial mathematics, VIII (Geelong, 1980)*, volume 884 of *Lecture Notes in Math.*, pages 14–22. Springer, Berlin-New York, 1981.

[Als13]   B. Alspach. Johnson graphs are Hamilton-connected. *Ars Math. Contemp.*, 6(1):21–23, 2013.

[ALSO04]   E. Agrell, J. Lassing, E. G. Strom, and T. Ottosson. On the optimality of the binary reflected Gray code. *IEEE Transactions on Information Theory*, 50(12):3170–3182, 2004.

[AP99]   S. R. Arikati and U. N. Peled. The realization graph of a degree sequence with majorization gap 1 is Hamiltonian. *Linear Algebra Appl.*, 290(1–3):213–235, 1999.

[Arn11]   J. Arndt. *Matters computational. Ideas, algorithms, source code*. Springer, 2011.

[AS82]   J. Aubert and B. Schneider. Décomposition de la somme cartésienne d'un cycle et de l'union de deux cycles hamiltoniens en cycles hamiltoniens. *Discrete Math.*, 38(1):7–16, 1982.

[BA08]   M. Bras-Amorós. Fibonacci-like behavior of the number of numerical semigroups of a given genus. *Semigroup Forum*, 76(2):379–384, 2008.

[Bab95]   L. Babai. Automorphism groups, isomorphism, reconstruction. In *Handbook of combinatorics, Vol. 2*, pages 1447–1540. Elsevier Sci. B. V., Amsterdam, 1995.

[Bar03]   J.-L. Baril. Gray code for Cayley permutations. *Comput. Sci. J. Moldova*, 11(2):124–136, 2003.

[Bar07]   J.-L. Baril. Gray code for permutations with a fixed number of cycles. *Discrete Math.*, 307(13):1559–1571, 2007.

[Bar08]   J.-L. Baril. Efficient generating algorithm for permutations with a fixed number of excedances. *Pure Math. Appl. (PU.M.A.)*, 19(2–3):61–69, 2008.

[Bar09]   J.-L. Baril. More restrictive Gray codes for some classes of pattern avoiding permutations. *Inform. Process. Lett.*, 109(14):799–804, 2009.

[Bar13]   J.-L. Baril. Gray code for permutations with a fixed number of left-to-right minima. *Ars Combin.*, 111:225–239, 2013.

[Bar14]   S. Bard. Gray code numbers of complete multipartite graphs. Master's thesis, University of Victoria, 2014.

[Bar23]   J. Barron. Gray codes for numerical semigroups. Master's thesis, University of Warwick, 2023.

[BB95]   B. Bose and B. Broeg. Lee distance Gray codes. In *Proceedings of 1995 IEEE International Symposium on Information Theory*, page 236, 1995.

[BBGP04]   S. Bacchelli, E. Barcucci, E. Grazzini, and E. Pergola. Exhaustive generation of combinatorial objects by ECO. *Acta Inform.*, 40(8):585–602, 2004.

[BBKA95]   B. Bose, B. Broeg, Y. Kwon, and Y. Ashir. Lee distance and topological properties of k-ary n-cubes. *IEEE Trans. Comput.*, 44(8):1021–1030, 1995.

[BBP+15]   A. Bernini, S. Bilotta, R. Pinzani, A. Sabri, and V. Vajnovszki. Gray code orders for q-ary words avoiding a given factor. *Acta Inform.*, 52(7–8):573–592, 2015.

[BCS09]   A. E. Brouwer, P. Csorba, and A. Schrijver. The number of dominating sets of a finite graph is odd. https://www.win.tue.nl/~aeb/preprints/domin4a.pdf, 2009.

[BER76]   J. R. Bitner, G. Ehrlich, and E. M. Reingold. Efficient generation of the binary reflected Gray code and its applications. *Comm. ACM*, 19(9):517–521, 1976.

[Ber78]   J.-C. Bermond. Hamiltonian graphs. In L. W. Beineke and R. J. Wilson, editors, *Selected topics in graph theory*, pages 127–167. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], London-New York, 1978.

[BH09a]   C. Biró and D. M. Howard. The first three levels of an order preserving Hamiltonian path in the subset lattice. *Order*, 26(2):101–107, 2009.

[BH09b]   P. Bose and F. Hurtado. Flips in planar graphs. *Comput. Geom.*, 42(1):60–80, 2009.

[BI76]   J. A. Bondy and A. W. Ingleton. Pancyclic graphs. II. *J. Combin. Theory Ser. B*, 20(1):41–46, 1976.

[Big79]   N. Biggs. Some odd graph theory. In *Second International Conference on Combinatorial Mathematics (New York, 1978)*, volume 319 of *Ann. New York Acad. Sci.*, pages 71–81. New York Acad. Sci., New York, 1979.

[BKV22]   J.-L. Baril, S. Kirgizov, and V. Vajnovszki. Gray codes for Fibonacci q-decreasing words. *Theoret. Comput. Sci.*, 927:120–132, 2022.

[BMdS09]   J.-L. Baril and C. Moreira dos Santos. Gray code for compositions of n with parts 1 and p. *Adv. Appl. Discret. Math.*, 3(1):67–84, 2009.

[BR96]   B. Bultena and F. Ruskey. Transition restricted Gray codes. *Electron. J. Combin.*, 3(1):Paper 11, 11 pp., 1996.

[BR98]   B. Bultena and F. Ruskey. An Eades-McKay algorithm for well-formed parentheses strings. *Inform. Process. Lett.*, 68(5):255–259, 1998.

[Bre73]   L. M. Bregman. Certain properties of nonnegative matrices and their permanents. *Dokl. Akad. Nauk SSSR*, 211:27–30, 1973.

[Bru80]   R. A. Brualdi. Matrices of zeros and ones with fixed row and column sum vectors. *Linear Algebra Appl.*, 33:159–231, 1980.

[BS96]   G. S. Bhat and C. D. Savage. Balanced Gray codes. *Electron. J. Combin.*, 3(1):Paper 25, 11 pp., 1996.

[BS03]   D. W. Bass and I. H. Sudborough. Pancake problems with restricted prefix reversals and some corresponding Cayley networks. *J. Parallel Distributed Comput.*, 63(3):327–336, 2003.

[BV04]   J.-L. Baril and V. Vajnovszki. Gray code for derangements. *Discrete Appl. Math.*, 140(1–3):207–221, 2004.

[BV05]   J.-L. Baril and V. Vajnovszki. Minimal change list for Lucas strings and some graph theoretic consequences. *Theoret. Comput. Sci.*, 346(2–3):189–199, 2005.

[BW84]   M. Buck and D. Wiedemann. Gray codes with restricted density. *Discrete Math.*, 48(2–3):163–171, 1984.

[Cav75]   S. R. Cavior. An upper bound associated with errors in Gray code. *IEEE Trans. Information Theory*, IT-21(5):596, 1975.

[CCMS16]   M. Celaya, K. Choo, G. MacGillivray, and K. Seyffarth. Reconfiguring k-colourings of complete bipartite graphs. *Kyungpook Math. J.*, 56(3):647–655, 2016.

[CD06]   M. P. Carr and S. L. Devadoss. Coxeter complexes and graph-associahedra. *Topology Appl.*, 153(12):2155–2168, 2006.

[CDG92]   F. Chung, P. Diaconis, and R. Graham. Universal cycles for combinatorial structures. *Discrete Math.*, 110(1–3):43–59, 1992.

[CEY07]   G. J. Chang, S.-P. Eu, and C.-H. Yeh. On the (n, t)-antipodal Gray codes. *Theoret. Comput. Sci.*, 374(1–3):82–90, 2007.

[CG96]   S. J. Curran and J. A. Gallian. Hamiltonian cycles and paths in Cayley graphs and digraphs—a survey. *Discrete Math.*, 156(1–3):1–18, 1996.

[CGP97]   L. H. Clark, J. C. George, and T. D. Porter. On the number of 1-factors in the n-cube. *Congr. Numer.*, 127:67–69, 1997. Proceedings of the Twenty-eighth Southeastern International Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1997).

[CGS21]   B. Cameron, A. Grubb, and J. Sawada. A pivot Gray code listing for the spanning trees of the fan graph. In C.-Y. Chen, W.-K. Hon, L.-J. Hung, and C.-W. Lee, editors, *Proceedings of the 27th International Computing and Combinatorics Conference (COCOON 2021), Tainan, Taiwan, October 24–26*, volume 13025 of *Lecture Notes in Comput. Sci.*, pages 49–60. Springer, 2021.

[Cha70a]   P. J. Chase. Algorithm 382: Combinations of M out of N objects [G6]. *Commun. ACM*, 13(6):368, 1970.

[Cha70b]   P. J. Chase. Algorithm 383: Permutations of a set with repetitions [G6]. *Commun. ACM*, 13(6):368–369, 1970.

[Cha89]   P. J. Chase. Combination generation and graylex ordering. *Congr. Numer.*, 69:215–242, 1989. Proceedings of the Eighteenth Manitoba Conference on Numerical Mathematics and Computing (Winnipeg, MB, 1988).

[Che85]   C.-S. Cheng. Run orders of factorial designs. In L. M. LeCam and R. A. Olshen, editors, *Proc. of the Berkeley Conf. in Honor of J. Neyman and J. Kiefer, Vol. II*, pages 619–633. Wadsworth, Inc., 1985.

[Che98]   Y. Q. Chen. On Hamiltonicity of vertex-transitive graphs and digraphs of order p^4. *J. Combin. Theory Ser. B*, 72(1):110–121, 1998.

[CHJ+74]   G. Chartrand, A. M. Hobbs, H. A. Jung, S. F. Kapoor, and C. St. J. A. Nash-Williams. The square of a block is Hamiltonian connected. *J. Combin. Theory Ser. B*, 16:290–292, 1974.

[CHM14]   D. Christofides, J. Hladký, and A. Máthé. Hamilton cycles in dense vertex-transitive graphs. *J. Combin. Theory Ser. B*, 109:34–72, 2014.

[CHM+23]   J. Cardinal, H. P. Hoang, A. Merino, O. Mička, and T. Mütze. Combinatorial generation via permutation languages. V. Acyclic orientations. *SIAM J. Discrete Math.*, 37(3):1509–1547, 2023.

[Chv75]   V. Chvátal. On certain polytopes associated with graphs. *J. Combinatorial Theory Ser. B*, 18:138–154, 1975.

[CIRS03]   L. S. Chandran, L. Ibarra, F. Ruskey, and J. Sawada. Generating and characterizing the perfect elimination orderings of a chordal graph. In *Random generation of combinatorial objects and bijective combinatorics*, volume 307, pages 303–317, 2003.

[CL87]   B. L. Chen and K.-W. Lih. Hamiltonian uniform subset graphs. *J. Combin. Theory Ser. B*, 42(3):257–263, 1987.

[CL91]   M. Y. Chan and S.-J. Lee. On the existence of Hamiltonian circuits in faulty hypercubes. *SIAM J. Discrete Math.*, 4(4):511–527, 1991.

[CM11]   K. Choo and G. MacGillivray. Gray code numbers for graphs. *Ars Math. Contemp.*, 4(1):125–139, 2011.

[CM15]   A. Conflitti and R. Mamede. Gray codes and lexicographical combinatorial generation for nonnesting and sparse nonnesting set partitions. *Theoret. Comput. Sci.*, 592:87–96, 2015.

[CM17]   A. Conflitti and R. Mamede. Gray codes for noncrossing and nonnesting partitions of classical types. *Internat. J. Algebra Comput.*, 27(5):455–475, 2017.

[CMM22]   J. Cardinal, A. Merino, and T. Mütze. Efficient generation of elimination trees and graph associahedra. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2128–2140. [Society for Industrial and Applied Mathematics (SIAM)], Philadelphia, PA, 2022.

[CNDS16]   M. Cooke, C. North, M. Dewar, and B. Stevens. A note on Beckett-Gray codes and the relationship of Gray codes to data structures. *arXiv:1608.06001*, 2016.

[Coh63]   M. Cohn. Affine m-ary Gray codes. *Information and Control*, 6:70–78, 1963.

[Con99]   M. Conder. Explicit definition of the binary reflected Gray codes. *Discrete Math.*, 195(1–3):245–249, 1999.

[Cor92]   P. F. Corbett. Rotator graphs: An efficient topology for point-to-point multiprocessor networks. *IEEE Trans. Parallel Distributed Syst.*, 3(5):622–626, 1992.

[cos]   The Combinatorial Object Server. `http://combos.org`.

[CQ81]   C. C. Chen and N. F. Quimpo. On strongly Hamiltonian abelian group graphs. In *Combinatorial mathematics, VIII (Geelong, 1980)*, volume 884 of *Lecture Notes in Math.*, pages 23–34. Springer, Berlin-New York, 1981.

[CS19]   M. Cavers and K. Seyffarth. Reconfiguring vertex colourings of 2-trees. *Ars Math. Contemp.*, 17(2):653–698, 2019.

[CSW89]   J. H. Conway, N. J. A. Sloane, and A. R. Wilks. Gray codes for reflection groups. *Graphs Combin.*, 5(4):315–325, 1989.

[CSW99]   T. Chinburg, C. D. Savage, and H. S. Wilf. Combinatorial families that are exponentially far from being listable in Gray code sequence. *Trans. Amer. Math. Soc.*, 351(1):379–402, 1999.

[CSW21]   B. Cameron, J. Sawada, and A. Williams. A Hamilton cycle in the k-sided pancake network. In *Combinatorial algorithms*, volume 12757 of *Lecture Notes in Comput. Sci.*, pages 137–151. Springer, Cham, [2021] ©2021.

[Cum66]   R. L. Cummins. Hamilton circuits in tree graphs. *IEEE Trans. Circuit Theory*, CT-13:82–90, 1966.

[CvdHJ08]   L. Cereceda, J. van den Heuvel, and M. Johnson. Connectedness of the graph of vertex-colourings. *Discrete Math.*, 308(5–6):913–919, 2008.

[CW93]   R. C. Compton and S. G. Williamson. Doubly adjacent Gray codes for the symmetric group. *Linear and Multilinear Algebra*, 35(3–4):237–293, 1993.

[CW95]   E. R. Canfield and S. G. Williamson. A loop-free algorithm for generating the linear extensions of a poset. *Order*, 12(1):57–75, 1995.

[Dav65]   D. W. Davies. Longest "separated" paths and loops in an N cube. *IEEE Trans. Electron. Comput.*, 14(2):261, 1965.

[dB]   De Bruijn sequence and universal cycle constructions. `http://debruijnsequence.org`.

[dBvETK51] N. G. de Bruijn, Ca. van Ebbenhorst Tengbergen, and D. Kruyswijk. On the set of divisors of a number. *Nieuw Arch. Wiskunde (2)*, 23:191–193, 1951.

[DD07]   C. Degni and A. A. Drisko. Gray-ordered binary necklaces. *Electron. J. Combin.*, 14(1):Paper 7, 23 pp., 2007.

[DDGŠ09] D. Dimitrov, T. Dvořák, P. Gregor, and R. Škrekovski. Gray codes avoiding matchings. *Discrete Math. Theor. Comput. Sci.*, 11(2):123–147, 2009.

[DDGŠ13] D. Dimitrov, T. Dvořák, P. Gregor, and R. Škrekovski. Linear time construction of a compressed Gray code. *European J. Combin.*, 34(1):69–81, 2013.

[DEHW23] E. Downing, S. Einstein, E. Hartung, and A. Williams. Catalan squares and staircases: Relayering and repositioning Gray codes. In *Proceedings of the 35th Canadian Conference on Computational Geometry (CCCG 2023)*, pages 273–282. 2023.

[Dei85]   K. Deimer. A new upper bound for the length of snakes. *Combinatorica*, 5(2):109–120, 1985.

[Dev09]   S. L. Devadoss. A realization of graph associahedra. *Discrete Math.*, 309(1):271–276, 2009.

[DFMV08] W. M. B. Dukes, M. F. Flanagan, T. Mansour, and V. Vajnovszki. Combinatorial Gray codes for classes of pattern avoiding permutations. *Theoret. Comput. Sci.*, 396(1–3):35–49, 2008.

[DG07]   T. Dvořák and P. Gregor. Hamiltonian paths with prescribed edges in hypercubes. *Discrete Math.*, 307(16):1982–1998, 2007.

[Dic74]   A. W. Dickinson. Some run orders requiring a minimum number of factor level changes for the 2^4 and 2^5 main effect plans. *Technometrics*, 16(1):31–37, 1974.

[DKM21]   S. Du, K. Kutnar, and D. Marušič. Resolving the Hamiltonian problem for vertex-transitive graphs of order a product of two primes. *Combinatorica*, 41(4):507–543, 2021.

[DLM+12] S. Durocher, P. C. Li, D. Mondal, F. Ruskey, and A. Williams. Cool-lex order and k-ary Catalan structures. *J. Discrete Algorithms*, 16:287–307, 2012.

[Dou69]   R. J. Douglas. Upper bounds on the length of circuits of even spread in the d-cube. *J. Combin. Theory*, 7:206–214, 1969.

[Dvo05]   T. Dvořák. Hamiltonian cycles with prescribed edges in hypercubes. *SIAM J. Discrete Math.*, 19(1):135–144, 2005.

[EHH01]   M. El-Hashash and A. Hassan. On the Hamiltonicity of two subgraphs of the hypercube. *Congr. Numer.*, 148:7–32, 2001. Proceedings of the Thirty-second Southeastern International Conference on Combinatorics, Graph Theory and Computing (Baton Rouge, LA, 2001).

[Ehr73a] G. Ehrlich. Algorithm 466: Four combinatorial algorithms [G6]. *Commun. ACM*, 16(11):690–691, 1973.

[Ehr73b] G. Ehrlich. Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *J. Assoc. Comput. Mach.*, 20:500–513, 1973.

[EHR84]   P. Eades, M. Hickey, and R. C. Read. Some Hamilton paths and a minimal change algorithm. *J. Assoc. Comput. Mach.*, 31(1):19–29, 1984.

[EM84]   P. Eades and B. McKay. An algorithm for generating subsets of fixed size with a strong minimal change property. *Inform. Process. Lett.*, 19(3):131–133, 1984.

[Enn93]   T. C. Enns. Hamiltonian circuits and paths in subset graphs with circular adjacency. *Discrete Math.*, 122(1–3):153–165, 1993.

[EP96]   T. Etzion and K. G. Paterson. Near optimal single-track Gray codes. *IEEE Trans. Inform. Theory*, 42(3):779–789, 1996.

[Er84]   M. C. Er. On generating the N-ary reflected Gray codes. *IEEE Transactions on Computers*, C-33(8):739–741, 1984.

[Etz07]   T. Etzion. Single-track Gray codes and sequences. In *Sequences, subsequences, and consequences*, volume 4893 of *Lecture Notes in Comput. Sci.*, pages 129–133. Springer, Berlin, 2007.

[Evd69]   A. A. Evdokimov. Maximal length of circuit in a unitary n-dimensional cube. *Mathematical notes of the Academy of Sciences of the USSR*, 6(3):642–648, 1969. Original Russian text in *Mat. Zametki*, 6:309–319, 1969.

[Evd80]   A. A. Evdokimov. Numeration of subsets of a finite set. *Metody Diskret. Analiz.*, (34):8–26, 1980.

[Evd14]   A. A. Evdokimov. Circuit codes and the snake-in-the-box problem. *Uch. Zap. Kazan. Univ. Ser. Fiz.-Mat. Nauki*, 156(3):55–65, 2014.

[Eve73]   S. Even. *Algorithmic combinatorics*. The Macmillan Company, New York; Collier-Macmillan Publishers, London, 1973.

[FB07]   M. Flahive and B. Bose. Balancing cyclic R-ary Gray codes. *Electron. J. Combin.*, 14(1):Paper 31, 14 pp., 2007.

[Feu02]   W. Feussner. Ueber Stromverzweigung in netzförmigen Leitern. *Ann. Phys.*, 314(13):1304–1329, 1902.

[FG12]   J Fink and P Gregor. Long cycles in hypercubes with optimal number of faulty vertices. *J. Comb. Optim.*, 24(3):240–265, 2012.

[FHŠ93]   H. Fleischner, P. Horák, and J. Širáň. Generating Hamiltonian cycles in complete graphs. *Acta Math. Univ. Comenian. (N.S.)*, 62(2):155–159, 1993.

[Fin07]   J. Fink. Perfect matchings extend to Hamilton cycles in hypercubes. *J. Combin. Theory Ser. B*, 97(6):1074–1076, 2007.

[Fin19]   J. Fink. Matchings extend into 2-factors in hypercubes. *Combinatorica*, 39(1):77–84, 2019.

[FK11]   S. Felsner and K. Knauer. Distributive lattices, polyhedra, and generalized flows. *European J. Combin.*, 32(1):45–59, 2011.

[FKMS20]   S. Felsner, L. Kleist, T. Mütze, and L. Sering. Rainbow cycles in flip graphs. *SIAM J. Discrete Math.*, 34(1):1–39, 2020.

[FL13]   H. Fan and G. Liu. Properties of Hamilton cycles of circuit graphs of matroids. *Front. Math. China*, 8(4):801–809, 2013.

[Fla08]   M. Flahive. Balancing cyclic R-ary Gray codes. II. *Electron. J. Combin.*, 15(1):Paper 128, 16 pp., 2008.

[Fle74]   H. Fleischner. The square of every two-connected graph is Hamiltonian. *J. Combin. Theory Ser. B*, 16:29–34, 1974.

[Flo56]   I. Flores. Reflected number systems. *IRE Trans. Electron. Comput.*, 5(2):79–82, 1956.

[FM96]   R. Feldmann and P. Mysliwietz. The shuffle exchange network has a Hamiltonian path. *Math. Systems Theory*, 29(5):471–485, 1996.

[FM14]   S. Finbow and G. MacGillivray. Hamiltonicity of Bell and Stirling colour graphs. Unpublished manuscript, 2014.

[Fre78]   M. L. Fredman. Observations on the complexity of generating quasi-Gray codes. *SIAM J. Comput.*, 7(2):134–146, 1978.

[Fre82]   H. Fredricksen. A survey of full length nonlinear shift register cycle algorithms. *SIAM Rev.*, 24(2):195–221, 1982.

[FS09]   T. Feder and C. Subi. Nearly tight bounds on the number of Hamiltonian circuits of the hypercube and generalizations. *Inform. Process. Lett.*, 109(5):267–272, 2009.

[FT95]   S. Felsner and W. T. Trotter. Colorings of diagrams of interval orders and α-sequences of sets. *Discrete Math.*, 144(1–3):23–31, 1995. Combinatorics of ordered sets (Oberwolfach, 1991).

[GG03]   L. Goddyn and P. Gvozdjak. Binary Gray codes with long bit runs. *Electron. J. Combin.*, 10:Paper 27, 10 pp., 2003.

[Gil58]   E. N. Gilbert. Gray codes and paths on the n-cube. *Bell System Tech. J.*, 37:815–826, 1958.

[GJM+22]   P. Gregor, S. Jäger, T. Mütze, J. Sawada, and K. Wille. Gray codes and symmetric chains. *J. Combin. Theory Ser. B*, 153:31–60, 2022.

[GK76]   C. Greene and D. J. Kleitman. Strong versions of Sperner's theorem. *J. Combin. Theory Ser. A*, 20(1):80–88, 1976.

[GKMM12]   H. H. Glover, K. Kutnar, A. Malnič, and D. Marušič. Hamilton cycles in $(2, \text{odd}, 3)$-Cayley graphs. *Proc. Lond. Math. Soc. (3)*, 104(6):1171–1197, 2012.

[GLN88]   L. Goddyn, G. M. Lawrence, and E. Nemeth. Gray codes with optimized run lengths. *Utilitas Math.*, 34:179–192, 1988.

[GM18]   P. Gregor and T. Mütze. Trimming and gluing Gray codes. *Theor. Comp. Sci.*, 714:74–95, 2018.

[GMM23a]   P. Gregor, A. Merino, and T. Mütze. Star transposition Gray codes for multiset permutations. *J. Graph Theory*, 103(2):212–270, 2023.

[GMM23b]   P. Gregor, O. Mička, and T. Mütze. On the central levels problem. *J. Combin. Theory Ser. B*, 160:163–205, 2023.

[GMM24]   P. Gregor, A. Merino, and T. Mütze. The Hamilton compression of highly symmetric graphs. *Ann. Comb.*, 28(2):379–437, 2024.

[GMN18]   P. Gregor, T. Mütze, and J. Nummenpalo. A short proof of the middle levels theorem. *Discrete Anal.*, pages Paper No. 8, 12 pp., 2018.

[GMS18]   G. Gutierres, R. Mamede, and J. L. Santos. Gray codes for signed involutions. *Discrete Math.*, 341(9):2590–2601, 2018.

[GMS19]   G. Gutierres, R. Mamede, and J. L. Santos. Optimal Gray code for involutions. *Inform. Process. Lett.*, 148:19–22, 2019.

[Gou91]   R. J. Gould. Updating the Hamiltonian problem—a survey. *J. Graph Theory*, 15(2):121–157, 1991.

[GP85]   M. Guan and W. Pulleyblank. Eulerian orientations and circulations. *SIAM J. Algebraic Discrete Methods*, 6(4):657–664, 1985.

[GR87]   R. J. Gould and R. Roth. Cayley digraphs and $(1, j, n)$-sequencings of the alternating groups A_n. *Discrete Math.*, 66(1–2):91–102, 1987.

[Gra53]   F. Gray. Pulse code communication, 1953. March 17, 1953 (filed Nov. 1947). U.S. Patent 2,632,058.

[Gre09]   P. Gregor. Perfect matchings extending on subcubes to Hamiltonian cycles of hypercubes. *Discrete Math.*, 309(6):1711–1713, 2009.

[GŠ10]   P. Gregor and R. Škrekovski. On generalized middle-level problem. *Inform. Sci.*, 180(12):2448–2457, 2010.

[Haa12]   R. Haas. The canonical coloring graph of trees and cycles. *Ars Math. Contemp.*, 5(1):149–157, 2012.

[Hav83]   I. Havel. Semipaths in directed cubes. In *Graphs and other combinatorial topics (Prague, 1982)*, volume 59 of *Teubner-Texte Math.*, pages 101–108. Teubner, Leipzig, 1983.

[Hea63]   B. R. Heap. Permutations by interchanges. *Comput. J.*, 6(3):293–298, 1963.

[HH72]   C. A. Holzmann and F. Harary. On the tree graph of a matroid. *SIAM J. Appl. Math.*, 22:187–193, 1972.

[HHM+99]   M. C. Hernando, F. Hurtado, A. Márquez, M. Mora, and M. Noy. Geometric tree graphs of points in convex position. *Discrete Appl. Math.*, 93(1):51–66, 1999. Proceedings of 13th European Workshop on Computational Geometry CG'97 (Würzburg, 1997).

[HHMW22]   E. Hartung, H. P. Hoang, T. Mütze, and A. Williams. Combinatorial generation via permutation languages. I. Fundamentals. *Trans. Amer. Math. Soc.*, 375(4):2255–2291, 2022.

[HHN02]   C. Hernando, F. Hurtado, and M. Noy. Graphs of non-crossing perfect matchings. *Graphs Combin.*, 18(3):517–532, 2002.

[HHNOP09]   C. Huemer, F. Hurtado, M. Noy, and E. Omaña-Pulido. Gray codes for non-crossing partitions and dissections of a convex polygon. *Discrete Appl. Math.*, 157(7):1509–1520, 2009.

[HHNRC05]   M. E. Houle, F. Hurtado, M. Noy, and E. Rivera-Campo. Graphs of triangulations and perfect matchings. *Graphs Combin.*, 21(3):325–331, 2005.

[HHP08]   C. Huemer, F. Hurtado, and J. Pfeifle. The rotation graph of k-ary trees is Hamiltonian. *Inform. Process. Lett.*, 109(2):124–129, 2008.

[HHW88]   F. Harary, J. P. Hayes, and H.-J. Wu. A survey of the theory of hypercube graphs. *Comput. Math. Appl.*, 15(4):277–289, 1988.

[HK78]   D. Hausmann and B. Korte. Colouring criteria for adjacency on $0-1$-polyhedra. *Math. Programming Stud.*, (8):106–127, 1978. Polyhedral combinatorics.

[HKMP13]   A. M. Hinz, S. Klavžar, U. Milutinović, and C. Petr. *The tower of Hanoi—myths and maths*. Birkhäuser/Springer Basel AG, Basel, 2013. With a foreword by I. Stewart.

[HM18]   R. Haas and G. MacGillivray. Connectivity and Hamiltonicity of canonical colouring graphs of bipartite and complete multipartite graphs. *Algorithms (Basel)*, 11(4):Paper No. 40, 14 pp., 2018.

[HM21]   H. P. Hoang and T. Mütze. Combinatorial generation via permutation languages. II. Lattice congruences. *Israel J. Math.*, 244(1):359–417, 2021.

[HN99]   F. Hurtado and M. Noy. Graph of triangulations of a convex polygon and tree of triangulations. *Comput. Geom.*, 13(3):179–188, 1999.

[HNS97]   M. Habib, L. Nourine, and G. Steiner. Gray codes for the ideals of interval orders. *J. Algorithms*, 25(1):52–66, 1997.

[HÖ14]   H. Haanpää and P. R. J. Östergård. Counting Hamiltonian cycles in bipartite graphs. *Math. Comp.*, 83(286):979–995, 2014.

[Hol17] A. E. Holroyd. Perfect snake-in-the-box codes for rank modulation. *IEEE Trans. Inform. Theory*, 63(1):104–110, 2017.

[HPB96] A. P. Hiltgen, K. G. Paterson, and M. Brandestini. Single-track Gray codes. *IEEE Trans. Inf. Theory*, 42(5):1555–1561, 1996.

[HR88] T. Hough and F. Ruskey. An efficient implementation of the Eades, Hickey, Read adjacent interchange combination generation algorithm. *J. Combin. Math. Combin. Comput.*, 4:79–85, 1988.

[HR18] F. Herter and G. Rote. Loopless Gray code enumeration and the Tower of Bucharest. *Theoret. Comput. Sci.*, 748:40–54, 2018.

[HS14] R. Haas and K. Seyffarth. The *k*-dominating graph. *Graphs Combin.*, 30(3):609–617, 2014.

[Hsu93] W.-J. Hsu. Fibonacci cubes—a new interconnection technology. *IEEE Trans. Parallel Distributed Syst.*, 4(1):3–12, 1993.

[Hur94] G. Hurlbert. The antipodal layers problem. *Discrete Math.*, 128(1–3):237–245, 1994.

[JM95] T. A. Jenkyns and D. McCarthy. Generating all *k*-subsets of $\{1 \cdots n\}$ with minimal changes. *Ars Combin.*, 40:153–159, 1995.

[Joh63] S. M. Johnson. Generation of permutations by adjacent transposition. *Math. Comp.*, 17:282–285, 1963.

[Joh11] J. R. Johnson. An inductive construction for Hamilton cycles in Kneser graphs. *Electron. J. Combin.*, 18(1):Paper 189, 12 pp., 2011.

[JR94] M. Jiang and F. Ruskey. Determining the Hamilton-connectedness of certain vertex-transitive graphs. *Discrete Math.*, 133(1–3):159–169, 1994.

[JSH09] B. Jackson, B. Stevens, and G. Hurlbert. Research problems on Gray codes and universal cycles. *Discrete Math.*, 309(17):5341–5348, 2009.

[JW80] J. T. Joichi and D. E. White. Gray codes in graphs of subsets. *Discrete Math.*, 31(1):29–41, 1980.

[JWW80] J. T. Joichi, D. E. White, and S. G. Williamson. Combinatorial Gray codes. *SIAM J. Comput.*, 9(1):130–141, 1980.

[JWW20] M. Jacques, D. Wong, and K. Woo. Generating Gray codes for weak orders in constant amortized time. *Discrete Math.*, 343(10):Paper 111992, 10 pp., 2020.

[Kar68] J. J. Karaganis. On the cube of a graph. *Canad. Math. Bull.*, 11:295–296, 1968.

[Kat66] Gy. Katona. On a conjecture of Erdős and a stronger form of Sperner's theorem. *Studia Sci. Math. Hungar.*, 1:59–63, 1966.

[Kau58] W. H. Kautz. Unit-distance error-checking codes. *IRE Trans. Electron. Comput.*, 7(2):179–180, 1958.

[Kay76] R. Kaye. A Gray code for set partitions. *Information Processing Lett.*, 5(6):171–173, 1976.

[Ker21] K. Kerr. Successor rule for a restricted growth string Gray code. Unpublished manuscript, 2021.

[Kin06] A. King. Generating indecomposable permutations. *Discrete Math.*, 306(5):508–518, 2006.

[KL75] V. L. Kompel'maher and V. A. Liskovec. Sequential generation of permutations by means of a transposition basis. *Kibernetika (Kiev)*, (3):17–21, 1975.

[KL02] J. F. Korsh and P. S. LaFollette. Loopless generation of linear extensions of a poset. *Order*, 19(2):115–126, 2002.

[KL04a] J. F. Korsh and P. S. LaFollette. Constant time generation of derangements. *Inform. Process. Lett.*, 90(4):181–186, 2004.

[KL04b] J. F. Korsh and P. S. LaFollette. Loopless array generation of multiset permutations. *Comput. J.*, 47(5):612–621, 2004.

[Kli82] P. Klingsberg. A Gray code for compositions. *J. Algorithms*, 3(1):41–44, 1982.

[KLL11] J. Korsh, P. LaFollette, and S. Lipschutz. A loopless implementation of a Gray code for signed permutations. *Publ. Inst. Math. (Beograd) (N.S.)*, 89(103):37–47, 2011.

[KM08] K. Kutnar and D. Marušič. Hamiltonicity of vertex-transitive graphs of order 4*p*. *European J. Combin.*, 29(2):423–438, 2008.

[KM09] K. Kutnar and D. Marušič. Hamilton cycles and paths in vertex-transitive graphs—current directions. *Discrete Math.*, 309(17):5491–5500, 2009.

[KMZ12] K. Kutnar, D. Marušič, and C. Zhang. Hamilton paths in vertex-transitive graphs of order 10*p*. *European J. Combin.*, 33(6):1043–1077, 2012.

[Kno94] M. Knor. Gray codes in graphs. *Math. Slovaca*, 44(4):395–412, 1994.

[Knu11] D. E. Knuth. *The art of computer programming. Vol. 4A. Combinatorial algorithms. Part 1.* Addison-Wesley, Upper Saddle River, NJ, 2011.

[Knu20]   D. E. Knuth. *The art of computer programming. Vol. 4–Pre-fascicle 8A. Hamiltonian paths and cycles.* 2020. Available at `https://www-cs-faculty.stanford.edu/~knuth/fasc8a.ps.gz`.

[Koc96]   K. J. Kochut. Snake-in-the-box codes for dimension 7. *J. Combin. Math. Combin. Comput.*, 20:175–185, 1996.

[Kor01]   J. F. Korsh. Loopless generation of up-down permutations. *Discrete Math.*, 240(1–3):97–122, 2001.

[KR92]   C. W. Ko and F. Ruskey. Generating permutations of a bag by interchanges. *Inform. Process. Lett.*, 41(5):263–269, 1992.

[KR93]   Y. Koda and F. Ruskey. A Gray code for the ideals of a forest poset. *J. Algorithms*, 15(2):324–340, 1993.

[Kre96]   G. Kreweras. Matchings and Hamiltonian cycles on hypercubes. *Bull. Inst. Combin. Appl.*, 16:87–91, 1996.

[KRL08]   D. Kim, M. Ryu, and S. Lee. Antipodal Gray codes for structured light. In *2008 IEEE International Conference on Robotics and Automation (ICRA 2008), Pasadena, CA, USA, May 19–23*, pages 3016–3021, 2008.

[KS99]   D. L. Kreher and D. R. Stinson. *Combinatorial algorithms: Generation, enumeration, and search.* CRC Press, 1999.

[KS04]   C. E. Killian and C. D. Savage. Antipodal Gray codes. *Discrete Math.*, 281(1–3):221–236, 2004.

[KŠ09]   K. Kutnar and P. Šparl. Hamilton paths and cycles in vertex-transitive graphs of order 6*p*. *Discrete Math.*, 309(17):5444–5460, 2009.

[Leh65]   D. H. Lehmer. Permutation by adjacent interchanges. *Amer. Math. Monthly*, 72(2, part II):36–46, 1965.

[LHC94]   J. Liu, W.-J. Hsu, and M. J. Chung. Generalized Fibonacci cubes are mostly Hamiltonian. *J. Graph Theory*, 18(8):817–829, 1994.

[Lip79]   W. Lipski, Jr. More on permutation generation methods. *Computing*, 23(4):357–365, 1979.

[LL07]   P. Li and G. Liu. Cycles in circuit graphs of matroids. *Graphs Combin.*, 23(4):425–431, 2007.

[LL08a]   P. Li and G. Liu. Hamilton cycles in circuit graphs of matroids. *Comput. Math. Appl.*, 55(4):654–659, 2008.

[LL08b]   G. Liu and P. Li. Paths in circuit graphs of matroids. *Theoret. Comput. Sci.*, 396(1–3):258–263, 2008.

[LL12]   P. Li and G. Liu. The properties of graphs of matroids. In Y. Zhang, editor, *New Frontiers in Graph Theory*, chapter 10. IntechOpen, Rijeka, 2012.

[LM96]   G. M. Lawrence and W. E. McClintock. Compact ultraviolet imaging microchannel plate detectors using coded anode converter (CODACON) readout systems. In *Proceedings SPIE*, volume 2831, pages 104–111, 1996.

[LMNW23] Z. Lipták, F. Masillo, G. Navarro, and A. Williams. Constant time and space updates for the sigma-tau problem. In F. M. Nardini, N. Pisanti, and R. Venturini, editors, *String Processing and Information Retrieval - 30th International Symposium, SPIRE 2023, Pisa, Italy, September 26-28, 2023, Proceedings*, volume 14240 of *Lecture Notes in Computer Science*, pages 323–330. Springer, 2023.

[Lov70]   L. Lovász. Problem 11. In *Combinatorial Structures and Their Applications (Proc. Calgary Internat. Conf., Calgary, Alberta, 1969)*, page 497. Gordon and Breach, New York, 1970.

[LP07]   T. Lam and A. Postnikov. Alcoved polytopes. I. *Discrete Comput. Geom.*, 38(3):453–478, 2007.

[LP24]   A. V. Ledezma and A. G. Pastine. *s*-stable Kneser graph are Hamiltonian. *arXiv:2401.15198*, 2024.

[LPR⁺19] G. H. J. Lanel, H. K. Pallage, J. K. Ratnayake, S. Thevasha, and B. A. K. Welihinda. A survey on Hamiltonicity in Cayley graphs and digraphs on different groups. *Discrete Math. Algorithms Appl.*, 11(5):Paper 1930002, 18 pp., 2019.

[LRvBR93] J. M. Lucas, D. Roelants van Baronaigien, and F. Ruskey. On rotations and the generation of binary trees. *J. Algorithms*, 15(3):343–366, 1993.

[LS03]   S. C. Locke and R. Stong. Problem 10892: Spanning cycles in hypercubes. *Amer. Math. Monthly*, 110:440–441, 2003.

[LS09]   Y. Li and J. Sawada. Gray codes for reflectable languages. *Inform. Process. Lett.*, 109(5):296–300, 2009.

[Luc87]   J. M. Lucas. The rotation graph of binary trees is Hamiltonian. *J. Algorithms*, 8(4):503–535, 1987.

[Lud81]   J. Ludman. Gray code generation for MPSK signals. *IEEE Transactions on Communications*, 29(10):1519–1522, 1981.

[Lün89]   H. Lüneburg. *Tools and fundamental constructions of combinatorial mathematics*. Bibliographisches Institut, Mannheim, 1989.

[LW22a]   P. Lapey and A. Williams. Pop & push: ordered tree iteration in $\mathcal{O}(1)$-time. In *33rd International Symposium on Algorithms and Computation*, volume 248 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Paper No. 53, 16 pp. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022.

[LW22b]   P. W. Lapey and A. Williams. A shift Gray code for fixed-content Łukasiewicz words. In C. Bazgan and H. Fernau, editors, *Combinatorial Algorithms - 33rd International Workshop, IWOCA 2022, Trier, Germany, June 7-9, 2022, Proceedings*, volume 13270 of *Lecture Notes in Computer Science*, pages 383–397. Springer, 2022.

[LW23]   B. Liu and D. Wong. Generating cyclic rotation Gray codes for stamp foldings and semi-meanders. In *Combinatorial algorithms*, volume 13889 of *Lecture Notes in Comput. Sci.*, pages 271–281. Springer, Cham, [2023] ©2023.

[LZ94]   X.-L. Li and F.-J. Zhang. Hamiltonicity of a type of interchange graphs. *Discrete Appl. Math.*, 51(1–2):107–111, 1994. Proceedings of 2nd Twente Workshop on Graphs and Combinatorial Optimization (Enschede, 1991).

[LZB92]   S. Latifi, S.-Q. Zheng, and N. Bagherzadeh. Optimal ring embedding in hypercubes with faulty links. In *Digest of Papers: FTCS-22, The Twenty-Second Annual International Symposium on Fault-Tolerant Computing, Boston, Massachusetts, USA, July 8-10, 1992*, pages 178–184. IEEE Computer Society, 1992.

[Mad70]   W. Mader. Über den Zusammenhang symmetrischer Graphen. *Arch. Math. (Basel)*, 21:331–336, 1970.

[Mar85]   D. Marušič. Vertex transitive graphs and digraphs of order p^k. In *Cycles in graphs (Burnaby, B.C., 1982)*, volume 115 of *North-Holland Math. Stud.*, pages 115–128. North-Holland, Amsterdam, 1985.

[Mar87]   D. Marušič. Hamiltonian cycles in vertex symmetric graphs of order $2p^2$. *Discrete Math.*, 66(1–2):169–174, 1987.

[Mar88]   D. Marušič. On vertex-transitive graphs of order qp. *J. Combin. Math. Combin. Comput.*, 4:97–114, 1988.

[McD15]   D. C. McDonald. Connectedness and Hamiltonicity of graphs on vertex colorings. *arXiv:1507.05344*, 2015.

[MF84]   M. Mor and A. S. Fraenkel. Cayley permutations. *Discrete Math.*, 48(1):101–112, 1984.

[Mil63]   W. H. Mills. Some complete cycles on the n-cube. *Proc. Amer. Math. Soc.*, 14:640–643, 1963.

[Min65]   G. Minty. A simple algorithm for listing all the trees of a graph. *IEEE Trans. Circuit Theory*, 12(1):120, 1965.

[ML72]   G. H. J. Meredith and E. K. Lloyd. The Hamiltonian graphs O_4 to O_7. In *Combinatorics (Proc. Conf. Combinatorial Math., Math. Inst., Oxford, 1972)*, pages 229–236, 1972.

[ML73]   G. H. J. Meredith and E. K. Lloyd. The footballers of Croam. *J. Combin. Theory Ser. B*, 15:161–166, 1973.

[MM23a]   A. Merino and T. Mütze. Combinatorial generation via permutation languages. III. Rectangulations. *Discrete Comput. Geom.*, 70(1):51–122, 2023.

[MM23b]   A. Merino and T. Mütze. Traversing combinatorial 0/1-polytopes via optimization. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science—FOCS 2023*, pages 1282–1291. IEEE Computer Soc., Los Alamitos, CA, [2023] ©2023.

[MMM22]   A. Merino, O. Mička, and T. Mütze. On a combinatorial generation problem of Knuth. *SIAM J. Comput.*, 51(3):379–423, 2022.

[MMN23]   A. Merino, T. Mütze, and Namrata. Kneser graphs are Hamiltonian. In *STOC'23—Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 963–970. ACM, New York, [2023] ©2023.

[MMP21]   M. Milich, T. Mütze, and M. Pergel. On flips in planar matchings. *Discrete Appl. Math.*, 289:427–445, 2021.

[MMW22]   A. Merino, T. Mütze, and A. Williams. All your bases are belong to us: Listing all bases of a matroid by greedy exchanges. In P. Fraigniaud and Y. Uno, editors, *11th International Conference on Fun with Algorithms, FUN 2022, May 30 to June 3, 2022, Island of Favignana, Sicily, Italy*, volume 226 of *LIPIcs*, pages 22:1–22:28. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[MN08]   T. Mansour and G. Nassar. Gray codes, loopless algorithm and partitions. *J. Math. Model. Algorithms*, 7(3):291–310, 2008.

[MN10]   T. Mansour and G. Nassar. Loop-free Gray code algorithms for the set of compositions. *J. Math. Model. Algorithms*, 9(4):343–356, 2010.

[MN19]   C. M. Mynhardt and S. Nasserasr. Reconfiguration of colourings and dominating sets in graphs. In F. Chung, R. Graham, F. Hoffman, L. Hogben, R. Mullin, and D. B. West, editors, *50 Years of Combinatorics, Graph Theory, and Computing*. Chapman and Hall/CRC, 2019.

[MN20]   T. Mütze and J. Nummenpalo. A constant-time algorithm for middle levels Gray codes. *Algorithmica*, 82(5):1239–1258, 2020.

[MN24]   T. Mütze and Namrata. Hamiltonicity of Schrijver graphs and *s*-stable Kneser graphs. *arXiv:2401.01681*, 2024.

[MNV11]   T. Mansour, G. Nassar, and V. Vajnovszki. Loop-free Gray code algorithm for the **e**-restricted growth functions. *Inform. Process. Lett.*, 111(11):541–544, 2011.

[MNW21]   T. Mütze, J. Nummenpalo, and B. Walczak. Sparse Kneser graphs are Hamiltonian. *J. Lond. Math. Soc. (2)*, 103(4):1253–1275, 2021.

[MP82]   D. Marušič and T. D. Parsons. Hamiltonian paths in vertex-symmetric graphs of order 5*p*. *Discrete Math.*, 42(2–3):227–242, 1982.

[MP16]   T. Manneville and V. Pilaud. Graph properties of graph associahedra. *Sém. Lothar. Combin.*, 73:Art. B73d, 31 pp., [2014-2016].

[MPCZS01] E. Munarini, C. Perelli Cippo, and N. Zagaglia Salvi. On the Lucas cubes. *Fibonacci Quart.*, 39(1):12–21, 2001.

[MS05]   K. Mikawa and I. Semba. An $O(1)$ time algorithm for generating Fibonacci strings. *Electronics and Communications in Japan, Part 2*, 88(9):67–72, 2005.

[MS17]   T. Mütze and P. Su. Bipartite Kneser graphs are Hamiltonian. *Combinatorica*, 37(6):1207–1219, 2017.

[MSW18]   T. Mütze, C. Standke, and V. Wiechert. A minimum-change version of the Chung-Feller theorem for Dyck paths. *European J. Combin.*, 69:260–275, 2018.

[Müt16]   T. Mütze. Proof of the middle levels conjecture. *Proc. London Math. Soc.*, 112(4):677–713, 2016.

[Müt24a]   T. Mütze. A book proof of the middle levels theorem. *Combinatorica*, 44(1):205–208, 2024.

[Müt24b]   T. Mütze. On Hamilton cycles in graphs defined by intersecting set systems. *Notices Amer. Math. Soc.*, 71(5):583–592, 2024.

[MV13]   T. Mansour and V. Vajnovszki. Efficient generation of restricted growth words. *Inform. Process. Lett.*, 113(17):613–616, 2013.

[Naa01]   M. Naatz. A note on a question of C. D. Savage. *SIAM J. Discrete Math.*, 14(1):116–120, 2001.

[Nad84]   D. Naddef. Pancyclic properties of the graph of some 0-1 polyhedra. *J. Combin. Theory Ser. B*, 37(1):10–26, 1984.

[Nak24]   S. Nakano. A Gray code of ordered trees. In X. Chen and B. Li, editors, *Theory and Applications of Models of Computation — 18th Annual Conference, TAMC 2024, Hong Kong, China, May 13–15, 2024, Proceedings*, volume 14637 of *Lecture Notes in Computer Science*, pages 26–37. Springer, 2024.

[Nis18]   N. Nishimura. Introduction to reconfiguration. *Algorithms (Basel)*, 11(4):Paper 52, 25 pp., 2018.

[NP81]   D. J. Naddef and W. R. Pulleyblank. Hamiltonicity and combinatorial polyhedra. *J. Combin. Theory Ser. B*, 31(3):297–312, 1981.

[NP84]   D. J. Naddef and W. R. Pulleyblank. Hamiltonicity in (0-1)-polyhedra. *J. Combin. Theory Ser. B*, 37(1):41–52, 1984.

[NW75]   A. Nijenhuis and H. S. Wilf. *Combinatorial algorithms*. Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London, 1975. Computer Science and Applied Mathematics.

[oei21]   The On-Line Encyclopedia of Integer Sequences. http://oeis.org, 2024.

[ÖP14]   P. R. J. Östergård and V. H. Pettersson. On the maximum length of coil-in-the-box codes in dimension 8. *Discrete Appl. Math.*, 179:193–200, 2014.

[Ord67]   R. J. Ord-Smith. Algorithm 308: Generation of the permutations in pseudo-lexicographic order [G6]. *Commun. ACM*, 10(7):452, 1967.

[PI79]   W. H. Payne and F. M. Ives. Combination generators. *ACM Trans. Math. Softw.*, 5(2):163–172, 1979.

[Pil22]   V. Pilaud. Acyclic reorientation lattices and their lattice quotients. *Sém. Lothar. Combin.*, 86B:Art. 67, 12 pp., 2022.

[Pos09]   A. Postnikov. Permutohedra, associahedra, and beyond. *Int. Math. Res. Not. IMRN*, (6):1026–1106, 2009.

[Pot12]   V. N. Potapov. Construction of Hamiltonian cycles with a given spectrum of edge directions in an n-dimensional Boolean cube. *J. Appl. Ind. Math.*, 6(3):339–345, 2012. Original Russian text in *Diskretn. Anal. Issled. Oper.*, 19(2):76–83, 2012.

[PP01]   A. L. Perezhogin and V. N. Potapov. On the number of Hamiltonian cycles in a Boolean cube. *Diskretn. Anal. Issled. Oper. Ser. 1*, 8(2):52–62, 2001.

[PPR23]   A. Padrol, V. Pilaud, and J. Ritter. Shard polytopes. *Int. Math. Res. Not. IMRN*, (9):7686–7796, 2023.

[PR85]   A. Proskurowski and F. Ruskey. Binary tree Gray codes. *J. Algorithms*, 6(2):225–238, 1985.

[PR91]   G. Pruesse and F. Ruskey. Generating the linear extensions of certain posets by transpositions. *SIAM J. Discrete Math.*, 4(3):413–422, 1991.

[PR93]   G. Pruesse and F. Ruskey. Gray codes from antimatroids. *Order*, 10(3):239–252, 1993.

[PR94]   G. Pruesse and F. Ruskey. Generating linear extensions fast. *SIAM J. Comput.*, 23(2):373–386, 1994.

[PR95]   G. Pruesse and F. Ruskey. The prism of the acyclic orientation graph is Hamiltonian. *Electron. J. Combin.*, 2:Paper 5, 6 pp., 1995.

[PR09]   I. Pak and R. Radoičić. Hamiltonian paths in Cayley graphs. *Discrete Math.*, 309(17):5501–5508, 2009.

[Pro21]   J. Propp. Lattice structure for orientations of graphs. *arXiv:math/0209005*, 2021.

[PS19]   V. Pilaud and F. Santos. Quotientopes. *Bull. Lond. Math. Soc.*, 51(3):406–420, 2019.

[PT98]   K. G. Paterson and J. Tuliani. Some new circuit codes. *IEEE Trans. Inform. Theory*, 44(3):1305–1309, 1998.

[PW79]   E. S. Page and L. B. Wilson. *An introduction to computational combinatorics*, volume 9 of *Cambridge Computer Science Texts*. Cambridge University Press, New York, 1979.

[Ram90]   M. Ramras. A new method of generating Hamiltonian cycles on the n-cube. *Discrete Math.*, 85(3):329–331, 1990.

[RC81]   J. P. Robinson and M. Cohn. Counting sequences. *IEEE Trans. Comput.*, 30(1):17–23, 1981.

[RCUG01]   E. Rivera-Campo and V. Urrutia-Galicia. Hamilton cycles in the path graph of a set of points in convex position. *Comput. Geom.*, 18(2):65–72, 2001.

[Rea12]   N. Reading. From the Tamari lattice to Cambrian lattices and beyond. In *Associahedra, Tamari lattices and related structures*, volume 299 of *Progr. Math.*, pages 293–322. Birkhäuser/Springer, Basel, 2012.

[RND77]   E. M. Reingold, J. Nievergelt, and N. Deo. *Combinatorial algorithms: theory and practice*. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1977.

[Roy73]   M. K. Roy. Reflection-free permutations, rosary permutations, and adjacent transposition algorithms. *Commun. ACM*, 16(5):312–313, 1973.

[RP90]   F. Ruskey and A. Proskurowski. Generating binary trees by transpositions. *J. Algorithms*, 11(1):68–84, 1990.

[RS59]   E. Rapaport-Strasser. Cayley color groups and Hamilton lines. *Scripta Math.*, 24:51–58, 1959.

[RS93]   F. Ruskey and C. D. Savage. Hamilton cycles that extend transposition matchings in Cayley graphs of S_n. *SIAM J. Discrete Math.*, 6(1):152–166, 1993.

[RS94a]   D. J. Rasmussen and C. D. Savage. Hamilton-connected derangement graphs on S_n. *Discrete Math.*, 133(1–3):217–223, 1994.

[RS94b]   F. Ruskey and C. D. Savage. Gray codes for set partitions and restricted growth tails. *Australas. J. Combin.*, 10:85–96, 1994.

[RS96]   F. Ruskey and C. D. Savage. A Gray code for combinations of a multiset. *European J. Combin.*, 17(5):493–500, 1996.

[RSW95]   D. Rasmussen, C. D. Savage, and D. B. West. Gray code enumeration of families of integer partitions. *J. Combin. Theory Ser. A*, 70(2):201–229, 1995.

[RSW12]   F. Ruskey, J. Sawada, and A. Williams. Binary bubble languages and cool-lex order. *J. Combin. Theory Ser. A*, 119(1):155–169, 2012.

[Rus88a]   F. Ruskey. Adjacent interchange generation of combinations. *J. Algorithms*, 9(2):162–180, 1988.

[Rus88b]   F. Ruskey. Research problems 90 and 91. *Discrete Math.*, 70:111–112, 1988.

[Rus89]   F. Ruskey. Transposition generation of alternating permutations. *Order*, 6(3):227–233, 1989.

[Rus92]   F. Ruskey. Generating linear extensions of posets by transpositions. *J. Combin. Theory Ser. B*, 54(1):77–101, 1992.

[Rus93]   F. Ruskey. Simple combinatorial Gray codes constructed by reversing sublists. In *Algorithms and computation (Hong Kong, 1993)*, volume 762 of *Lecture Notes in Comput. Sci.*, pages 201–208. Springer, Berlin, 1993.

[Rus03]   F. Ruskey. Combinatorial generation. Book draft, 2003.

[RvB00]   D. Roelants van Baronaigien. A loopless Gray-code algorithm for listing k-ary trees. *J. Algorithms*, 35(1):100–107, 2000.

[RvBR92]   D. Roelants van Baronaigien and F. Ruskey. Generating permutations with given ups and downs. *Discrete Appl. Math.*, 36(1):57–65, 1992.

[RW08]   F. Ruskey and A. Williams. Generating balanced parentheses and binary trees by prefix shifts. In J. Harland and P. Manyem, editors, *Proceedings of the Fourteenth Symposium on Computing: The Australasian Theory Symposium (CATS 2008), Wollongong, NSW, Australia, January 22–25*, volume 77 of *CRPIT*, pages 107–115. Australian Computer Society, 2008.

[RW09]   F. Ruskey and A. Williams. The coolest way to generate combinations. *Discrete Math.*, 309(17):5305–5320, 2009.

[RW10]   F. Ruskey and A. Williams. An explicit universal cycle for the $(n-1)$-permutations of an n-set. *ACM Trans. Algorithms*, 6(3):Art. 45, 12 pp., 2010.

[Sav89]   C. D. Savage. Gray code sequences of partitions. *J. Algorithms*, 10(4):577–595, 1989.

[Sav90]   C. D. Savage. Generating permutations with k-differences. *SIAM J. Discrete Math.*, 3(4):561–573, 1990.

[Sav93]   C. D. Savage. Long cycles in the middle two levels of the Boolean lattice. *Ars Combin.*, 35(A):97–108, 1993.

[Sav97]   C. D. Savage. A survey of combinatorial Gray codes. *SIAM Rev.*, 39(4):605–629, 1997.

[Saw05]   J. Sawada. Oracles for vertex elimination orderings. *Theoret. Comput. Sci.*, 341(1-3):73–90, 2005.

[Saw07]   J. Sawada. A simple Gray code to list all minimal signed binary representations. *SIAM J. Discrete Math.*, 21(1):16–25, 2007.

[Sch78]   A. Schrijver. Vertex-critical subgraphs of Kneser graphs. *Nieuw Arch. Wisk. (3)*, 26(3):454–461, 1978.

[Sch99]   J. Schmid. Boolean layer cakes. *Theoret. Comput. Sci.*, 217(2):255–278, 1999. Special issue: ORDAL'96 (Ottawa, ON).

[SE99]   M. Schwartz and T. Etzion. The structure of single-track Gray codes. *IEEE Trans. Inform. Theory*, 45(7):2383–2396, 1999.

[Sed77]   R. Sedgewick. Permutation generation methods. *Comput. Surveys*, 9(2):137–164, 1977.

[Sek60]   M. Sekanina. On an ordering of the set of vertices of a connected graph. *Spisy Přírod. Fak. Univ. Brno*, No. 412:137–141, 1960.

[Sha49]   C. E. Shannon. The synthesis of two-terminal switching circuits. *Bell System Tech. J.*, 28:59–98, 1949.

[Sha68]   H. Shank. A note on Hamilton circuits in tree graphs. *IEEE Trans. Circuit Theory*, 15(1):86, 1968.

[Sie93]   G. Sierksma. The skeleton of the symmetric traveling salesman polytope. *Discrete Appl. Math.*, 43(1):63–74, 1993.

[Sie98]   G. Sierksma. Interchange graphs and the Hamiltonian cycle polytope. *Discrete Appl. Math.*, 81(1–3):217–224, 1998.

[Sim78]   G. J. Simmons. Almost all n-dimensional rectangular lattices are Hamilton-laceable. *Congr. Numer.*, XXI:649–661, 1978. Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1978).

[SK78]   B. D. Sharma and R. K. Khanna. On m-ary Gray codes. *Inform. Sci.*, 15(1):31–43, 1978.

[SL12]   J. Sawada and R. Li. Stamp foldings, semi-meanders, and open meanders: fast generation algorithms. *Electron. J. Combin.*, 19(2):Paper 43, 16 pp., 2012.

[Sla78]   P. J. Slater. Generating all permutations by graphical transpositions. *Ars Combin.*, 5:219–225, 1978.

[Sla79]   P. J. Slater. A problem concerning restricted Gray codes, open problem session. *Congr. Numer.*, XXIII–XIV:918–919, 1979. Proceedings of the Tenth Southeastern Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1979).

[Sla89]   P. J. Slater. Research problems 109 and 110. *Discrete Math.*, 76(3):293–294, 1989.

[Smi97]   M. J. Smith. Generating spanning trees. Master's thesis, University of Victoria, 1997.

[Sne94]   H. S. Snevily. The snake-in-the-box problem: a new upper bound. *Discrete Math.*, 133(1–3):307–314, 1994.

[Squ94]   M. B. Squire. Two new Gray codes for acyclic orientations. Technical Report 94-14, Department of Computer Science, North Carolina State University, 1994.

[Squ96]   M. B. Squire. Gray codes for *A*-free strings. *Electron. J. Combin.*, 3(1):Paper 17, 29 pp., 1996.

[Squ98]   M. B. Squire. Generating the acyclic orientations of a graph. *J. Algorithms*, 26(2):275–290, 1998.

[SS88]   Y. Saad and M. H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7):867–872, 1988.

[SSK05]   Y. Suzuki, N. Sawada, and K. Kaneko. Hamiltonian cycles and paths in burnt pancake graphs. In M. J. Oudshoorn and S. Rajasekaran, editors, *Proceedings of the ISCA 18th International Conference on Parallel and Distributed Computing Systems, Imperial Palace Hotel, Las Vegas, NV, USA, September 12–14, 2005*, pages 85–90. ISCA, 2005.

[SSW93]   C. D. Savage, M. B. Squire, and D. B. West. Gray code results for acyclic orientations. *Congr. Numer.*, 96:185–204, 1993. Proceedings of the Twenty-fourth Southeastern International Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1993).

[SSW03]   C. D. Savage, I. Shields, and D. B. West. On the existence of Hamiltonian paths in the cover graph of $M(n)$. *Discrete Math.*, 262(1–3):241–252, 2003.

[ST14]   N. Streib and W. T. Trotter. Hamiltonian cycles and symmetric chains in Boolean lattices. *Graphs Combin.*, 30(6):1565–1586, 2014.

[Sta86]   R. P. Stanley. Two poset polytopes. *Discrete Comput. Geom.*, 1(1):9–23, 1986.

[Sta92]   G. Stachowiak. Hamilton paths in graphs of linear extensions for unions of posets. *SIAM J. Discrete Math.*, 5(2):199–206, 1992.

[Sta15]   R. P. Stanley. *Catalan numbers*. Cambridge University Press, New York, 2015.

[Ste64]   H. Steinhaus. *One hundred problems in elementary mathematics*. Basic Books, Inc., Publishers, New York, 1964. With a foreword by M. Gardner.

[Sup05]   I. N. Suparta. A simple proof for the existence of exponentially balanced Gray codes. *Electron. J. Combin.*, 12:Note 19, 5 pp., 2005.

[Sup17]   I. N. Suparta. Some classes of bipartite graphs induced by Gray codes. *Electron. J. Graph Theory Appl. (EJGTA)*, 5(2):312–324, 2017.

[SV15]   A. Sabri and V. Vajnovszki. Two reflected Gray code-based orders on some restricted growth sequences. *Comput. J.*, 58(5):1099–1111, 2015.

[SV17]   A. Sabri and V. Vajnovszki. More restricted growth functions: Gray codes and exhaustive generation. *Graphs Combin.*, 33(3):573–582, 2017.

[SvZ08]   I. N. Suparta and A. J. van Zanten. A construction of Gray codes inducing complete graphs. *Discrete Math.*, 308(18):4124–4132, 2008.

[SW95]   C. D. Savage and P. Winkler. Monotone Gray codes and the middle levels problem. *J. Combin. Theory Ser. A*, 70(2):230–248, 1995.

[SW07]   J. Sawada and D. C. Wong. A fast algorithm to generate Beckett-Gray codes. *Electronic Notes in Discrete Mathematics*, 29:571–577, 2007.

[SW12a]   J. Sawada and A. Williams. Efficient oracles for generating binary bubble languages. *Electron. J. Combin.*, 19(1):Paper 42, 20 pp., 2012.

[SW12b]   B. Stevens and A. Williams. Hamilton cycles in restricted and incomplete rotator graphs. *J. Graph Algorithms Appl.*, 16(4):785–810, 2012.

[SW13a]   J. Sawada and A. Williams. A Gray code for fixed-density necklaces and Lyndon words in constant amortized time. *Theoret. Comput. Sci.*, 502:46–54, 2013.

[SW13b]   X. S. Shen and A. Williams. A 'hot potato' Gray code for permutations. *Electronic Notes in Discrete Mathematics*, 44:89–94, 2013.

[SW14]   B. Stevens and A. Williams. The coolest way to generate binary strings. *Theory Comput. Syst.*, 54(4):551–577, 2014.

[SW16a]   J. Sawada and A. Williams. Greedy flipping of pancakes and burnt pancakes. *Discrete Appl. Math.*, 210:61–74, 2016.

[SW16b]   J. Sawada and A. Williams. Successor rules for flipping pancakes and burnt pancakes. *Theoret. Comput. Sci.*, 609(part 1):60–75, 2016.

[SW19]   X. S. Shen and A. Williams. A middle levels conjecture for multiset permutations with uniform-frequency. Williams College Technical Report CSTR-201901. Available at `http://tmuetze.de/papers/sw1.pdf`, 2019.

[SW20]   J. Sawada and A. Williams. Solving the sigma-tau problem. *ACM Trans. Algorithms*, 16(1):Art. 11, 17 pp., 2020.

[SW21]   X. S. Shen and A. Williams. A *k*-ary middle levels conjecture. In *Book of Abstracts of the 23rd Thailand-Japan Conference on Discrete and Computational Geometry, Graphs, and Games (TJCDCG^3 2021)*, pages 16–17, 2021.

[SW23]   J. Sawada and A. Williams. Constructing the first (and coolest) fixed-content universal cycle. *Algorithmica*, 85(6):1754–1785, 2023.

[SWW17]   J. Sawada, A. Williams, and D. Wong. Necklaces and Lyndon words in colexicographic and binary reflected Gray code order. *J. Discrete Algorithms*, 46/47:25–35, 2017.

[SWW22]   J. Sawada, A. Williams, and D. Wong. Flip-swap languages in binary reflected Gray code order. *Theoret. Comput. Sci.*, 933:138–148, 2022.

[SZ98]   C. D. Savage and C.-Q. Zhang. The connectivity of acyclic orientation graphs. *Discrete Math.*, 184(1–3):281–287, 1998.

[Tak99a]   T. Takaoka. An $O(1)$ time algorithm for generating multiset permutations. In *Algorithms and computation (Chennai, 1999)*, volume 1741 of *Lecture Notes in Comput. Sci.*, pages 237–246. Springer, Berlin, 1999.

[Tak99b]   T. Takaoka. $O(1)$ time algorithms for combinatorial generation by tree traversal. *Comput. J.*, 42(5):400–408, 1999.

[Tch82]   M. Tchuente. Generation of permutations by graphical exchanges. *Ars Combin.*, 14:115–122, 1982.

[Tie24a]   L. Tiemens. Another balanced transposition Gray code for permutations. Unpublished manuscript, 2024.

[Tie24b]   L. Tiemens. Symmetric chain decompositions of combinatorial posets. Master's thesis, TU Berlin, 2024.

[TL73]   D. T. Tang and C. N. Liu. Distance-2 cyclic chaining of constant-weight codes. *IEEE Trans. Computers*, C-22:176–180, 1973.

[TMS24]   I. Tasoulas, K. Manes, and A. Sapounakis. Hamiltonian intervals in the lattice of binary paths. *Electron. J. Combin.*, 31(1):Paper No. 1.39, 31 pp., 2024.

[Too56]   G. C. Tootill. The use of cyclic permuted codes in relay counting circuits. *Proceedings of the IEE - Part B: Radio and Electronic Engineering*, 103:432–436, 1956.

[Tro62]   H. F. Trotter. Algorithm 115: PERM. *Commun. ACM*, 5(8):434–435, 1962.

[Tro92]   W. T. Trotter. *Combinatorics and partially ordered sets*. Johns Hopkins Series in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 1992. Dimension theory.

[Tur67]   J. Turner. Point-symmetric graphs with a prime number of points. *J. Combin. Theory*, 3:136–145, 1967.

[Ued00]   T. Ueda. Gray codes for necklaces. *Discrete Math.*, 219(1–3):235–248, 2000.

[Und78]   P. Underground. On graphs with Hamiltonian squares. *Discrete Math.*, 21(3):323, 1978.

[Vaj01]   V. Vajnovszki. A loopless generation of bitstrings without *p* consecutive ones. In *Combinatorics, computability and logic (Constanţa, 2001)*, Springer Ser. Discrete Math. Theor. Comput. Sci., pages 227–240. Springer, London, 2001.

[Vaj02]   V. Vajnovszki. Gray visiting Motzkins. *Acta Inform.*, 38(11-12):793–811, 2002.

[Vaj03]   V. Vajnovszki. A loopless algorithm for generating the permutations of a multiset. *Theoret. Comput. Sci.*, 307(2):415–431, 2003. Special issue: Random generation of combinatorial objects and bijective combinatorics.

[Vaj07]   V. Vajnovszki. Gray code order for Lyndon words. *Discrete Math. Theor. Comput. Sci.*, 9(2):145–151, 2007.

[Vaj08]   V. Vajnovszki. More restrictive Gray codes for necklaces and Lyndon words. *Inform. Process. Lett.*, 106(3):96–99, 2008.

[vdH13]   J. van den Heuvel. The complexity of change. In S. R. Blackburn, S. Gerke, and M. Wildon, editors, *Surveys in Combinatorics 2013*, volume 409 of *London Mathematical Society Lecture Note Series*, pages 127–160. Cambridge University Press, 2013.

[VS80]   V. E. Vickers and J. Silverman. A technique for generating specialized Gray codes. *IEEE Trans. Comput.*, 29(4):329–331, 1980.

[VV11]   V. Vajnovszki and R. Vernay. Restricted compositions and permutations: from old to new Gray codes. *Inform. Process. Lett.*, 111(13):650–655, 2011.

[VW06]   V. Vajnovszki and T. Walsh. A loop-free two-close Gray-code algorithm for listing *k*-ary Dyck words. *J. Discrete Algorithms*, 4(4):633–648, 2006.

[VW23]   V. Vajnovszki V. and D. Wong. Greedy Gray codes for Dyck words and ballot sequences. In W. Wu
            and G. Tong, editors, *Computing and Combinatorics - 29th International Conference, COCOON
            2023, Hawaii, HI, USA, December 15-17, 2023, Proceedings, Part II*, volume 14423 of *Lecture Notes
            in Computer Science*, pages 29–40. Springer, 2023.

[vZS04]   A. J. van Zanten and I. N. Suparta. Totally balanced and exponentially balanced Gray codes.
            Diskretn. Anal. Issled. Oper. Ser. 1, 11(4):81–98, 2004.

[Wal98]   T. R. Walsh. Generation of well-formed parenthesis strings in constant worst-case time. *J. Algorithms*,
            29(1):165–173, 1998.

[Wal00]   T. R. Walsh. Loop-free sequencing of bounded integer compositions. *J. Combin. Math. Combin.
            Comput.*, 33:323–345, 2000. Special issue: Papers in honour of E. J. Cockayne.

[Wal01]   T. Walsh. Gray codes for involutions. *J. Combin. Math. Combin. Comput.*, 36:95–118, 2001.

[Wal03]   T. Walsh. Generating Gray codes in $O(1)$ worst-case time per word. In *Discrete mathematics and
            theoretical computer science*, volume 2731 of *Lecture Notes in Comput. Sci.*, pages 73–88. Springer,
            Berlin, 2003.

[Wan93]   T. M. Y. Wang. A note on Gray codes for neckties. Unpublished manuscript, 1993.

[Was16]   K. Wasa. Enumeration of enumeration algorithms. *arXiv:1605.05102*, 2016.

[WE02]   E. L. Wilmer and M. D. Ernst. Graphs induced by Gray codes. *Discrete Math.*, 257(2–3):585–598,
            2002. Kleitman and combinatorics: a celebration (Cambridge, MA, 1999).

[Wel61]   M. B. Wells. Generation of permutations by transposition. *Math. Comp.*, 15:192–195, 1961.

[Wes93]   D. B. West. Generating linear extensions by adjacent transpositions. *J. Combin. Theory Ser. B*,
            58(1):58–64, 1993.

[WG84]   D. Witte and J. A. Gallian. A survey: Hamiltonian cycles in Cayley graphs. *Discrete Math.*,
            51(3):293–304, 1984.

[Wil85]   S. G. Williamson. *Combinatorics for computer science.* Computers and Math Series. Computer
            Science Press, Rockville, MD, 1985.

[Wil89]   H. S. Wilf. *Combinatorial algorithms: an update*, volume 55 of *CBMS-NSF Regional Conference
            Series in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia,
            PA, 1989.

[Wil09a]   A. Williams. Loopless generation of multiset permutations using a constant number of variables
            by prefix shifts. In *Proceedings of the Twentieth ACM-SIAM Symposium on Discrete Algorithms
            (SODA 2009)*, pages 987–996. SIAM, Philadelphia, PA, 2009.

[Wil09b]   A. M. Williams. *Shift Gray codes.* PhD thesis, University of Victoria, 2009.

[Wil13]   A. Williams. The greedy Gray code algorithm. In *Algorithms and data structures*, volume 8037 of
            Lecture Notes in Comput. Sci., pages 525–536. Springer, Heidelberg, 2013.

[Woj89]   J. Wojciechowski. A new lower bound for snake-in-the-box codes. *Combinatorica*, 9(1):91–99, 1989.

[WS96]   T. M. Y. Wang and C. D. Savage. A Gray code for necklaces of fixed density. *SIAM J. Discrete
            Math.*, 9(4):654–673, 1996.

[WV06]   M. Weston and V. Vajnovszki. Gray codes for necklaces and Lyndon words of arbitrary base. *Pure
            Math. Appl. (PU.M.A.)*, 17(1-2):175–182, 2006.

[WW77]   D. E. White and S. G. Williamson. Recursive matching algorithms and linear orders on the subset
            lattice. *J. Combin. Theory Ser. A*, 23(2):117–127, 1977.

[WW91]   D. G. Wagner and J. West. Construction of uniform Gray codes. *Congr. Numer.*, 80:217–223, 1991.
            Proceedings of the Twentieth Manitoba Conference on Numerical Mathematics and Computing
            (Winnipeg, MB, 1990).

[XCU10]   L. Xiang, K. Cheng, and K. Ushijima. Efficient generation of Gray codes for reflectable languages.
            In D. Taniar, O. Gervasi, B. Murgante, E. Pardede, and B. O. Apduhan, editors, *Proceedings of
            the Tenth International Conference on Computational Science and Its Applications (ICCSA 2010),
            Fukuoka, Japan, March 23–26, Part IV*, volume 6019 of *Lecture Notes in Comput. Sci.*, pages
            418–426. Springer, 2010.

[XM09]   J.-M. Xu and M. Ma. Survey on path and cycle embedding in some networks. *Front. Math. China*,
            4(2):217–252, 2009.

[XUT00]   L. Xiang, K. Ushijima, and C. Tang. Efficient loopless generation of Gray codes for k-ary trees.
            Inform. Process. Lett., 76(4–6):169–174, 2000.

[YRN18]   K. Yamanaka, M. S. Rahman, and S. Nakano. Enumerating floorplans with columns. *IEICE Trans.
            Fundam. Electron. Commun. Comput. Sci.*, 101-A(9):1392–1397, 2018.

[Yue74]   C. K. Yuen. The separability of Gray code. *IEEE Trans. Information Theory*, IT-20:668, 1974.

[Zak84]   S. Zaks. A new algorithm for generation of permutations. *BIT*, 24(2):196–204, 1984.

[Zém97]   G. Zémor. An upper bound on the size of the snake-in-the-box. *Combinatorica*, 17(2):287–298, 1997.

[ZG86a]   F.-J. Zhang and X.-F. Guo. Hamilton cycles in Euler tour graph. *J. Combin. Theory Ser. B*, 40(1):1–8, 1986.

[ZG86b]   F.-J. Zhang and X.-F. Guo. Hamiltonian cycles in perfect matching graphs. *J. Xinjiang Univ. Nat. Sci.*, 3(4):10–16, 1986.

[ZG87]   F.-J. Zhang and X.-F. Guo. Hamilton cycles in directed Euler tour graphs. *Discrete Math.*, 64(2–3):289–298, 1987.

[Zha15]   J.-Y. Zhang. Vertex-transitive digraphs of order p^5 are Hamiltonian. *Electron. J. Combin.*, 22(1):Paper 1.76, 12 pp., 2015.