A Survey of Combinatorial Gray Codes

Carla Savage *
Department of Computer Science
North Carolina State University
Box 8206
Raleigh, NC 27695-8206

Abstract

The term *combinatorial Gray code* was introduced in 1980 to refer to any method for generating combinatorial objects so that successive objects differ in some pre-specified, small way. This notion generalizes the classical *binary reflected Gray code* scheme for listing n-bit binary numbers so that successive numbers differ in exactly one bit position, as well as work in the 1960's and 70's on minimal change listings for other combinatorial families, including permutations and combinations.

The area of combinatorial Gray codes was popularized by Herbert Wilf in his invited address at the SIAM Discrete Mathematics Conference in 1988 and his subsequent SIAM monograph in which he posed some open problems and variations on the theme. This resulted in much recent activity in the area and most of the problems posed by Wilf are now solved.

In this paper, we survey the area of combinatorial Gray codes, describe recent results, variations, and trends, and highlight some open problems.

1 Introduction

One of the earliest problems addressed in the area of combinatorial algorithms was that of efficiently generating items in a particular combinatorial class in such a way that each item is generated exactly once. Many practical problems require for their solution the sampling of a random object from a combinatorial class or, worse, an exhaustive search through all objects in the class. Whereas early work in combinatorics focused on *counting*, by 1960, it was clear that with the aid of a computer it would be feasible to *list* the objects in combinatorial classes [Leh64]. However, in order for such a listing to be possible, even for objects of

*Research supported in parts by the National Science Foundation Grant No. CCR8906500, the National Security Agency Grant No. MDA904-H-1025, and DIMACS (Center for Discrete Mathematics and Theoretical Computer Science), a National Science Foundation Science and Technology Center, NSF-STC88-09648.

moderate size, combinatorial generation methods must be extremely efficient. A common approach has been to try to generate the objects as a list in which successive elements differ only in a small way. The classic example is the binary reflected Gray code [Gil58, Gra53] which is a scheme for listing all n-bit binary numbers so that successive numbers differ in exactly one bit. The advantage anticipated by such an approach is two-fold. First, generation of successive objects might be faster. Although for many combinatorial families, a straightforward lexicographic listing algorithm requires only constant average time per element, for other families, such as linear extensions, such performance has only been achieved by a Gray code approach [PR94]. Secondly, for the application at hand, it is likely that combinatorial objects which differ in only a small way are associated with feasible solutions which differ by only a small computation. For example in [NW78], Nijenhuis and Wilf show how to use a binary Gray code to speed up computation of the permanent. Aside from computational considerations, open questions in several areas of mathematics can be posed as Gray code problems. Finally, and perhaps one of the main attractions of the area, Gray codes typically involve elegant recursive constructions which provide new insights into the structure combinatorial families.

The term combinatorial Gray code first appeared in [JWW80] and is now used to refer to any method for generating combinatorial objects so that successive objects differ in some pre-specified, usually small, way. However, the origins of minimal change listings can be found in the early work of Gray [Gra53], Wells [Wel61], Trotter [Tro62], Johnson [Joh63], Chase [Cha70], Ehrlich [Ehr73], and Nijenhuis and Wilf [NW78]. Examples of results in this area include (1) listing all permutations of $1 \ldots n$ so that consecutive permutations differ only by the swap of one pair of adjacent elements [Joh63, Tro62], listing all k-element subsets of an n-element set in such a way that consecutive sets differ by exactly one element [BER76, BW84, EHR84, EM84, NW78, Rus88a], (3) listing all strings of balanced parentheses so that consecutive strings differ only in that one left and one right parenthesis have been interchanged [PR85, RP90], (4) listing all binary trees so that consecutive trees differ only by a rotation at a single node [Luc87, LRR93], (5) listing all partitions of an integer n so that in successive partitions, one part has increased by one and one part has decreased by one [Sav89], (6) listing the linear extensions of certain posets so that successive elements differ only by a transposition [Rus92, PR91, Sta92, Wes93], and (7) listing the elements of a Coxeter group so that successive elements differ by a reflection [CSW89].

Gray codes have found applications in such diverse areas as circuit testing [RC81], signal encoding [Lud81], ordering of documents on shelves [Los92], data compression [Ric86], statistics [DH94], graphics and image processing [ASD90], processor allocation in the hypercube [CS90], hashing [Fal88], computing the permanent [NW78], and information storage and retrieval [CCC92].

In recent variations on combinatorial Gray codes, generation problems have been considered in which the difference between successive objects, although fixed, is not required to be small. An example is the problem of listing all permutations of $1 \dots n$ so that consecutive permutations differ in *every* location [Wil89]. The problem of generating all objects in a combinatorial class, each exactly once, so that successive objects differ in a pre-specified way, can be formulated as a Hamilton path/cycle problem: the vertices of the graph are the objects themselves, two vertices being joined by an edge if they differ from each other in the pre-specified way. This graph has a Hamilton path if and only if the required listing of combinatorial objects exists. A Hamilton cycle corresponds to a cyclic listing in which the first and last items also differ in the pre-specified way. But since the problem of determining whether a given graph has a Hamilton path or cycle is NP-complete [GJ79], there is no efficient general algorithm for discovering combinatorial Gray codes.

Frequently in Gray code problems, however, the associated graph possesses a great deal of symmetry. Specifically, it may belong to the class of *vertex transitive* graphs. A graph G is *vertex transitive* if for any pair of vertices u, v of G, there is an automorphism ϕ of G with $\phi(u) = v$. For example, permutations differing by adjacent transpositions give rise to a vertex transitive graph, as do k-subsets of an n-set differing by one element. It is a well-known open problem, due to Lovász, whether every undirected, connected, vertex transitive graph has a Hamilton path [Lov70]. Thus, schemes for generating combinatorial Gray codes in many cases provide new examples of vertex transitive graphs with Hamilton paths or cycles, from which we hope to gain insight into the more general open questions. It is also unknown whether all connected Cayley graphs (a subclass of the vertex transitive graphs) are hamiltonian. For many Gray code problems, especially those involving permutations, the associated graph is a Cayley graph.

Although many Gray code schemes seem to require strategies tailored to the problem at hand, a few general techniques and unifying structures have emerged. The paper [JWW80] considers families of combinatorial objects, whose size is defined by a recurrence of a particular form, and some general results are obtained about constructing Gray codes for these families. Ruskey shows in [Rus92] that certain Gray code listing problems can be viewed as special cases of the problem of listing the linear extensions of an associated poset so that successive extensions differ by a transposition. In the other direction, the discovery of a Gray code frequently gives new insight into the structure of the combinatorial class involved.

So, the area of combinatorial Gray codes includes many questions of interest in combinatorics, graph theory, group theory, and computing, including some well-known open problems. Although there has been steady progress in the area over the past fifteen years, the recent spurt of activity can be traced to the invited address of Herbert Wilf at the SIAM Conference on Discrete Mathematics in San Francisco in June 1988, *Generalized Gray Codes*, in which Wilf described some results and open problems. (These are also reported in his SIAM monograph [Wil89].) Most of the open problems posed by Wilf have now been solved, as well as several related problems, and it is our intention here to follow up on this work. In this paper, we give a brief survey the area of combinatorial Gray codes, describe recent results, variations and trends, and highlight some (new and old) open problems.

This paper is organized into sections as follows: 1. Introduction; 2. Binary Numbers and Variations; 3. Permutations; 4. Subsets, Combinations, and Compositions; 5. Integer Partitions; 6. Set Partitions and Restricted Growth Functions; 7. Catalan Families; 8. Necklaces and Variations; 9. Linear Extension of Posets; 10. Acyclic Orientations; 11. Cayley Graphs and Permutation Gray Codes; 12. Generalizations of de Bruijn Sequences; 13. Concluding Remarks.

In the remainder of this section, we discuss some notation and terminology which will be used throughout the paper.

A Gray code listing of a class of combinatorial objects will be called *max-min* if the first element on the list is the lexicographically largest in the class and the last element is the lexicographically smallest. The Gray code is *cyclic* if the first and last elements on the list differ in the same way prescribed for successive elements of the list by the adjacency criterion.

In many situations, the graph associated with a particular adjacency criterion is bipartite. If the sizes of the two partite sets differ by more than one, the graph cannot have a Hamilton cycle and thus there is no Gray code listing of the objects corresponding to the vertices, at least for the given adjacency criterion. In this case, we say that a *parity problem* exists.

An algorithm to exhaustively list elements of a class $\mathcal{C}$ is called *loop-free* if the worst case time delay between listing successive elements is constant; the algorithm is called *CAT* if, after listing the first element, the total time required by the algorithm to list all elements is $O(N)$, where N is the total number of elements in the class $\mathcal{C}$. The term CAT was coined by Frank Ruskey to stand for constant amortized time per element.

Finally, we note that a Gray code for a combinatorial class is intrinsically bound to the representation of objects in the class. If sets A and B are two alternative representations of a class C under the bijections $\alpha : C \to A$ and $\beta : C \to B$, the closeness of $\alpha(x)$ and $\alpha(y)$ need not imply closeness of $\beta(x)$ and $\beta(y)$. That is, Gray codes are not necessarily preserved under bijection. Examples of this will be seen for several families, including integer partitions, set partitions, and Catalan families.

2 Binary Numbers and Variations

A Gray code for binary numbers is a listing of all n-bit numbers so that successive numbers (including the first and last) differ in exactly one bit position. The best known example is the *binary reflected Gray code* [Gil58, Gra53] which can be described as follows. If L_n denotes the listing for n-bit numbers, then L_1 is the list 0, 1; for $n > 1$, L_n is formed by taking the list for L_{n-1} and pre-pending a bit of '0' to every number, then following that

a. Binary Reflected		b. Balanced		c. Maximum Gap		d. Non-composite	
00000	11000	00000	10111	00000	00101	00000	01001
00001	11001	10000	10101	10000	10101	10000	11001
00011	11011	11000	10001	11000	11101	11000	10001
00010	11010	11100	11001	11100	11001	11100	10101
00110	11110	11110	11101	11110	11011	11101	10100
00111	11111	11111	01101	11111	11010	01101	00100
00101	11101	01111	01100	10111	10010	00101	01100
00100	11100	01110	01000	10011	10110	00001	01000
01100	10100	00110	01010	10001	10100	00011	01010
01101	10101	00010	11010	00001	00100	10011	11010
01111	10111	00011	11011	01001	01100	10010	11011
01110	10110	01011	10011	01000	01101	10110	11111
01010	10010	01001	10010	01010	01111	11110	10111
01011	10011	00001	10110	01110	01011	01110	00111
01001	10001	00101	10100	00110	00011	01111	00110
01000	10000	00111	00100	00111	00010	01011	00010

Figure 1: Examples of 5-bit binary Gray codes.

list by the reverse of L_{n-1} with a bit of '1' prepended to every number. So, for example, $L_2 = 00, 01, 11, 10$, $L_3 = 000, 001, 011, 010, 110, 111, 101, 100$ and L_5 is shown in Figure 1(a). Since the first and last elements of L_n also differ in one bit position, the code is in fact a cycle. It can be implemented efficiently in the sense that successive elements can be generated in worst case constant time [BER76]. Note that a binary Gray code can be viewed as a Hamilton cycle in the n-cube.

In practice, Gray codes with certain additional properties may be desirable (see [GLN88] for a survey). For example, note that as the elements of L_n are scanned, the lowest order (rightmost) bit changes 2^{n-1} times, whereas the highest order bit changes only twice, counting the return to the first element. In certain applications, it is necessary that the number of bit changes be more uniformly distributed among the bit positions, i.e., a *balanced* Gray code is required. Uniformly balanced Gray codes are known to exist only for n a power of two [WW91]. For general n, the requirement that the code be cyclic is relaxed and one requires a Gray code in which each bit position changes either $\lfloor (2^n - 1)/n \rfloor$ or $\lceil (2^n - 1)/n \rceil$

00000	11000	01010	11110
00001	10000	01011	11100
00011	10001	01001	11101
00010	10101	01101	11001
00110	10100	00101	11011
00100	10110	00111	10011
01100	10010	01111	10111
01000	11010	01110	11111

Figure 2: A monotone Gray code for $n = 5$.

times [LS81, VS80]. (See Figure 1(b) for an example from [VS80].) A heuristic method for the general case is proposed in [VS80]. However, the claim that this always produces optimally balanced Gray codes remains to be proven.

In other applications, the requirement is to maximize the *gap* in a Gray code, which is defined in [GLN88] to be the shortest maximal consecutive sequence of 0's (or 1's) among all bit positions. (See Figure 1(c) for an example from [GLN88] in which the gap is 4, which is best possible for $n = 5$.) Goddyn and Gvozdjak report a construction in which GAP(n)/n goes to 1 as n goes to infinity [GG]. Another variation, non-composite n-bit Gray codes, requires that no contiguous subsequence correspond to a path in any k-cube for $2 \leq k \leq n$. Non-composite Gray codes have been constructed for all n [Ram90]. (See Figure 1(d) for an example from [Ram90].)

A new constraint is considered in [SW95]. Define the *density* of a binary string to be the number of 1's in the string. Clearly, no Gray code for binary strings can list them in non-decreasing order of density. However, suppose the requirement is relaxed somewhat. Call a Gray code *monotone* if it runs through the density levels two at a time, that is, consecutive pairs of strings of densities $i, i+1$ precede those of densities $j, j+1$ for all $0 \leq i < j < n$. It is shown in [SW95] that monotone Gray codes can be constructed for all n. An example for $n = 5$ is shown in Figure 2.

Let $\mathcal{B}_n$ be the Boolean lattice of subsets of the set $[n] = \{1,\ 2, \ldots,\ n\}$, ordered by inclusion, and let $\mathcal{H}_n$ denote the Hasse diagram of $\mathcal{B}_n$. The correspondence

$$b_1 b_2 \ldots b_n \rightarrow \{i \mid b_i = 1\ , 1 \leq i \leq n\}$$

$\emptyset$

1 2 3 4 5

1,2 2,3 1,3 3,4 1,4 2,4 4,5 2,5 3, 5 1, 5

1,2,3 1,3,4 1,2,4 2,3,4 2,4,5 2,3,5 3,4,5 1,3,5 1,4,5 1,2,5

1,2,3,4 2,3,4,5 1,3,4,5 1,2,4,5 1,2,3,5

1,2,3,4,5

Figure 3: The Hamilton path in $\mathcal{H}_n$ corresponding to the monotone Gray code in Figure 2.

is a bijection from n-bit binary numbers to subsets of $[n]$ and, under this bijection, a binary Gray code corresponds to a Hamilton path in $\mathcal{H}_n$.

The vertices of $\mathcal{H}_n$ can be partitioned into level sets, $V_0, V_1, \ldots, V_n$, where V_i contains all of the i-element subsets of $[n]$. Then, a monotone Gray code is a Hamilton path in $\mathcal{H}_n$ in which edges between levels i and $i+1$ must precede edges between levels j and $j+1$ if $i < j$ (Figure 3.)

Monotone Gray codes have applications to the theory of interconnection networks, providing an embedding of the hypercube into a linear array which minimizes dilation in both directions [SW95]. In Section 4 we discuss their relationship to the middle two levels problem.

Fix a binary string α and let $B(n, \alpha)$ be the set of *clean words* for α, i.e., the n-bit strings which do not contain α as a contiguous substring. Does the subgraph of the n-cube induced by $B(n, \alpha)$ have a Hamilton path, i.e., is there a Gray code for $B(n, \alpha)$? Squire has shown that the answer is yes if α can be written as $\alpha = \beta\beta \cdots \beta$, where β is a string with the property that no nontrivial prefix of β is also a suffix of β; otherwise there are parity problems for infinitely many n [Squ96].

It is natural to consider an extension of binary Gray codes to m-ary Gray codes. It was shown in [JWW80], using a generalization of the binary reflected Gray code scheme, that it is always possible to list the Cartesian product of finite sets so that successive elements differ only in one coordinate. A similar result is obtained in [Ric86] where each coordinate i is allowed to assume values in some fixed range $0, \ldots, N_i - 1$. Squire generalizes results on clean words to m-ary Gray codes [Squ96], but leaves open the case when m is odd.

Another listing problem for binary numbers, posed by Doug West, involves a change in the underlying graph. View an n-bit string as a subset of $\{1, \ldots n\}$ under the natural bijection $x_1 \ldots x_n \rightarrow \{ \ i \mid x_i = 1 \ \}$. Call two sets adjacent if they differ only in that one element increases by 1, one element decreases by 1, or the element '1' is deleted. The problem is to determine whether there is a Hamilton path in the corresponding graph, called the *augmentation graph*. When $n(n-1)/2$ is even, a parity argument shows there is no Hamilton path. Otherwise, the question is open for $n > 7$.

$n = 2$	$n = 4$	
1 2	1 2 3 4	4 3 2 1
2 1	1 2 4 3	3 4 2 1
	1 4 2 3	3 2 4 1
	4 1 2 3	3 2 1 4
$n = 3$	4 1 3 2	2 3 1 4
	1 4 3 2	2 3 4 1
1 2 3	1 3 4 2	2 4 3 1
1 3 2	1 3 2 4	4 2 3 1
3 1 2	3 1 2 4	4 2 1 3
3 2 1	3 1 4 2	2 4 1 3
2 3 1	3 4 1 2	2 1 4 3
2 1 3	4 3 1 2	2 1 3 4

Figure 4: Johnson-Trotter scheme for generating permutations by adjacent transpositions.

1 2 3 4	3 1 2 4	2 3 1 4
4 1 2 3	4 3 1 2	4 2 3 1
2 3 4 1	1 2 4 3	3 1 4 2
3 4 1 2	2 4 3 1	1 4 2 3
1 3 2 4	3 2 1 4	2 1 3 4
4 1 3 2	4 3 2 1	4 2 1 3
3 2 4 1	2 1 4 3	1 3 4 2
2 4 1 3	1 4 3 2	3 4 2 1

Figure 5: Generating permutations by derangements, due to Lynn Yarbrough.

Consider another criterion: two binary strings are adjacent if they differ either by (1) a rotation one position left or right or (2) by a negation of the last bit. The underlying graph is the *shuffle-exchange network* and a Hamilton path would be a “Gray code” for binary strings respecting this adjacency criterion. The existence of a Hamilton path in the shuffle-exchange graph, a long-standing open problem, was recently established by Feldman and Mysliwietz [FM93].

3 Permutations

Algorithms for generating all permutations of $1\ 2\ \ldots\ n$, for a given integer $n \geq 1$ were surveyed by Sedgewick in [Sed77]. Efficiency considerations provided the motivation for several early attempts to generate permutations in such a way that successive permutations differ only by the exchange of two elements. Such a Gray code for permutations was shown to be possible in several papers, including [Boo65, Boo67, Hea63, Wel61], which are described in [Sed77]. One disadvantage of these algorithms is that the elements exchanged are not necessarily in adjacent positions.

It was shown independently by Johnson [Joh63] and Trotter [Tro62] that it is possible to generate permutations by transpositions even if the two elements exchanged are required to be in adjacent positions. The recursive scheme, illustrated in Figure 4, inserts into each permutation on the list for $n-1$ the element 'n' in each of the n possible positions, moving alternately from right to left, then left to right.

A contrary approach to the problem is to require that permutations be listed so that each one differs from its predecessor in *every* position, that is, by a *derangement*. This problem was posed independently in [Rab84, Wil89]. The existence of such a list when $n \neq 3$ was established in [Met85] using Jackson's theorem [Jac80] and a constructive solution was presented in [EW85]. A simpler construction, ascribed to Lynn Yarbrough, is discussed in [RS87]. Yarbrough's solution is illustrated in Figure 5 and works as follows. Take each permutation on the Johnson-Trotter list for $n-1$, append an 'n', and rotate the resulting permutation, one position at a time, through its n possible cyclic shifts. As a final twist, swap the last two cyclic shifts. It is straightforward to argue that successive permutations differ in every position, using the property of the Johnson-Trotter list that successive permutations differ by adjacent transpositions.

To generalize the problems of generating permutations, at one extreme, by adjacent transpositions, and at the other extreme, by derangements, consider the following. Given n and k satisfying $n \geq k \geq 2$, is it possible to list all permutations so that successive permutations differ in exactly k positions? This is shown to be possible, unless $k = 3$, in

[Put89] and in [Sav90], where the listing is cyclic. It was shown further in [RS94a] that the k positions (in which successive permutations differ) could be required to be *contiguous*. Putnam claims in [Put90] that when k is even (odd) all permutations (even permutations) can be generated by *k-cycles* of elements in contiguous positions. (Putnam's k-cycles need not be of the form $(i, \ \ i+1, \ \ldots, \ i+k-1)$.

An interesting question arose in connection with a problem on Hamilton cycles in Cayley graphs (see Section 11.) Is it possible to generate permutations by "doubly adjacent" transpositions, i.e., so that successive transpositions are of neighboring pairs? Pair $(i, i+1)$ is considered to be a neighbor of $(i+1, i+2)$ if $i+2 \leq n$ and of $(i-1, i)$ if $i-1 \geq 1$. The Johnson-Trotter scheme satisfies this requirement for $n \leq 3$, but not for $n \geq 4$. Such a listing was shown to be possible by Chris Compton in his Ph.D. thesis [Com90]. It might be hoped that this could result in a very efficient permutation generation algorithm: it would become unnecessary to decide which of the $n-1$ adjacent pairs to transpose, only whether the next transposition is to the left or right of the current one. However, in its current form, Compton's algorithm is not practical, and is quite complex, even with the simplifications in [CW93].

4 Subsets, Combinations, and Compositions

Since there is a bijection between the subsets of an n-element set (an *n-set*) and the n-bit binary numbers, any binary Gray code defines a Gray code for subsets: two binary numbers differing in one bit correspond to two subsets differing by the addition or deletion of one element.

For the subclass of combinations (k-subsets of an n-set for fixed k), several Gray codes have been surveyed in [Wil89]. As observed in [BER76], a Gray code for combinations can be extracted from the binary reflected Gray code for n-bit numbers: delete from the binary reflected Gray code list all those elements corresponding to subsets which do not have exactly k elements. That which remains is a list of all k-subsets in which successive sets differ in exactly one element (see Figure 6(a) and compare to Figure 1(a)). The same list is generated by the *revolving door algorithm* in [NW78] and it can be described by a

a. Revolving Door $(n,k) = (5,3)$	b. Strong Minimal Change $(n,k) = (5,3)$	c. Adjacent Interchange $(n,k) = (6,3)$	
123	123	123	246
134	124	124	256
234	134	125	156
124	234	126	146
145	235	136	145
245	135	135	245
345	125	134	345
135	145	234	346
235	245	235	356
125	345	236	456

Figure 6: Examples of Gray codes for combinations.

simple recursive expression.

A more stringent requirement is to list all k-sets with the *strong minimal change property* [EM84]. That is, if a k-set is represented as a sorted k-tuple of its elements, successive k-sets differ in only one position (see Figure 6(b)). Eades and McKay have shown that such a listing is always possible. An earlier solution was reported by Chase in [Cha70].

Perhaps the most restrictive Gray code which has been proposed for combinations is to generate k-subsets of an n-set so that successive sets differ in exactly one element and this element has either increased or decreased by one. This is called the *adjacent interchange property* since if the sets are represented as binary n-tuples, successive n-tuples may differ only by the interchange of a 1 and a 0 in adjacent positions (see Figure 6(c)). However, this is not always possible: it was shown that k-subsets of an n-set can be generated by adjacent interchanges if (i) k=0, 1, n, or $n-1$ or (ii) n is even and k is odd. In all other cases, parity problems prevent adjacent interchange generation [BW84, EHR84, HR88, Rus88a]. It was shown by Chase [Cha89] and by a simpler construction in [Rus93] that combinations can be generated so that successive elements differ either by an adjacent transposition or by the transposition of two bits that have a single '0' bit between them.

There are several open problems about paths between levels of the Hasse diagram of the Boolean lattice, B_n. The most notorious is the *middle two levels* problem which is attributed in [KT88] to Dejter, Erdős, and Trotter and by others to Hável and Kelley. The middle two levels of B_{2k+1} have the same number of elements and induce a bipartite, vertex transitive graph on the k- and $k+1$- element subsets of $[2k+1]$. The question is whether there is a Hamilton cycle in the middle two levels of B_{2k+1}. At first glance, it would appear that one could take a Gray code listing of the k-subsets, in which successive elements differ in one element, and, by taking unions of successive elements, create a list of $k+1$-subsets. Alternating between the lists would give a walk in the middle two levels graph, but, unfortunately, not a Hamilton path, at least not for any known Gray code on k-subsets.

The graph formed by the middle two levels is a connected, undirected, vertex-transitive graph. Thus, either it has a Hamilton path, or it provides a counterexample to the Lovász conjecture. One approach to this problem which has been considered is to try to form a Hamilton cycle as the union of two edge-disjoint matchings. In [DSW88], it was shown that a Hamilton cycle in the middle two levels cannot be the union of two *lexicographic* matchings. However, other matchings may work and new matchings in the middle two levels have been defined [KT88, DKS94].

The largest value of k for which a Hamilton cycle is known to exist is $k = 11$ ($n = 23$.) See Figure 7 for an example when $k = 3$. This unpublished work was done by Moews and Reid using a computer search [MR]. To speed up the search, they used a necklace-based approach, gambling that there would be a Hamilton path through necklaces which could be lifted to a Hamilton cycle in the original graph. We feel that a focus on the middle two levels of the necklace poset, as described in Section 8, is a promising approach to the middle two levels problem.

Is there at least a good lower bound on the length of a longest cycle in the middle two levels of the Boolean lattice? Since this graph is vertex-transitive, a result of Babai [Bab79] shows that there is a cycle of length at least $(3N(k))^{1/2}$, where $N(k)$ is the total number of vertices in the middle two levels of B_{2k+1}. A result of Dejter and Quintana gives a cycle of length $\Omega(N(k)^t)$ where $t = (\log 3)/(\log 4) \approx 0.793$ [DQ87]. This was improved in [Sav93] to

4 6 7	1 2 6	2 4 7	1 4 6	1 3 5
1 4 6 7	1 2 6 7	2 4 5 7	1 3 4 6	1 3 5 6
1 4 7	1 6 7	2 5 7	3 4 6	3 5 6
1 2 4 7	1 5 6 7	2 3 5 7	3 4 5 6	3 5 6 7
1 2 7	5 6 7	2 3 5	3 4 5	3 5 7
1 2 5 7	2 5 6 7	2 3 5 6	3 4 5 7	1 3 5 7
1 5 7	2 6 7	2 5 6	3 4 7	1 3 7
1 4 5 7	2 4 6 7	2 4 5 6	1 3 4 7	1 2 3 7
4 5 7	2 4 6	2 4 5	1 3 4	1 2 3
4 5 6 7	2 3 4 6	2 3 4 5	1 3 4 5	1 2 3 6
4 5 6	2 3 6	2 3 4	1 4 5	1 3 6
1 4 5 6	2 3 6 7	1 2 3 4	1 2 4 5	1 3 6 7
1 5 6	2 3 7	1 2 4	1 2 5	3 6 7
1 2 5 6	2 3 4 7	1 2 4 6	1 2 3 5	3 4 6 7

Figure 7: A Hamilton cycle in the middle two levels of B_7.

$t \approx 0.837$. In a welcome breakthrough, Felsner and Trotter showed the existence of cycles of length at least $0.25N(k)$ [FT95]. The monotone Gray code, described in Section 1, contains as a subpath, a path in the middle two levels of length at least $0.5N(k)$ [SW95]. In [SW95], this was strengthened to get 'nearly Hamilton' cycles in the following sense: for every $\epsilon > 0$, there is an $h \geq 1$ so that if a Hamilton cycle exists in the middle two levels of B_{2k+1} for $1 \leq k \leq h$, then there is a cycle of length at least $(1-\epsilon)N(k)$ in the mid-levels of B_{2k+1} for *all* $k \geq 1$. Since Hamilton cycles are known for $1 \leq k \leq 11$, the construction guarantees a cycle of length at least $0.839N(k)$ in the middle two levels of B_{2k+1} for all $k \geq 1$.

A variation on this problem is the *antipodal layers problem:* for which values of k is there a Hamilton path among the k-sets and $n-k$ - sets of $\{1,\ldots,n\}$ for all n, where two sets are joined by an edge if and only if one is a subset of the other? Results for limited values of k and n are given in [Hur94] and [Sim].

A *composition* of n into k parts is a sequence $(x_1,\ldots,x_k)$ of nonnegative integers whose sum is n. This is traditionally viewed as a placement of n balls into k boxes. Nijenhuis and Wilf asked in the first edition of [NW78] (p. 292, problem 34) whether it was possible to

a. $P(7,6)$		b. $P_3(17,13)$		c. $D(12,12)$	
6 1	3 1^4	13 4	4^4 1^2	12	7 5
5 1^2	3 2 1^2	13 1^4	4^3 1^5	11 1	7 4 1
5 2	3 2^2	10 1^7	4^2 1^9	10 2	7 3 2
4 3	2^3 1	10 4 1^3	7 4 1^6	9 3	6 3 2 1
3 1^2	2^2 1^3	10 7	7 1^{10}	9 2 1	6 4 2
4 2 1	2 1^5	7^2 1^3	4 1^{13}	8 3 1	5 4 3
41^3	1^7	7 4^2 1^2	1^{17}	8 4	5 4 2 1

Figure 8: Gray codes for various families of integer partitions

generate the k-compositions of n so that each is obtained from its predecessor by moving one ball from its box to another. Knuth solved this in 1974 while reading the galleys of the book and in [Kli82], Klingsberg gives a CAT implementation of Knuth's Gray code.

Combinations and compositions can be simultaneously generalized as follows. Let $C(s;m_1,\ldots,m_t)$ denote the set of all ordered t-tuples $(x_1,\ldots,x_t)$ satisfying $x_1+\ldots+x_t=s$ and $0\leq x_i\leq m_i$ for $i=1,\ldots t$. If $m_i\geq s$ for $i=1,\ldots t$, $C(s;m_1,\ldots,m_t)$ is the set of s-combinations of a t-element set. If X is the multiset consisting of m_i copies of element i for $i=1,\ldots t$, then $C(s;m_1,\ldots,m_t)$ is the collection of s-element submultisets, or *s-combinations* of X. In [Ehr73], Ehrlich provides a loopless algorithm to generate multiset combinations so that successive elements differ in only two positions, but not necessarily by just ±1 in those positions. It is shown in [RS96] that a Gray code still exists when the two position can change by only ±1, thereby generalizing Gray code results for both combinations and compositions.

5 Integer Partitions

A partition of an integer n is a sequence of positive integers $x_1\geq x_2\geq\ldots\geq x_k$ satisfying $x_1+x_2+\ldots+x_k=n$. Algorithms for generating integer partitions in standard orders such as lexicographic and antilexicographic were presented in [FL80] and [NW78]. The performance of the algorithms in [FL80] is analyzed in [FL81].

An integer partition in standard representation, $\pi = (x_1, x_2, \ldots, x_k)$, can also be written as a list of pairs $(y_1, m_1), \ldots, (y_l, m_l)$, where the y_i are the distinct integers appearing in the sequence π, and m_i is the number of times y_i appears in π. Ruskey notes in [Rus95] that a lexicographic listing of partitions in this ordered pairs representation has the property that successive elements of the list differ at most in the last three ordered pairs.

Wilf asked the following question regarding a Gray code for integer partitions in the standard representation $\pi = (x_1, x_2, \ldots, x_k)$ [Wil88]: Is there a way to list the partitions of an integer n in such a way that consecutive partitions on the list differ only in that one part has increased by 1 and one part has decreased by 1? (A part of size 1 may decrease to 0 or a 'part' of size 0 may increase to 1.) Yoshimura demonstrated that this was possible for integers $n = 1, \ldots, 12$ [Yos87]. In [Sav89], it is shown constructively to be possible for all n. The result is a bit more general: for all $n \geq k \geq 1$, there is a way to list the set $P(n, k)$, of all partitions of n into integers of size at most k, in Gray code order. Unless $(n, k) = (6, 4)$, the Gray code is max-min. As a consequence, each of the following can also be listed in Gray code order for all n, k satisfying $n \geq k \geq 1$: (1) all partitions of n whose largest part is k, (2) all partitions of n into k or fewer parts, and (3) all partitions of n into exactly k parts. See Figure 8(a) for a Gray code listing of $P(7, 6)$. Exponents in the figure indicate the number of multiple copies.

The approach in [Sav89], was to decompose the partitions problem, $P(n, k)$, into subproblems of two forms, a 'P' form, which was the same form as the original problem, and a new 'M' form. It was then shown that the P and M forms could be recursively defined in terms of (smaller versions of) both forms, thereby yielding a doubly recursive construction of the partitions Gray code. The algorithm has been implemented [Bee90] and can be modified to run in time $O(|P(n, k)|)$.

This strategy has been applied to yield Gray codes for other families of integer partitions. Let $P_\delta(n, k)$ be the set of all partitions of n into parts of size at most k, in which the parts are required to be congruent to 1 modulo δ. When $\delta = 1$, $P_\delta(n, k)$ is just $P(n, k)$. When $\delta = 2$, the elements of $P_\delta(n, k)$ are the partitions of n into odd parts of size at most k. It is shown in [RSW95] that $P_\delta(n, k)$ can be listed so that between successive partitions, one part increases by δ (or δ ones may appear) and another part decreases by δ (or δ ones may disappear.) (See Figure 8(b).) The Gray code is max-min unless $(n,k) = (4\delta+2, 3\delta+1)$, where a max-min Gray code is impossible.

For the case of $D(n,k)$, the set of partitions of n into odd parts of size at most k, the same strategy can be applied, but the construction becomes more complex. Surprisingly, it is still possible to list $D(n,k)$ in Gray code order: between successive partitions one part increases by two and one part decreases by two [RSW95]. (See Figure 8(c).) The Gray code is max-min unless (n,k) is $(9,6)$ or $(12,6)$, in which cases a max-min Gray code is impossible. One observation that follows from this work is that although there are bijections between the sets of partitions of n into odd parts and partitions of n into distinct parts, no bijection can preserve such Gray codes.

The same techniques can be used to investigate Gray codes in other families of integer partitions, but each family has its own quirks: a small number of cases must be handled specially and subsets needed for linking recursively listed pieces can become empty. Nevertheless, we conjecture that each of the following families have Gray code enumerations, for arbitrary values of the parameters n, k, $\delta \geq 1$, $t \geq 1$, $d \geq 1$: partitions of n into (a) distinct odd parts, (b) distinct parts congruent to 1 modulo δ, (c) at most t copies of each part, (d) parts congruent to 1 modulo δ, at most t copies of each part, and (e) exactly d distinct parts.

6 Set Partitions and Restricted Growth Functions

A *set partition* is a decomposition of $\{1, \ldots, n\}$ into a disjoint union of nonempty subsets called *blocks*. Let $S(n)$ denote the set of all partitions of $\{1, \ldots, n\}$. For example, $S(4)$ is shown in Figure 9(a). The *restricted growth functions* (RG functions) of length n, denoted $R(n)$, are those strings $a_1 \ldots a_n$ of non-negative integers satisfying $a_1 = 0$ and $a_i \leq 1 + \max\{a_1, \ldots, a_{i-1}\}$ [SW86].

There is a well-known bijection between $S(n)$ and $R(n)$. For $\pi \in S(n)$, order the blocks of π according to their smallest element, for example, the blocks of $\pi = \{\{9\}, \{1, 2, 7\}, \{4, 10, 11\}, \{3, 5, 6, 8\}\}$ would be ordered $\{1, 2, 7\}$, $\{3, 5, 6, 8\}$, $\{4, 10, 11\}$, $\{9\}$. Label

(a)$S(4)$	(b)$L(4)$ in lexicographic order	(c) Knuth's Gray code	(d) modified Knuth	(e)Ehrlich's algorithm
$\{1,\ 2,\ 3,\ 4\}$	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
$\{1,\ 2,\ 3\},\ \{4\}$	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
$\{1,\ 2,\ 4\},\ \{3\}$	0 0 1 0	0 0 1 2	0 0 1 2	0 0 1 1
$\{1,\ 2\},\ \{3,\ 4\}$	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 2
$\{1,\ 2\},\ \{3\},\{4\}$	0 0 1 2	0 0 1 0	0 0 1 0	0 0 1 0
$\{1,\ 3,\ 4\},\ \{2\}$	0 1 0 0	0 1 2 0	0 1 1 0	0 1 1 0
$\{1,\ 3\},\ \{2,\ 4\}$	0 1 0 1	0 1 2 1	0 1 1 1	0 1 1 2
$\{1,\ 3\},\ \{2\},\{4\}$	0 1 0 2	0 1 2 2	0 1 1 2	0 1 1 1
$\{1,\ 4,\},\ \{2,\ 3\}$	0 1 1 0	0 1 2 3	0 1 2 2	0 1 2 1
$\{1\},\ \{2,\ 3,\ 4\}$	0 1 1 1	0 1 1 2	0 1 2 3	0 1 2 2
$\{1\},\ \{2,\ 3\},\{4\}$	0 1 1 2	0 1 1 1	0 1 2 1	0 1 2 3
$\{1,\ 4\},\ \{2\},\{3\}$	0 1 2 0	0 1 1 0	0 1 2 0	0 1 2 0
$\{1\},\ \{2,\ 4\},\{3\}$	0 1 2 1	0 1 0 0	0 1 0 0	0 1 0 0
$\{1\},\ \{2\},\{3,\ 4\}$	0 1 2 2	0 1 0 1	0 1 0 1	0 1 0 2
$\{1\},\ \{2\},\{3\},\{4\}$	0 1 2 3	0 1 0 2	0 1 0 2	0 1 0 1

Figure 9: Listings of $S(4)$ and $R(4)$.

the blocks of π in order by $0,\ 1,\ 2, \ldots$. The bijection assigns to π the string $a_1 \ldots a_n \in R(n)$, where for $1 \le i \le n$, a_i is the label of the block containing i. The associated string for π above is 0 0 1 2 1 1 0 1 3 2 2. For $n = 4$, the bijection is illustrated in the first two columns of Figure 9.

In [Kay76], Kaye gives a CAT implementation of a Gray code for $S(n)$, attributed to Knuth in [Wil89]. This was another problem posed by Nijenhuis and Wilf in their book [NW78] (p. 292, problem 25) and solved by Knuth while reading the galleys. In this Gray code, successive set partitions differ only in that one element moved to an adjacent block (Figure 9(c).) However, the associated RG functions may differ in many positions. Ruskey [Rus95] describes a modification of Knuth's algorithm in which one element moves to a block at most two away between successive partitions and the associated RG functions differ only in one position by at most two (Figure 9(d).)

Call a Gray code for RG functions *strict* if successive elements differ in only one position and in that position by only ± 1. Strict Gray codes for $R(n)$ were considered in an early paper of Ehrlich where it was shown that for infinitely many values of n, they do not exist [Ehr73]. Nevertheless, Ehrlich was able to find an efficient listing algorithm for $R(n)$ (loop-free) which has the following interesting property: successive elements differ in one position and the element in that position can change by 1, or, if it is the largest element in the string, it can change to 0. Conversely, a 0 can change to a the the largest value v in the string or to $v + 1$. For example, 0 1 0 2 0 2 1 can change to 0 1 0 2 0 0 1 or to 0 1 0 2 3 2 1, and conversely. In the associated list of set partitions, this change corresponds to moving one element to an adjacent block in the partition, where the first and last blocks are considered adjacent (Figure 9(e).)

Ehrlich's results are generalized in [RS94b] to the set of *restricted growth tails*, $T(n, k)$, which are strings of non-negative integers satisfying $a_1 \leq k$ and $a_i \leq 1+\max\{a_1, \ldots, a_{i-1},\ k-1\}$. (These are a variation of the $T(n, m)$ used in [Wil85] for ranking and unranking set partitions.) Note that $T(n, 0) = R(n)$. Because of parity problems, for all k there are infinitely many values of n for which $T(n, k)$ has no *strict* Gray code, that is, one in which only one position changes by 1. However, Gray codes satisfying Ehrlich's relaxed criterion are constructed and they can be made cyclic or max-min, properties not possessed by the earlier Gray codes.

Consider now set partitions into a fixed number of blocks. For $n \geq 1$ and $0 \leq b \leq n$, let $S_b(n)$ be the set of partitions of $\{1, \ldots, n\}$ into exactly b blocks. The bijection between $S(n)$ and $R(n)$ restricts to a bijection between $S_b(n)$ and

$$R_b(n) = \{a_1 \ldots a_n \in R(n) \mid \max\{a_1, \ldots a_n\} = b - 1\}.$$

The Ehrlich paper presents a loop-free algorithm for generating $S_b(n)$ in which successive partitions differ only in that two elements have moved to different blocks [Ehr73]. Ruskey describes a Gray code for $R_b(n)$ (and a CAT implementation) in which successive elements differ in only one position, but possibly by more than 1 in that position [Rus93]. It is shown in [RS94b] that in general, $R_b(n)$ does not have a strict Gray code, even under the relaxed criterion of Ehlich.

It remains open whether there are strict Gray codes for $R(n)$ and $T(n, k)$ when the parity difference is 0.

7 Catalan Families

In several families of combinatorial objects, the size is counted by the Catalan numbers, defined for $n \geq 0$ by

$$C_n = \frac{1}{n+1} \binom{2n}{n}.$$

These include binary trees on n vertices [SW86], well-formed sequences of $2n$ parentheses [SW86], and triangulations of a labeled convex polygon with $n+2$ vertices [STT88]. Since bijections are known between most members of the Catalan family, a Gray code for one member of the family gives implicitly a listing scheme for every other member of the family. However, the resulting lists may not look like Gray codes, since bijections need not preserve minimal changes between elements.

The problem of generating all binary trees with a given number of nodes was considered in several early papers, including [RH77], [Zak80], and [Zer85]. However, Gray codes in the Catalan family were first considered in [PR85], where binary trees were represented as strings of balanced parentheses. It was shown in [PR85] that strings of balanced parentheses could be listed so that consecutive strings differ only by the interchange of one left and one right parenthesis. For example '(()())(())' could follow '(()(())())'. The same problem was considered in [RP90] with the additional restriction that only *adjacent* left and right parentheses could be interchanged. For example, now '(()())(())' could not follow '(()(())())', but could follow '((()))(())'. The result of [RP90] is that all balanced strings of n pairs of parentheses can be generated by adjacent interchanges if and only if n is even or $n < 5$, and for these cases, a CAT algorithm is given.

A different minimal change criterion, focusing on binary trees, was considered in [Luc87] and [LRR93]: list all binary trees on n nodes so that consecutive trees differ only by a left or right rotation at a single node. The rotation operation is common in data structures where it is used to restructure binary search trees, while preserving the ordering properties. It was shown that such a Gray code is always possible and it can be generated efficiently

[LRR93]. With a more intricate construction, Lucas was able to show that the associated graph was hamiltonian [Luc87], giving a cyclic Gray code.

It so happens that under a particular bijection between binary trees with n nodes and the set of all triangulations of a labeled convex polygon with $n + 2$ vertices, rotation in a binary tree corresponds to the flip of a diagonal in the triangulation [STT88]. So, the results of [Luc87, LRR93] also give a listing of all triangulations of a polygon so that successive triangulations differ only by the flip of a single diagonal.

8 Necklaces and Variations

An *n-bead, k-color necklace* is an equivalence class of k-ary n-tuples under rotation. Figure 10 lists the lexicographically smallest representatives of the n-bead, k-color necklaces for $(n, k) = (5, 2), (7, 2)$, and $(3, 3)$. Wilf asked if it is possible to generate necklaces efficiently, possibly in constant time per necklace. A proposed solution, the FKM algorithm of Fredricksen, Kessler, and Maiorana, had no proven upper bound better than $O(nk^n)$ [FK86, FM78]. In [WS90] a new algorithm was presented with time complexity $O(nN_k^n)$, where N_k^n is the number of n-bead necklaces in k colors. Subsequently, a tight analysis of the original FKM algorithm showed that it could, in fact, be implemented to run in time $O(N_k^n)$, giving an optimal solution [RSW92].

Neither of the algorithms above gives a Gray code for necklaces. Can representatives of all binary n-bead necklaces be listed so that successive strings differ only in one bit position (as in Figure 10)? A parity argument shows that this is impossible for even n, but for odd n the question remains open. However, in the case of necklaces with a fixed number of 1's, Wang showed, with a very intricate construction, how to construct a Gray code in which successive necklace representatives differ only by the swap of a 0 and a 1 [Wan94, WS96] (Figure 11.)

It remains open whether necklaces with a fixed number of 1's can be generated in constant amortized time, either by a modification the FKM algorithm, by a Gray code, or by any other method.

The Gray code adjacency criterion can be generalized to necklaces with $k \geq 2$ beads

a. 5-bead binary	b. 7-bead binary		c. 3-bead ternary
11111	1111111	0010011	222
01111	0111111	0000011	122
01011	0110111	0000111	122
00011	0010111	0001111	111
00111	0010101	0101111	011
00101	0000101	0101011	021
00001	0001101	0001011	022
00000	0011101	0001001	012
	0011111	0000001	002
	0011011	0000000	001

Figure 10: Examples of Gray codes for necklaces.

by requiring that successive necklaces differ in exactly one position and in that position by only 1. We conjecture that this can be done if and only if nk is odd. (Parity problems prevent a Gray code when nk is even.)

For necklaces of fixed weight, when is it possible to list all n-bead k-color necklaces of weight w so that successive necklaces differ in exactly two positions, one of which has increased by one and the other, decreased by one? We know of no counterexamples.

To construct a slightly different set of objects, call two k-ary strings equivalent if one is a rotation or a *reversal* of the other. The equivalence classes under this relation are called *bracelets*. Lisonek [Lis93] shows how to modify the necklace algorithm of [WS90] to generate bracelets. We know of no Gray code for bracelets and it is open whether it is possible to generate bracelets in constant amortized time. When beads have distinct colors, bracelets are the *rosary permutations* of [Har71, Rea72].

Define a new relation $\mathcal{R}$ on n-bead binary necklaces by $x\mathcal{R}y$ if some member of x becomes a member of y by changing a 0 to a 1 in one bit position. The reflexive transitive closure, $\mathcal{R}^*$ is a partial order and the resulting poset is the *necklace poset*. For $k \geq 0$ and $n = 2k+1$, the middle two levels of this poset, consisting of necklaces of density k and $k+1$, have the same number of elements. Does the bipartite subgraph induced by the middle two levels have a Hamilton path? This graph is not vertex-transitive, but it may encapsulate the

7 beads, 4 ones	9 beads, 3 ones	8 beads, 4 ones
0001111	000000111	00001111
0101011	000001011	00010111
0011011	000010011	00100111
0010111	000100011	00101011
0011101	000100101	00101101
	000101001	00011101
	000011001	00110101
	000010101	00110011
	000001101	00011011

Figure 11: Examples of Gray codes for binary necklaces with a fixed number of ones.

"hard part" of the middle two levels problem described in Section 4.

A *necktie* of n bands in k colors is an equivalence class of k-ary n-tuples under reversal. If a necktie is identified with the lexicographically smallest element in its equivalence class, Wang [Wan93] shows that for $n \geq 3$, a Gray code exists if and only if either n or k is odd. For this result, two neckties are adjacent if and only if they differ only in one position and in that position by ± 1 modulo k. Further results on neckties appear in [RW94].

9 Linear Extension of Posets

A partially ordered set $(S, \preceq)$ is a set S together with a binary relation $\preceq$ on S which is reflexive, transitive, and antisymmetric. A linear extension of a poset is a permutation $x_1 x_2 \ldots x_n$ of the elements of the poset which is consistent with the partial order, that is, if $x_i \preceq x_j$ in the partial order, then $i \leq j$. The problem of efficiently generating all the linear extensions of a poset, in any order, has been studied in [KV83, KS74, VR81]. The area of Gray codes for linear extensions of a poset was introduced by Frank Ruskey in [Rus88b, PR91] as a setting in which to generalize the study of Gray codes for combinatorial objects. For example, if the Hasse diagram of the poset consists of two disjoint chains, one of length m and the other of length n, then there is a one-to-one correspondence between the linear extensions of the poset and the combinations of m objects chosen from $m + n$.

If the poset consists of a collection of disjoint chains, the linear extensions correspond to multiset permutations. Other examples are described in [Rus92].

To study the existence of Gray codes, Ruskey constructs a transposition graph corresponding to a given poset. The vertices are the linear extensions of the poset, two vertices being joined by an edge if they differ by a transposition. The resulting graph is bipartite. In [Rus88b], Ruskey makes the conjecture that whenever the parity difference is at most one, the graph of the poset has a Hamilton path. The conjecture is shown to be true for some special cases in [Rus92], including posets whose Hasse diagram consists of disjoint chains and for series parallel posets in [PR93]. The techniques which have been successful so far involve cutting and linking together listings for various subposets in rather intricate ways.

In most cases where it is known how to list linear extensions by transpositions, it is also possible to require *adjacent* transpositions, although possibly with a more complicated construction [PR91, RS93, Sta92, Wes93]. It has been shown that if the linear extensions of a poset Q, with $|Q|$ even, can be listed by adjacent transpositions, then so can the linear extensions of $Q|P$, for *any* poset P [Sta92], where $Q|P$ represents the union of posets P and Q with the additional relations $\{p \preceq q \mid p \in P, q \in q\}$.

However, most problems in this area remain open. For example, even if the Hasse diagram of the poset consists of a single tree, the parity difference may be greater than one. This makes an inductive approach difficult. If the Hasse diagram consists of two trees, each with an odd number of vertices, the parity difference will be at most one, but it is unknown whether the linear extensions can be listed in this case. The problem is also open for posets whose Hasse diagram is a grid or tableau tilted ninety degrees [Rus].

Calculating the parity difference itself can be difficult and Ruskey [Rus] has several examples of posets for which the parity difference is unknown. (Some parity differences are calculated in [KR88].) Recently, Stachowiak has shown that computing the parity difference is #P-complete [Sta]. Even counting the number of linear extensions of a poset is an open problem for some specific posets, for example, the Boolean lattice [SK87]. Brightwell and Winkler have recently shown that the problem of counting the number of linear extensions of a given poset is #-P complete [BW92]. On the brighter side, Pruesse and Ruskey [PR94] have found a CAT algorithm for listing linear extensions so that successive extensions differ by one or two adjacent transpositions and Canfield and Williamson [CW95] have shown how to make it loop-free.

In [PR93], Pruesse and Ruskey consider antimatroids, of which posets are a special case. Analogous to the case of linear extensions of a poset, they show that the sets of an antimatroid can be listed so that successive sets differ by at most two elements.

10 Acyclic Orientations

For an undirected graph G, an *acyclic orientation* of G is a function on the edges of G which assigns a direction (u, v) or (v, u) to each edge uv of G in such a way that the resulting digraph has no directed cycles. Consider the problem of listing the acyclic orientations of G so that successive list elements differ only by the orientation of a single edge. It is not hard to see that when G is a tree with n edges, such a listing corresponds to an n bit binary Gray code; when G is K_n, an acyclic orientation corresponds to a permutation of the vertices and the Johnson-Trotter Gray code for permutations provides the required listing for acyclic orientations.

Denote by $AO(G)$ the graph whose vertices are the acyclic orientations of G, two vertices adjacent if and only if the corresponding orientations differ only in the orientation of a single edge. The graph $AO(G)$ is bipartite and is connected as long as G is simple. Edelman asked, whenever the partite sets have the same size, whether $AO(G)$ is hamiltonian. It is shown in [SSW93] that the answer is yes for several classes of graphs, including trees, odd length cycles, complete graphs, odd ladder graphs, and chordal graphs. On the other hand, the parity difference is shown to be more than one for several cases, including cycles of even length and the complete bipartite graphs $K_{m,n}$ with $m, n > 1$ and m or n even. The problem appears to be difficult and it is even open whether $AO(K_{m,n})$ is hamiltonian when mn is odd. However, the square of $AO(G)$ is hamiltonian for any G [PR95, SZ95, Squ94c], which means that acyclic orientations can be listed so that successive elements differ in the orientations of at most two edges.

The problem of counting acyclic orientations is #P-complete [Lin86] and it is an open question whether there is a CAT algorithm to generate them. The fastest listing algorithm known, due to Squire [Squ94b], requires $O(n)$ average time per orientation, where n is the number of vertices of the graph.

The linear extensions and acyclic orientations problems can be simultaneously generalized as follows. For a simple undirected graph G and a subset R of the edges of G, fix an acyclic orientation σ_R of the edges of R. Let $AO_R(G)$ be the subgraph of $AO(G)$ induced by the acyclic orientations of G which agree with σ_R on R. Is this bipartite graph hamiltonian whenever the parity difference allows?

When $R = \emptyset$, $AO_R(G)$ is the acyclic orientations graph of G. $AO_R(G)$ becomes the linear extensions adjacency graph of an n element poset P when when $G = K_n$ and R and σ_R are defined by the covering relations in P. In contrast to the situation for linear extensions and acyclic orientations, the square of $AO_R(G)$ is not necessarily hamiltonian. Counterexamples appear in [Squ94c] and [PR95].

11 Cayley Graphs and Permutation Gray Codes

Many Gray code problems for permutations are best discussed in the setting of Cayley graphs. Given a finite group G and a set X of elements of G, the Cayley graph of G, $C[G, X]$, is the undirected graph whose vertices are the elements of G and in which there is an edge joining u and v if and only if $ux = v$ or $vx = u$ for some $x \in X$. Equivalently, uv is an edge in G if and only if $u^{-1}v$ or $v^{-1}u$ is in X. $C[G, X]$ is always vertex transitive and is connected if and only if $X \cup X^{-1}$ generates G. It is an open question whether every Cayley graph is hamiltonian. (There are generating sets for which the Cayley digraph is *not* hamiltonian [Ran48].) This is a special case of the more general conjecture of Lovàsz that every connected, undirected, vertex-transitive graph has a Hamilton path [Lov70]. Results on Hamilton cycles are surveyed in [CG96, WG84] for Cayley graphs, in [Als81] for vertex transitive graphs, and in [Gou91] for general graphs.

Suppose the group G is S_n, the symmetric group of permutations of n symbols, and let X be a generating set of S_n. Then a Hamilton cycle in the Cayley graph $C[G, X]$ can be regarded as a Gray code for permutations in which successive permutations differ only by a generator in X. Even in the special case of $G = S_n$, it is still open whether every Cayley graph of S_n has a Hamilton cycle. One of the most general results on the hamiltonicity of Cayley graphs for permutations was discovered by Kompel'makher and Liskovets in 1975. First note that for S_n generated by the basis $X = \{(12), (23), \ldots, (n-1\ n)\}$, the Johnson-Trotter algorithm from Section 3 for generating permutations by adjacent transpositions gives a Hamilton cycle in $C[G, X]$. Kompel'makher and Liskovets generalized this result to show that if X is *any* set of transpositions generating S_n, then $C[S_n, X]$ is hamiltonian [KL75]. Independently, and with a much simpler argument, Slater showed that these graphs have Hamilton paths [Sla78]. Tchuente [Tch82] extended the results of [KL75, Sla78] to show that the Cayley graph of S_n, on any generating set X of transpositions, is not only hamiltonian, but *Hamilton-laceable*, that is, for any two vertices u, v of different parity there is a Hamilton path which starts at u and ends at v.

It is unknown whether any of these results generalize to the case when X is a generating set of involutions (elements of order 2) for S_n. An involution need not be a transposition: for example the product of disjoint transpositions is an involution. In perhaps the simplest nontrivial case, when S_n is generated by three involutions, it is easy to show that if any two of the generators commute, then the Cayley graph is hamiltonian. (Cayley graphs arising in *change ringing* frequently have this property [Ran48, Whi83].) However if no two of the three involutions commute, it is open whether the Cayley graph is hamiltonian. As a specific example, we have not been able to determine whether there is a Gray code for permutations in which successive permutations may differ only by one of the three operations: (i) exchange positions 1 and 2, (ii) reverse the sequence, and (iii) reverse positions 2 through n of the sequence.

Conway, Sloane, and Wilks have a related result on Gray codes for reflection groups: if G is an irreducible Coxeter group (a group generated by geometric reflections) and if X is the canonical basis of reflections for the group, then $C[G, X]$ is hamiltonian [CSW89]. This result makes use of the fact that in any set of three or more generators from this basis, there is always some pair of generators which commute (the Coxeter diagram for the basis is a tree). It is straightforward to show for groups G and H, with generating sets X and Y, respectively, that if $C[G, X]$ and $C[H, Y]$ are hamiltonian, and at least one of G and H have even order, then $C[G \times H, X \times Y]$ is hamiltonian. As noted in [CSW89], since any reflection group R is a direct product of irreducible Coxeter groups $G_1 \times G_2 \times \ldots \times G_k$, with X_i the canonical basis for G_i, the Cayley graph of R with respect to the basis $X_1 \times X_2 \times \ldots \times X_n$ is hamiltonian.

This result has an interesting geometric interpretation. We can associate with a finite reflection group a tessellation of a surface in n-space by a spherical $(n-1)$-simplex. The spherical simplices of the tesselation correspond to group elements, and the boundary shared by two simplices corresponds to a reflection in the bounding hyperplane. Thus, a Hamilton cycle in the Cayley graph corresponds to a traversal of the surface, visiting each simplex exactly once.

It seems likely that if there do exist non-hamiltonian Cayley graphs of S_n, there will be examples in which the vertices have small degree, such as S_n generated by three involutions as described above. As a candidate counterexample, Wilf suggested the group of permutations generated by the two cycles (1 2) and (1 2 3 ... n) [Wil88]. Compton and Williamson were able to find a Hamilton cycle in this graph using their Gray code for generating permutations by doubly adjacent transpositions, described in Section 3 [CW93].

The results of [KL75, Sla78] were generalized in a different way in [RS93]. It was shown there that when $n \geq 5$, for any generating set X of transpositions of S_n and for any transposition $\tau \in X$, all permutations in S_n can be listed so that successive permutations differ only by a transposition in X and *every other transposition is* τ. That is, the perfect matching in $C[S_n, X]$ defined by τ is contained in a Hamilton cycle. One application of this result is to Cayley graphs for the alternating group, A_n, consisting of all even permutations of $1 \ldots n$. For example, letting $X = \{(1\ 2), (1\ 3), \ldots, (1\ n)\}$ and $\tau = (1\ n)$, the result in [RS93] implies that the Cayley graph of A_n with respect to the generating set $X = \{(1\ 2\ n), (1\ 3\ n), \ldots, (1\ n-1\ n)\ \}$ is hamiltonian. This result was obtained earlier with a direct argument by Gould and Roth [GR87].

12 Generalizations of de Bruijn Sequences

A *de Bruijn sequence of order n* is a circular binary sequence of length 2^n in which every n-bit number appears as a contiguous subsequence. This provides a Gray code listing of binary sequences in which successive elements differ by a rotation one position left followed by a change of the last element. It is known that these sequences exist for all n and the standard proof shows that a de Bruijn sequence of order n corresponds to an Euler tour in the de Bruijn digraph whose vertices are the binary n-tuples, with one edge $(x_1 \ldots x_{n-1}) \to (x_2 \ldots x_n)$ for every binary n-tuple $x_1 \ldots x_n$ [dB46, Mar34].

This result has been generalized for k-ary n-tuples [Fre82], for higher dimensions (de Bruijn toroids) [Coc88, FFMS85], and for k-ary toroids [HI95]. It is known also, for any m satisfying $n \leq m \leq 2^n$, that there is a cyclic binary sequence of length m in which no n-tuple appears more than once [Yoe62]. So, the de Bruijn graph contains cycles of all lengths $m \geq n$. Results on de Bruijn cycles have been applied to random number generation in information theory [Gol64] and in computer architecture, where the de Bruijn graph is recognized as a bounded degree derivative of the shuffle-exchange network [ABR90].

Chung, Diaconis, and Graham generalized the notion of a de Bruijn sequence for binary numbers to *universal cycles* for other families of combinatorial objects [CDG92]. Universal cycles for combinations were studied by Hurlbert in [Hur90] and some interesting problems remain open. A universal cycle of order n for permutations is a circular sequence $x_0 \ldots x_{N-1}$ of length $N = n!$ of symbols from $\{1, \ldots, M\}$ in which every permutation $\pi_1 \ldots \pi_n$ of $1 \ldots n$ is *order isomorphic* to some contiguous subsequence $x_{t+1} \ldots x_{t+n}$. Order isomorphic means that for $1 \leq i, j \leq n$, $\pi_i < \pi_j$ iff $x_{t+i} < x_{t+j}$. As an example, the sequence 123415342154213541352435 is a universal cycle of order 4 with $M = 5$. In [CDG92], the goal is to choose M as small as possible to guarantee the existence of a universal cycle. It is clear that M must satisfy $M \geq n + 1$ for $n > 2$. It is conjectured in [CDG92] that $M = n + 1$ for all $n > 2$, although the best upper bound they were able to obtain was $M \leq n + 6$. Even for $n = 5$ it is open whether $M = 6$ is sufficient.

As another approach, we can relax the constraint on the length of the sequence, while requiring $M = n$, and ask for the shortest circular sequence of symbols from $\{1, \ldots, n\}$ which contains every permutation as a contiguous subsequence at least once. Jacobson and West have a simple construction for such a sequence of length $2n!$ [JW].

13 Concluding Remarks

This paper has included a sampling of Gray code results in several areas, particularly those which have appeared since the survey of Wilf [Wil89], in which many of these problems were posed. Good references for early work on Gray codes are [Ehr73] and [NW78]. For a comprehensive treatment of Gray codes and other topics in combinatorial generation, we look forward to the book in preparation by Ruskey [Rus95]. Additional information on Gray codes also appears in the survey of Squire [Squ94a]. For surveys on related material, see [Als81] for long cycles in vertex transitive graphs, [Gou91] for hamiltonian cycles, [WG84] and the recent update [CG96] for Cayley graphs, and [Sed77] for permutations.

Acknowledgements I am grateful to Herb Wilf for collecting and sharing such an intriguing array of 'Gray code' problems. His work, as well as his enthusiasm, has been inspiring. I would also like to thank Frank Ruskey, my frequent co-author, for many provoking discussions, and for his constant supply of interesting problems. Both have provided helpful comments on earlier versions on this manuscript.

References

[ABR90] F. Annexstein, M. Baumslag, and A. L. Rosenberg. Group action graphs and parallel architectures. *SIAM J. Comput.*, 19(3):544–569, 1990.

[Als81] B. R. Alspach. The search for long paths and cycles in vertex-transitive graphs and digraphs. In *Combinatorial Mathematics VIII (Springer Lecture Notes 884)*, pages 14–22. Springer-Verlag, 1981.

[ASD90] D. J. Amalraj, N. Sundararajan, and G. Dhar. A data structure based on Gray code encoding for graphics and image processing. *Proceedings of the SPIE: International Society for Optical Engineering*, pages 65–76, 1990.

[Bab79] L. Babai. Long cycles in vertex-transitive graphs. *J. Graph Theory*, 3:301–304, 1979.

[Bee90] J. Beer. Implementation of an algorithm to list Gray code sequences of partitions. Manuscript, North Carolina State University, 1990.

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

[Boo65] J. Boothroyd. Algorithm 6. PERM. *Computer Bulletin*, 9(3):104, 1965.

[Boo67] J. Boothroyd. Permutation of the elements of a vector (Algorithm 29) and Fast permutation of the elements of a vector (Algorithm 30). *Comput. J.*, 10:310–312, 1967.

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

[BW92] G. Brightwell and P. Winkler. Counting linear extensions is #P-complete. *Order*, 8:225–242, 1992.

[CCC92] C. C. Chang, H. Y. Chen, and C. Y. Chen. Symbolic Gray codes as a data allocation scheme for two disc systems. *Comput. J.*, 35(3):299–305, 1992.

[CDG92] F. Chung, P. Diaconis, and R. Graham. Universal cycles for combinatorial structures. *Discrete Math.*, 110:43–60, 1992.

[CG96] S. J. Curran and J. A. Gallian. Hamiltonian cycles and paths in Cayley graphs and digraphs - a survey, 1996.

[Cha70] P. J. Chase. Algorithm 382: Combinations of M out of N objects. *Comm. ACM*, 13(6):368, 1970.

[Cha89] P. J. Chase. Combination generation and Graylex ordering. *Congr. Numer.*, 69:215–242, 1989.

[Coc88] J. C. Cocke. Toroidal tilings from de Bruijn-good cyclic sequences. *Discrete Math.*, 70:209–210, 1988.

[Com90] R. C. Compton. *Hamilton cycles in the Cayley graph of S_n and a doubly adjacent Gray code*. PhD thesis, Department of Mathematics, University of California, San Diego, 1990.

[CS90] M. Chen and K. G. Shin. Subcube allocation and task migration in hypercube machines. *IEEE Transactions on Computers*, 39(9):1146–1155, 1990.

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

[CW93] R. C. Compton and S. G. Williamson. Doubly adjacent Gray codes for the Symmetric group: how to braid n strands. *Linear and Multilinear Algebra*, 34:237–293, 1993.

[CW95] E. R. Canfield and S. G. Williamson. A loop-free algorithm for generating the linear extensions of a poset. *Order*, 12:1–18, 1995.

[dB46] N. G. de Bruijn. A combinatorial problem. *Proc. Nederl. Akad. Wetensch.*, 49:758–764, 1946.

[DH94] P. Diaconis and S. Holmes. Gray codes for randomization procedures. *Statistics and Computing*, 4:287–302, 1994.

[DKS94] D. Duffus, H. A. Kierstead, and H. S. Snevily. An explicit 1-factorization in the middle of the Boolean lattice. *J. Combin. Theory Ser. A*, 65(2):334–342, 1994.

[DQ87] I. J. Dejter and J. Quintana. Long cycles in revolving door graphs. *Congr. Numer.*, 60:163–168, 1987.

[DSW88] D. Duffus, B. Sands, and R. Woodrow. Lexicographic matchings cannot form hamiltonian cycles. *Order*, 5:149–161, 1988.

[Ehr73] G. Ehrlich. Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *J. ACM*, 20:500–513, 1973.

[EHR84] P. Eades, M. Hickey, and R. C. Read. Some Hamilton paths and a minimal change algorithm. *J. ACM*, 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.*, 19:131–133, 1984.

[EW85] R. B. Eggleton and W. D. Wallis. Problem 1186: Solution I. *Math. Mag.*, 58(2):112–113, 1985.

[Fal88] C. Faloutsos. Gray codes for partial match and range queries. *IEEE Transactions on Software Engineering*, 14(10):1381–1393, 1988.

[FFMS85] C. T. Fan, S. M. Fan, S. L. Ma, and M. K. Sin. On de Bruijn arrays. *Ars Combin.*, 19A:205–213, 1985.

[FK86] H. Fredricksen and I. J. Kessler. An algorithm for generating necklaces of beads in two colors. *Discrete Math.*, 61:181–188, 1986.

[FL80] T. I. Fenner and G. Loizou. A binary tree representation and related algorithms for generating integer partitions. *Comput. J.*, 23(4):332–337, 1980.

[FL81] T. I. Fenner and G. Loizou. An analysis of two related loop-free algorithms for generating integer partitions. *Acta Inform.*, 16:237–252, 1981.

[FM78] H. Fredricksen and J. Maiorana. Necklaces of beads in k colors and k-ary de Bruijn sequences. *Discrete Math.*, 23(3):207–210, 1978.

[FM93] R. Feldman and P. Mysliwietz. The shuffle-exchange network has a hamiltonian path. Technical Report 116, Fachbereich Mathematik-Informatik, Universitat-GH-Paderborn, 1993.

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

[FT95] S. Felsner and W. T. Trotter. Colorings of diagrams of interval orders and α-sequences. *Discrete Math.*, 144, 1995.

[GG] L. Goddyn and P. Grozdjak. Personal communication.

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

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability, a Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.

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

[Gol64] S. W. Golomb, editor. *Digital Communications with Space Applications.* Prentice Hall, 1964.

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

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

[Gra53] F. Gray. Pulse code communications. U.S. Patent 2632058, March 1953.

[Har71] K. Harada. Generation of rosary permutations expressed in hamiltonian circuits. *Comm. ACM*, 14:373–379, 1971.

[Hea63] B. R. Heap. Permutations by interchanges. *Comput. J.*, 6:293–94, 1963.

[HI95] G. Hurlbert and G. Isaak. On the de Bruijn torus problem. *J. Combin. Theory Ser. A*, 1995.

[HR88] T. Hough and F. Ruskey. An efficient implementation of the Eades, Hickey, Read adjacent interchange combination generation algorithm. *J. Combin. Math. Combin. Comput.*, 4:79–86, 1988.

[Hur90] G. Hurlbert. *Universal Cycles: On Beyond de Bruijn*. PhD thesis, Department of Mathematics, Rutgers University, 1990.

[Hur94] G. Hurlbert. The antipodal layers problem. *Discrete Math.*, 128:237–243, 1994.

[Jac80] B. Jackson. Hamilton cycles in regular 2-connected graphs. *J. Combin. Theory Ser. B*, 29:27–46, 1980.

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

[JW] M. Jacobson and D. B. West. Personal communication.

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

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

[KL75] V. L. Kompel'makher and V. A. Liskovets. Sequential generation of arrangements by means of a basis of transpositions. *Kibernetica*, 3:17–21, 1975.

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

[KR88] C. W. Ko and F. Ruskey. Solution of some multi-dimensional lattice path parity difference recurrence relations. *Discrete Math.*, 71:47–56, 1988.

[KS74] D. E. Knuth and J. Szwarcfiter. A structured program to generate all topological sorting arrangements. *Inform. Process. Lett.*, 2:153–157, 1974.

[KT88] H. A. Kierstead and W. T. Trotter. Explicit matchings in the middle levels of the Boolean lattice. *Order*, 5:163–171, 1988.

[KV83] A. D. Kalvin and Y. L. Varol. On the generation of all topological sortings. *J. Algorithms*, 4:150–162, 1983.

[Leh64] D. H. Lehmer. The machine tools of combinatorics. In E. Beckenbach, editor, *Applied Combinatorial Mathematics*, pages 5–31. John Wiley & Sons, 1964.

[Lin86] N. Linial. Hard enumeration problems in geometry and combinatorics. *SIAM J. Alg. Disc. Meth.*, 7(2):331–335, 1986.

[Lis93] P. Lisonek. Generating bracelets, 1993. Preprint.

[Los92] R. M. Losee. A Gray code based ordering for documents on shelves: Classificaton for browsing and retrieval. *Journal of the American Society for Information Science*, 43(4):312–322, 1992.

[Lov70] L. Lovász. Problem 11. In *Combinatorial Structures and their Applications*. Gorden and Breach, 1970.

[LRR93] 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.

[LS81] J. E. Ludman and J. L. Sampson. A technique for generating Gray codes. *J. Statist. Plann. Inference*, 5:171–180, 1981.

[Luc87] J. Lucas. The rotation graph of binary trees is hamiltonian. *J. Algorithms*, 8:503–535, 1987.

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

[Mar34] M. H. Martin. A problem in arrangements. *Bull. Amer. Math. Soc.*, 40:859–864, 1934.

[Met85] J. Metzger. Problem 1186. *Math. Mag.*, 58:113–114, 1985.

[MR] D. Moews and M. Reid. Electronic mail communication (via J. Gallian).

[NW78] A. Nijenhuis and H. S. Wilf. *Combinatorial Algorithms for Computers and Calculators*. Academic Press, 1978.

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

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

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

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

[PR95] G. Pruesse and F. Ruskey. The prism of the acyclic orientation graph is hamiltonian. *Electron. J. Combin.*, 2, 1995.

[Put89] T. C. Putnam. A Gray code variant: sequencing permutations via fixed points. Manuscript, 1989.

[Put90] T. C. Putnam. Combinatorial Gray code and a generalization of the Johnson- Trotter algorithm to contiguous k-cycles. Manuscript, 1990. Presented at the Fifth SIAM Conference on Discrete Mathematics, Atlanta.

[Rab84] S. Rabinowitz. Problem 1186. *Math. Mag.*, 57:109, 1984.

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

[Ran48] R. A. Rankin. A campanological problem in group theory. *Math. Proc. Cambridge Philos. Soc.*, 44:17–25, 1948.

[RC81] J. Robinson and M. Cohn. Counting sequences. *IEEE Transactions on Computers*, C-30:17–23, 1981.

[Rea72] R. C. Read. Note on the generation of rosary permutations. *Comm. ACM*, 15:775, 1972.

[RH77] F. Ruskey and T. C. Hu. Generating binary trees lexicographically. *SIAM J. Comput.*, 6(4):745–758, 1977.

[Ric86] D. Richards. Data compression and Gray-code sorting. *Inform. Process. Lett.*, 22:210–205, 1986.

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

[RS87] D. F. Rall and P. J. Slater. Generating all permutations by graphical derangements. Unpublished manuscript, 1987.

[RS93] F. Ruskey and C. D. Savage. Hamilton cycles which 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.*, 133:217–223, 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.*, 68:1–8, 1996.

[RSW92] F. Ruskey, C. D. Savage, and T. M. Y. Wang. Generating necklaces. *J. Algorithms*, 13:414–430, 1992.

[RSW95] D. J. 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.

[Rus] F. Ruskey. private communication.

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

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

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

[Rus93] F. Ruskey. Simple combinatorial Gray codes constructed by reversing sublists. In *Proceedings, 4th ISAAC, Hong Kong, Lecture Notes in Computer Science, 762*, pages 201–208. Springer-Verlag, 1993.

[Rus95] F. Ruskey. *Combinatorial Generation.* 1995. Book in preparation.

[RW94] F. Ruskey and T. M. Y. Wang. Generating neckties: algorithms, Gray codes, and parity differences, 1994. Preprint.

[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.

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

[Sim] J. E. Simpson. Hamiltonian bipartite graphs. Preprint.

[SK87] J. Sha and D. J. Kleitman. The number of linear extensions of subset ordering. *Discrete Math.*, 63:271–279, 1987.

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

[Squ94a] M. B. Squire. Combinatorial Gray codes and efficient generation. Technical Report 94-17, North Carolina State University, 1994.

[Squ94b] M. B. Squire. Generating the acyclic orientations of a graph. Submitted to *Journal of Algorithms*, 1994.

[Squ94c] M. B. Squire. Two new Gray codes for acyclic orientations. Technical Report 94-14, North Carolina State University, 1994.

[Squ96] M. B. Squire. Gray codes for A-free strings. *Electron. J. Combin.*, 3(R17), 1996.

[SSW93] C. D. Savage, M. B. Squire, and D. B. West. Gray code results for acyclic orientations. *Congr. Numer.*, 96:185–204, 1993.

[Sta] G. Stachowiak. Finding parity difference by involutions. Preprint.

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

[STT88] D. D. Sleator, R. E. Tarjan, and W. P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *Journal of the American Mathematical Society*, 1(3):647–681, 1988.

[SW86] D. Stanton and D. White. *Constructive Combinatorics*. Springer-Verlag, 1986.

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

[SZ95] C. D. Savage and C. Q. Zhang. A note on the connectivity of acyclic orientations graphs, 1995. Preprint.

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

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

[VR81] Y. L. Varol and D. Rotem. An algorithm to generate all topological sorting arrangements. *Comput. J.*, 24:83–84, 1981.

[VS80] V. E. Vickers and J. Silverman. A technique for generating specialized Gray codes. *IEEE Transactions on Computers*, C-29:329–331, 1980.

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

[Wan94] T. M. Y. Wang. *A Gray Code for Necklaces of Fixed Density.* PhD thesis, Department of Computer Science, North Carolina State University, 1994.

[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*, 57:58–64, 1993.

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

[Whi83] A. T. White. Ringing the changes. *Math. Proc. Camb. Phil. Soc.*, 94:203–215, 1983.

[Wil85] S. G. Williamson. *Combinatorics for Computer Science*. Computer Science Press, 1985.

[Wil88] H. S. Wilf. Generalized Gray codes. In *SIAM Conference on Discrete Mathematics*, 1988. Invited talk.

[Wil89] H. S. Wilf. *Combinatorial Algorithms: An Update*. SIAM, 1989.

[WS90] T. M. Y. Wang and C. D. Savage. A new algorithm for generating necklaces. In *Proceedings, Twenty-Eighth Annual Allerton Conference on Communication, Control, and Computing*, pages 72–81, 1990.

[WS96] T. M. Y. Wang and C. D. Savage. Gray codes for necklaces of fixed density, 1996.

[WW91] D. G. Wagner and J. West. Construction of uniform Gray codes. *Congr. Numer.*, 80:217–223, 1991.

[Yoe62] M. Yoeli. Binary ring sequences. *Amer. Math. Monthly*, 69:852–855, 1962.

[Yos87] N. Yoshimura. *Ranking and unranking algorithms for trees and other combinatorial objects*. PhD thesis, Computer and Information Science, University of Pennsylvania, 1987.

[Zak80] S. Zaks. Lexicographic generation of ordered trees. *Theoret. Comput. Sci.*, 10:63–82, 1980.

[Zer85] D. Zerling. Generating binary trees using rotations. *J. ACM*, 32(3):694–701, 1985.