and

$$\mathsf{G}^3 = [000, 001, 011, 010, 110, 111, 101, 100].$$

Figure 2.1 depicts the binary reflected Gray codes $\mathsf{G}^1, \ldots, \mathsf{G}^4$.

Our first result is to prove that any G^n is a Gray code. Since the codes G^n are defined recursively, it is most natural to prove the result by induction on n.

THEOREM 2.1 *For any integer $n \geq 1$, G^n is a Gray code.*

PROOF The proof is by induction on n. For $n = 1$, the result is easily seen to be true. As an induction hypothesis, suppose that G^{n-1} is a Gray code, for some integer $n \geq 2$. We will prove that G^n is a Gray code.

First, it is clear that G^n contains all 2^n binary n-tuples. By induction, the 2^{n-1} n-tuples that begin with "0" are contained in the first half of G^n, and the 2^{n-1} n-tuples that begin with "1" are contained in the second half of G^n.

It remains to verify the minimal change property. Consider two consecutive n-tuples in G^n, say G^n_i and G^n_{i+1}. There are in fact three cases to consider, depending on the value of i.

First, if $0 \leq i \leq 2^{n-1} - 2$, then G^n_i and G^n_{i+1} are formed from two consecutive $(n-1)$-tuples in G^{n-1} by prepending a "0" to each of them. Therefore, by induction, G^n_i and G^n_{i+1} have Hamming distance equal to 1.

The second case is $2^{n-1} \leq i \leq 2^n - 2$. This is similar to the first case. This time, G^n_i and G^n_{i+1} are formed from two consecutive $(n-1)$-tuples in G^{n-1} by prepending a "1" to each of them. Therefore, by induction, G^n_i and G^n_{i+1} have Hamming distance equal to 1.

The final case is $i = 2^{n-1} - 1$. $\mathsf{G}^n_{2^{n-1}-1}$ and $\mathsf{G}^n_{2^{n-1}}$ are both formed from $\mathsf{G}^{n-1}_{2^{n-1}-1}$, by prepending a "0" and a "1" respectively. Therefore, $\mathsf{G}^n_{2^{n-1}-1}$ and $\mathsf{G}^n_{2^{n-1}}$ have Hamming distance equal to 1. (Note that this last case works out precisely because of the fact that the second copy of G^{n-1} in G^n is in reverse order.)

By induction on n, the result is true for all integers $n \geq 1$. ∎

We now present Algorithm 2.3, which computes the successor function for the binary reflected gray Code G^n. Suppose that the binary vector $A = [a_{n-1}, \ldots, a_0]$ represents the set $T \subseteq \{1, \ldots, n\}$. Then $a_i = 1$ if $n - i \in T$ and $a_i = 0$ otherwise. Let $\mathsf{w}(A)$ denote the *Hamming weight* of A (i.e., the number of "1"s in the vector A); note that $\mathsf{w}(A) = |T|$.

Algorithm 2.3 works as follows. If $\mathsf{w}(A)$ is even, then the last bit of A (namely, a_0) is flipped; if $\mathsf{w}(A)$ is odd, then we find the first "1" from the right, and flip the next bit (to the left). The last vector in G^n, which has no successor, is $[1, 0, \ldots, 0]$. This corresponds to the set $\{1\}$.

Algorithm 2.3 is described in terms of the set T. It could alternatively be presented in terms of the binary vector A, if desired. The operation "Δ", which denotes the symmetric difference of two sets, was defined earlier.

Algorithm 2.3: GRAYCODESUCCESSOR (n, T)

if $|T|$ is even
  then $U \leftarrow T\Delta\{n\}$
  else
    $j \leftarrow n$
    while $j \notin T$ **and** $j > 0$
      do $j \leftarrow j - 1$
    if $j = 1$
      then return ("undefined")
    $U \leftarrow T\Delta\{j - 1\}$
    return (U)

The following theorem can be proved by induction on n.

THEOREM 2.2 *Algorithm 2.3 computes the function* successor *for the Gray code G^n.*

We now proceed to develop ranking and unranking algorithms for the binary reflected Gray code. These algorithms depend on certain relationships between the binary representations of the integers $r = 0, \ldots, 2^n - 1$ and the corresponding vectors G^n_r. Let us begin by tabulating these in the case $n = 3$:

r	binary representation of r	G^3_r
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

For an integer r such that $0 \leq r \leq 2^n - 1$, suppose that its binary representation is written as

$$b_n b_{n-1} \ldots b_1 b_0.$$

In other words,

$$r = \sum_{i=0}^{n} b_i 2^i,$$

and $b_n = 0$ since $r \leq 2^n - 1$. Also, suppose we write the vector G^n_r in the form

$$\mathsf{G}^n_r = a_{n-1} \ldots a_1 a_0,$$

as in Algorithm 2.3.

The relations in the next lemma will form the basis of the ranking and unranking algorithms.