Ordering the Boolean Cube Vectors by their Weights and with Minimal Change

Valentin Bakoev

"St. Cyril and St. Methodius" University,
Veliko Tarnovo, Bulgaria

9th International Conference on Algebraic Informatics (CAI 2022)

27–29 October 2022, Aristotle University of Thessaloniki, Greece

Outline of the talk

1. Introduction

The topic of generating all subsets of a given set occupies an important place in many of the famous books on combinatorial algorithms.

Many algorithms for generating all subsets use the binary representation of integers in n bits (i.e., binary vectors, bit strings, binary words of length n) as characteristic vectors of the subsets of an n-element set.

These algorithms, like many other combinatorial algorithms, are of two main types, depending on the order in which the binary vectors are generated: **in lexicographic order** or **in Gray code order**.

1. Introduction

There are problems where different arrangements of the binary vectors must be studied and used. For example, such a problem is: "*Given a Boolean function f of n variables by its truth table vector. Find (if exists) a vector* $\alpha \in \{0,1\}^n$ *of maximal (or minimal) weight, such that* $f(\alpha) = 1$.". This problem:

1. is discussed in

   

   Bakoev V.: Some problems and algorithms related to the weight order relation on the *n*-dimensional Boolean cube. *Discrete Mathematics, Algorithms and Applications* **13** (3), 2150021 (23 pages) (2021),

   where we define and study the **Weight-Lexicographic Ordering** (WLO), where the vectors of $\{0,1\}^n$ are sorted first by their (Hamming) weights, and then lexicographically.

1. Introduction

2. is closely related to the **problem of computing the algebraic degree of Boolean functions** which is an important cryptographic parameter. An efficient solution to this problem (that includes WLO, ANFT etc.) is proposed in

Bakoev V.: A Method for Fast Computing the Algebraic Degree of Boolean Functions. In: CompSysTech '20, Proc. of the 21st Intern. Conf. on Computer Systems and Technologies, pp. 141–147, Ruse, Bulgaria, June 19–20 (2020).

Here we present a continuation of the first paper. We use a similar approach to obtain another weight ordering of the vectors of $\{0,1\}^n$, where they are ordered **first** by their weights and **second** – every two consecutive vectors of the same weight differ in exactly two coordinates.

1. Introduction

So, the vectors of equal weight are sorted in the same way as by the well-known **revolving door algorithm** used to generate combinations with minimal change.

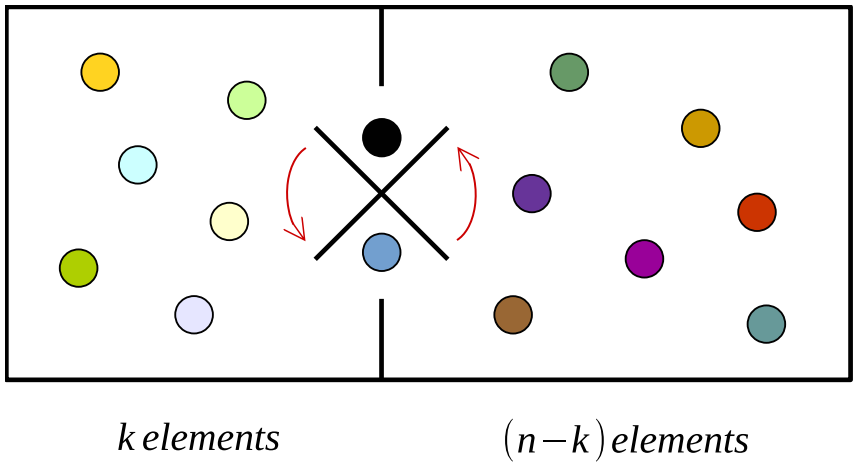

Figure 1: Illustration of the revolving door algorithm

1. Introduction

We propose an algorithm that:

- performs only integer additions and the binary representation of these integers forms the ordering under consideration;

1. Introduction

We propose an algorithm that:

- performs only integer additions and the binary representation of these integers forms the ordering under consideration;
- provides an alternative way to sort the vectors to be tested when computing the algebraic degree of Boolean functions;

1. Introduction

We propose an algorithm that:

- performs only integer additions and the binary representation of these integers forms the ordering under consideration;
- provides an alternative way to sort the vectors to be tested when computing the algebraic degree of Boolean functions;
- can be considered as a **non-recursive extension** of the revolving door algorithm because it generates **all subsets** of a given set in the order discussed;

1. Introduction

We propose an algorithm that:

- performs only integer additions and the binary representation of these integers forms the ordering under consideration;
- provides an alternative way to sort the vectors to be tested when computing the algebraic degree of Boolean functions;
- can be considered as a **non-recursive extension** of the revolving door algorithm because it generates **all subsets** of a given set in the order discussed;
- was used in the creation of sequence A351939 in the Online Encyclopedia of Integer Sequences (OEIS), https://oeis.org/.

2. Basic notions

An *n-dimensional Boolean cube*: $\{0,1\}^n = \{(x_1, x_2, \ldots, x_n) | x_i \in \{0,1\}$, for $i = 1, 2, \ldots, n\}$ and so $|\{0,1\}^n| = |\{0,1\}|^n = 2^n$.

Let $\alpha = (a_1, a_2, \ldots, a_n)$ and $\beta = (b_1, b_2, \ldots, b_n)$ be arbitrary vectors of $\{0,1\}^n$. Then:

- a *serial number* of α is the natural number $\#\alpha = \sum_{i=1}^{n} a_i.2^{n-i}$.

2. Basic notions

An *n-dimensional Boolean cube*: $\{0,1\}^n = \{(x_1, x_2, \ldots, x_n) | x_i \in \{0,1\}$, for $i = 1, 2, \ldots, n\}$ and so $|\{0,1\}^n| = |\{0,1\}|^n = 2^n$.

Let $\alpha = (a_1, a_2, \ldots, a_n)$ and $\beta = (b_1, b_2, \ldots, b_n)$ be arbitrary vectors of $\{0,1\}^n$. Then:

- a *serial number* of α is the natural number $\#\alpha = \sum_{i=1}^{n} a_i.2^{n-i}$.
- a *Hamming distance* between α and β is the natural number $d(\alpha, \beta)$ equal to the number of coordinates in which α and β differ.

2. Basic notions

An *n-dimensional Boolean cube*: $\{0,1\}^n = \{(x_1, x_2, \ldots, x_n) | x_i \in \{0,1\}$, for $i = 1, 2, \ldots, n\}$ and so $|\{0,1\}^n| = |\{0,1\}|^n = 2^n$.

Let $\alpha = (a_1, a_2, \ldots, a_n)$ and $\beta = (b_1, b_2, \ldots, b_n)$ be arbitrary vectors of $\{0,1\}^n$. Then:

- a *serial number* of α is the natural number $\#\alpha = \sum_{i=1}^{n} a_i.2^{n-i}$.
- a *Hamming distance* between α and β is the natural number $d(\alpha, \beta)$ equal to the number of coordinates in which α and β differ.
- α *lexicographically precedes* β, written as $\alpha \leq \beta$ when $\alpha = \beta$ or if $\exists\, i, 0 \leq i < n$, such that $a_1 = b_1, a_2 = b_2, \ldots, a_i = b_i$, but $a_{i+1} < b_{i+1}$. The corresponding *lexicographic ordering* relation $R_{\leq}$ over $\{0,1\}^n$ is a *total ordering* relation in $\{0,1\}^n$.

2. Basic notions

For an arbitrary $k \in \mathbb{N}, k \leq n$, the subset $L_{n,k} = \{\alpha | \, \alpha \in \{0,1\}^n : wt(\alpha) = k\}$ (of all n-dimensional binary vectors of weight k) is called a *kth layer* of $\{0,1\}^n$. So $|L_{n,k}| = \binom{n}{k}$, for $k = 0, 1, \ldots, n$.

2. Basic notions

For an arbitrary $k \in \mathbb{N}, k \leq n$, the subset $L_{n,k} = \{\alpha | \, \alpha \in \{0,1\}^n : wt(\alpha) = k\}$ (of all n-dimensional binary vectors of weight k) is called a *kth layer* of $\{0,1\}^n$. So $|L_{n,k}| = \binom{n}{k}$, for $k = 0, 1, \ldots, n$.
The family of all layers $L_n = \{L_{n,0}, L_{n,1}, \ldots, L_{n,n}\}$ is a **partition** of the n-dimensional Boolean cube into layers and then

$$\sum_{k=0}^{n} |L_{n,k}| = \sum_{k=0}^{n} \binom{n}{k} = 2^n = |\{0,1\}^n| \, .$$

2. Basic notions

For an arbitrary $k \in \mathbb{N}, k \leq n$, the subset $L_{n,k} = \{\alpha | \, \alpha \in \{0,1\}^n : wt(\alpha) = k\}$ (of all n-dimensional binary vectors of weight k) is called a *kth layer* of $\{0,1\}^n$. So $|L_{n,k}| = \binom{n}{k}$, for $k = 0, 1, \ldots, n$.
The family of all layers $L_n = \{L_{n,0}, L_{n,1}, \ldots, L_{n,n}\}$ is a **partition** of the n-dimensional Boolean cube into layers and then

$$\sum_{k=0}^{n} |L_{n,k}| = \sum_{k=0}^{n} \binom{n}{k} = 2^n = |\{0,1\}^n| \, .$$

The *sequence of layers* $L_{n,0}, L_{n,1}, \ldots, L_{n,n}$ defines an **order** of the vectors of $\{0,1\}^n$ according to their weights as follows: if $\alpha, \beta \in \{0,1\}^n$ and $wt(\alpha) < wt(\beta)$, then α precedes β in the sequence of layers, and when $wt(\alpha) = wt(\beta) = k$, then $\alpha, \beta \in L_{n,k}$ and there is no precedence between them.

2. Basic notions

The corresponding relation $R_{<_{wt}}$ is called *precedes by weight*.
It is defined as we said: for arbitrary $\alpha, \beta \in \{0,1\}^n$, $(\alpha, \beta) \in R_{<_{wt}}$ if $wt(\alpha) < wt(\beta)$. We set $(\alpha, \alpha) \in R_{<_{wt}}$ so that $R_{<_{wt}}$ is reflexive. So $R_{<_{wt}}$ is a *partial ordering* (but **not a total ordering**) in $\{0,1\}^n$ and we refer to it as a *Weight Ordering* (WO).

2. Basic notions

The corresponding relation $R_{<_{wt}}$ is called *precedes by weight*.
It is defined as we said: for arbitrary $\alpha, \beta \in \{0,1\}^n$, $(\alpha, \beta) \in R_{<_{wt}}$ if $wt(\alpha) < wt(\beta)$. We set $(\alpha, \alpha) \in R_{<_{wt}}$ so that $R_{<_{wt}}$ is reflexive. So $R_{<_{wt}}$ is a *partial ordering* (but **not a total ordering**) in $\{0,1\}^n$ and we refer to it as a *Weight Ordering* (WO).

Since the vectors of $L_{n,k}$ can be arranged in $\binom{n}{k}!$ ways, for $k = 0, 1, \ldots, n$, there are $\prod_{k=0}^{n} \binom{n}{k}!$ weight orderings of the vectors of $\{0,1\}^n$ – see A051459 in the OEIS.

2. Basic notions

The corresponding relation $R_{<_{wt}}$ is called *precedes by weight*.
It is defined as we said: for arbitrary $\alpha, \beta \in \{0,1\}^n$, $(\alpha, \beta) \in R_{<_{wt}}$ if $wt(\alpha) < wt(\beta)$. We set $(\alpha, \alpha) \in R_{<_{wt}}$ so that $R_{<_{wt}}$ is reflexive. So $R_{<_{wt}}$ is a *partial ordering* (but **not a total ordering**) in $\{0,1\}^n$ and we refer to it as a *Weight Ordering* (WO).

Since the vectors of $L_{n,k}$ can be arranged in $\binom{n}{k}!$ ways, for $k = 0, 1, \ldots, n$, there are $\prod_{k=0}^{n} \binom{n}{k}!$ weight orderings of the vectors of $\{0,1\}^n$ – see A051459 in the OEIS.

The most important among them are those in which the **lexicographic order** or the **order obtained by minimal change** is chosen as a second criterion for ordering all vectors of equal weights.

2. Basic notions

The corresponding relation $R_{<_{wt}}$ is called *precedes by weight*.
It is defined as we said: for arbitrary $\alpha, \beta \in \{0,1\}^n$, $(\alpha, \beta) \in R_{<_{wt}}$ if $wt(\alpha) < wt(\beta)$. We set $(\alpha, \alpha) \in R_{<_{wt}}$ so that $R_{<_{wt}}$ is reflexive. So $R_{<_{wt}}$ is a *partial ordering* (but **not a total ordering**) in $\{0,1\}^n$ and we refer to it as a *Weight Ordering* (WO).

Since the vectors of $L_{n,k}$ can be arranged in $\binom{n}{k}!$ ways, for $k = 0, 1, \ldots, n$, there are $\prod_{k=0}^{n} \binom{n}{k}!$ weight orderings of the vectors of $\{0,1\}^n$ – see A051459 in the OEIS.

The most important among them are those in which the **lexicographic order** or the **order obtained by minimal change** is chosen as a second criterion for ordering all vectors of equal weights.

Hereinafter, the term *Gray code* means binary reflected cyclic Gray code.

Definition 1

1) The vectors (0) and (1) of $\{0,1\}^1$ are in Gray code.
2) Let $\alpha_0, \alpha_1, \ldots, \alpha_{2^{n-1}-1}$ be the sequence of all vectors of $\{0,1\}^{n-1}$ in Gray code.
3) We perform two basic steps to obtain the vectors of $\{0,1\}^n$ in Gray code. First, we take the sequence of vectors $\alpha_0, \alpha_1, \ldots, \alpha_{2^{n-1}-1}$ of $\{0,1\}^{n-1}$ in Gray code and prefix 0 to each of its vectors. Second, we take the same sequence in *reverse order*, i.e., $\alpha_{2^{n-1}-1}, \ldots, \alpha_1, \alpha_0$ – it is obtained by *reflecting* the original sequence. We then prefix 1 to each vector of the reflected sequence. The resulting sequence

$$(0,\alpha_0),(0,\alpha_1),\ldots,(0,\alpha_{2^{n-1}-2}),(0,\alpha_{2^{n-1}-1}),$$
$$(1,\alpha_{2^{n-1}-1}),(1,\alpha_{2^{n-1}-2}),\ldots,(1,\alpha_1),(1,\alpha_0)$$

contains all vectors of $\{0,1\}^n$ ordered in *Gray code*.

3. Ordering the vectors of $\{0,1\}^n$ by their weights and with minimal change

Theorem 2 (beginning)

Let the vectors of $\{0,1\}^n$ be obtained according to Definition 1. Then:
1) The serial numbers of the vectors form the sequence of natural numbers $s(n)$ defined as:

$$s(n) = \begin{cases} 0, 1, \ \textit{if } n = 1, \\ s(n-1), s^r(n-1) + 2^{n-1}, \ \textit{if } n > 1. \end{cases}$$

*Thus, $s(n)$ is a **concatenation** of two sequences: $s(n-1)$ and $s^r(n-1) + 2^{n-1}$, where $s^r(n-1) + 2^{n-1}$ means the same sequence $s(n-1)$, but taken in **reverse order** and each of its terms is incremented by 2^{n-1}.*

3. Ordering the vectors of the Boolean cube by their weights and with minimal change

Theorem 2 (continuation)

2) The weights of the vectors form the sequence of natural numbers $w(n)$ defined as:

$$w(n) = \begin{cases} 0, 1, \ \textit{if } n = 1, \\ w(n-1), w^r(n-1) + 1, \ \textit{if } n > 1. \end{cases}$$

*Analogously, $w(n)$ is a **concatenation** of two sequences: $w(n-1)$ and $w^r(n-1)+1$, where $w^r(n-1)+1$ means the sequence $w(n-1)$, taken in **reverse order** and each of its terms is incremented by* 1.

3. Ordering the vectors of the Boolean cube by their weights and with minimal change

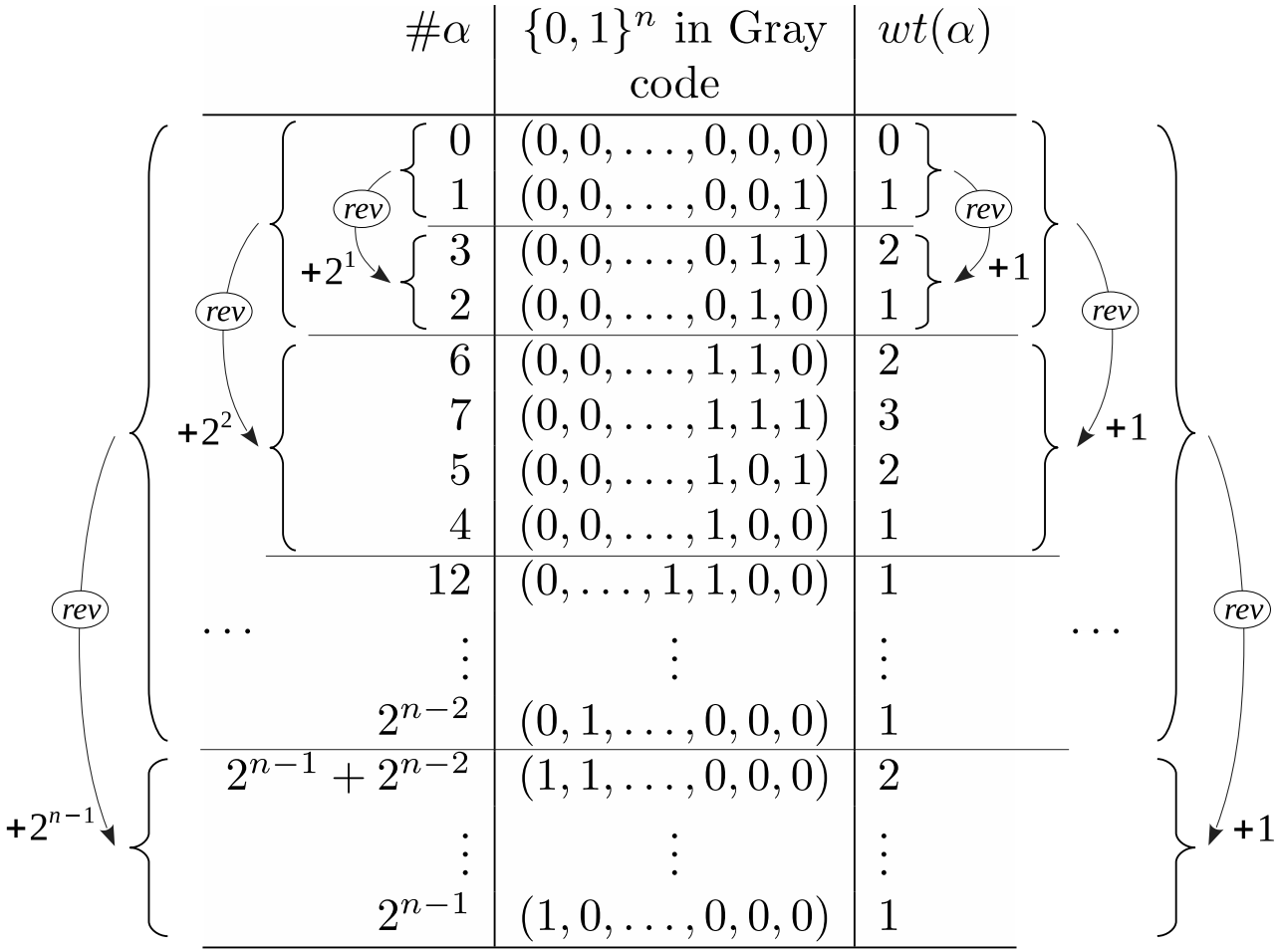

Figure 2: Illustration of the statements of Theorem 2: the sequences $s(n)$ and $w(n)$ in the left and right columns, respectively.

3. Ordering the vectors of the Boolean cube by their weights and with minimal change

Let $L_{n,k} = \{\alpha_1, \alpha_1, \ldots, \alpha_m\}$ be an arbitrary layer of $\{0,1\}^n$ – hence $m = \binom{n}{k}$.
We denote by $g_{n,k} = \#\alpha_1, \#\alpha_2, \ldots, \#\alpha_m$ the *sequence of serial numbers*, corresponding to the vectors of $L_{n,k}$, for $0 \leq k \leq n$.

3. Ordering the vectors of the Boolean cube by their weights and with minimal change

Let $L_{n,k} = \{\alpha_1, \alpha_1, \ldots, \alpha_m\}$ be an arbitrary layer of $\{0,1\}^n$ – hence $m = \binom{n}{k}$.
We denote by $g_{n,k} = \#\alpha_1, \#\alpha_2, \ldots, \#\alpha_m$ the *sequence of serial numbers*, corresponding to the vectors of $L_{n,k}$, for $0 \leq k \leq n$.

Then $G_n = g_{n,0}, g_{n,1}, \ldots, g_{n,k}, \ldots, g_{n,n}$ means the sequence of natural numbers representing the vectors of $\{0,1\}^n$ in this WO. Since $g_{n,k}$ contains $\binom{n}{k}$ terms, for $k = 0, 1, \ldots, n$, hence G_n has $\sum_{k=0}^{n} \binom{n}{k} = 2^n$ terms.

Following Definition 1 and Theorem 2, we define G_n inductively as follows:

3. Ordering the vectors of the Boolean cube by their weights and with minimal change

Definition 3

1) The WO sequence of $\{0,1\}^1$ is $G_1 = g_{1,0}, g_{1,1} = 0, 1$.
2) Let $G_{n-1} = g_{n-1,0}, g_{n-1,1}, \ldots, g_{n-1,n-1}$ be the WO sequence of $\{0,1\}^{n-1}$.
3) The WO sequence of n-dimensional Boolean cube $G_n = g_{n,0}, g_{n,1}, \ldots, g_{n,n}$ is defined as:

- $g_{n,0} = 0$ (it represents the WO of the layer $L_{n,0} = \{\vec{0}_n\}$);
- $g_{n,n} = 2^n - 1$ (it represents the WO of the layer $L_{n,n} = \{\vec{1}_n\}$);
- $g_{n,k} = g_{n-1,k}, g^r_{n-1,k-1} + 2^{n-1}$, for $k = 1, 2, \ldots, n-1$. Thus $g_{n,k}$ is a **concatenation** of two sequences: first the sequence $g_{n-1,k}$ is taken (or copied) and then the sequence $g^r_{n-1,k-1} + 2^{n-1}$ is added. Here $g^r_{n-1,k-1}$ means the terms of $g_{n-1,k-1}$ in **reverse order**. The sequence $g_{n,k}$ represents the WO of the layer $L_{n,k}$.

3. Ordering the vectors of the Boolean cube by their weights and with minimal change

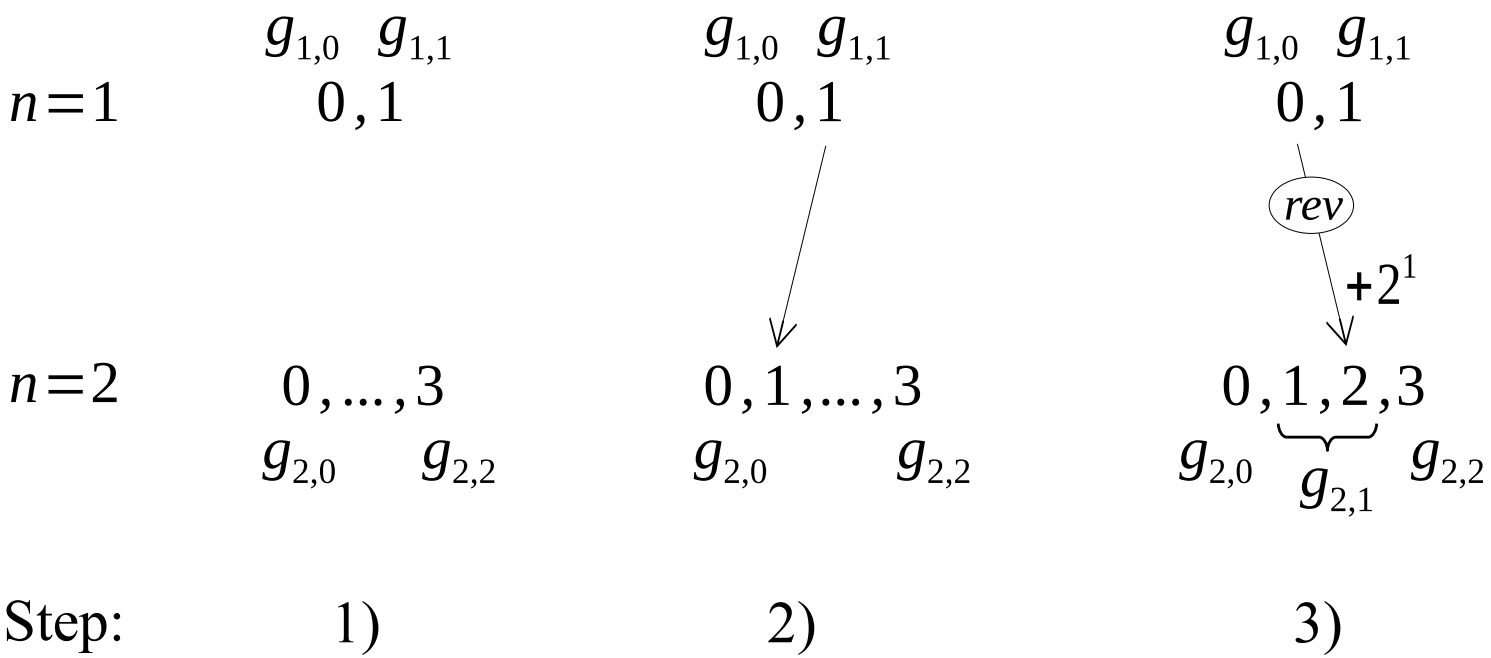

Figure 3: How the sequence G_2 is derived from G_1

3. Ordering the vectors of the Boolean cube by their weights and with minimal change

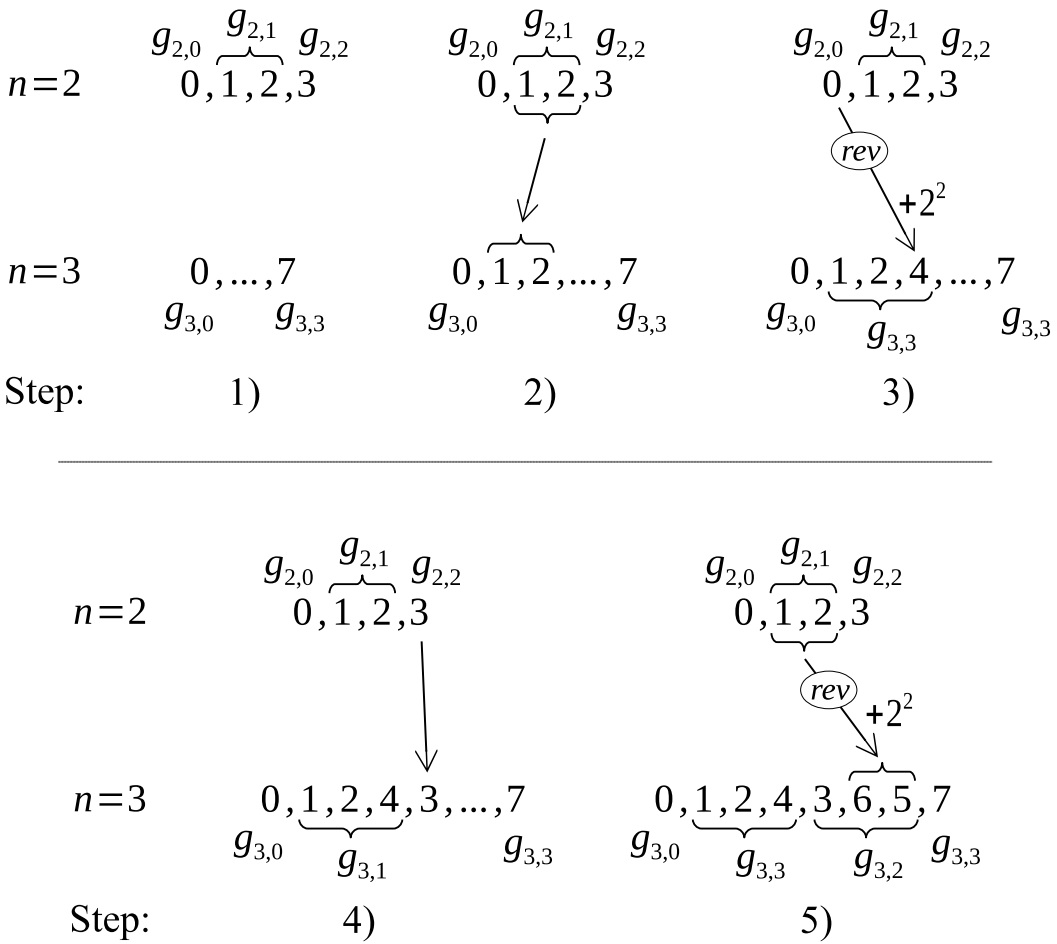

Figure 4: How the sequence G_3 is derived from G_2

3. Ordering the vectors of the Boolean cube by their weights and with minimal change

Definition 4

The sequence of vectors of equal weights $\alpha_1, \alpha_2, \ldots, \alpha_m$ is in a *minimal change ordering*, if every two consecutive vectors differ in exactly two coordinates, i.e., α_{i+1} is obtained from α_i by inverting exactly two of its coordinates, for $i = 1, 2, \ldots, m-1$.
If α_1 and α_m differ in exactly two coordinates too, the sequence is in a *cyclic minimal change ordering*.

3. Ordering the vectors of the Boolean cube by their weights and with minimal change

Theorem 5

Let $G_n = g_{n,0}, g_{n,1}, \ldots, g_{n,n}$ be the WO sequence of $\{0,1\}^n$ obtained according to Definition 3 for a given integer $n \in \mathbb{N}^+$. Then the vectors corresponding to:
(1) the first terms of $g_{n,k}$ and $g_{n,k+1}$ are adjacent vectors;
(2) the last terms of $g_{n,k}$ and $g_{n,k+1}$ are also adjacent vectors, for $k = 0, 1, \ldots, n-1$.
In addition, (3) the vectors corresponding to the terms of $g_{n,k}$ are in a cyclic minimal change ordering, for $k = 1, 2, \ldots, n-1$.
Then G_n defines ordering the Boolean cube vectors first by their weights and second with minimal change.

4. Algorithm for generating the vectors of the Boolean cube ordered by weights and with minimal change

Following Definition 3, we developed an algorithm called **algorithm A**. For a given integer $n \in \mathbb{N}^+$, it sequentially generates all sequences $G_1, G_2, \ldots, G_n$. It performs **3 main steps** in which it computes:

1. **The binomial coefficients** $\binom{m}{k}$, i.e., the lengths of the subsequences $g_{m,k}$, for $m = 1, 2, \ldots, n$ and for $k = 0, 1, \ldots, m$. So it fills the rows numbered by $0, 1, \ldots, n$ of Pascal's triangle in a two-dimensional array denoted by `P_t`.

4. Algorithm for generating the vectors of the Boolean cube ordered by weights and with minimal change

Following Definition 3, we developed an algorithm called **algorithm A**. For a given integer $n \in \mathbb{N}^+$, it sequentially generates all sequences $G_1, G_2, \ldots, G_n$. It performs **3 main steps** in which it computes:

1. **The binomial coefficients** $\binom{m}{k}$, i.e., the lengths of the subsequences $g_{m,k}$, for $m = 1, 2, \ldots, n$ and for $k = 0, 1, \ldots, m$. So it fills the rows numbered by $0, 1, \ldots, n$ of Pascal's triangle in a two-dimensional array denoted by `P_t`.
2. **The places (indices)** where each subsequence $g_{m,k}$ of G_m starts, for $m = 1, 2, \ldots, n$ and for $k = 0, 1, \ldots, m$. The results are stored in a two-dimensional array called `ss_beg`.

4. Algorithm for generating the vectors of the Boolean cube ordered by weights and with minimal change

Following Definition 3, we developed an algorithm called **algorithm A**. For a given integer $n \in \mathbb{N}^+$, it sequentially generates all sequences $G_1, G_2, \ldots, G_n$. It performs **3 main steps** in which it computes:

1. **The binomial coefficients** $\binom{m}{k}$, i.e., the lengths of the subsequences $g_{m,k}$, for $m = 1, 2, \ldots, n$ and for $k = 0, 1, \ldots, m$. So it fills the rows numbered by $0, 1, \ldots, n$ of Pascal's triangle in a two-dimensional array denoted by `P_t`.
2. **The places (indices)** where each subsequence $g_{m,k}$ of G_m starts, for $m = 1, 2, \ldots, n$ and for $k = 0, 1, \ldots, m$. The results are stored in a two-dimensional array called `ss_beg`.
3. **The terms** of the sequences G_m, for $m = 1, 2, \ldots, n$ and fills them in a two-dimensional array called `g`.

4. Algorithm for generating the vectors of the Boolean cube ordered by weights and with minimal change

The code (in C) of a function that implements the third main step is:

```
typedef unsigned long long ull;
// ...
void fill_g_seqs (int n) {
  g[1][0] = 0; // initialization
  g[1][1] = 1; // of G_1
   ull m = 2; // to be added to subsequences
   for (int row = 2; row <= n; row++) {   // main loop
     g[row][0] = 0; // zero term
      ull k = m + m - 1;
      g_seqs[row][k] = k; // last term
      k = 1; // number of the serial term
      for (int col = 1; col < row; col++) {
        // Preparing to copy
        ull seq_len = P_t[row-1][col];
        ull seq_beg = ss_beg[row-1][col];
```

4. Algorithm for generating the vectors of the Boolean cube ordered by weights and with minimal change

```
        // and copying a subsequence
        for (ull j = 0; j < seq_len; j++)
          g[row][k++] = g[row-1][seq_beg+j];
        // Preparing for reversing and ...
        seq_len = P_t[row-1][col-1];
        seq_beg = ss_beg[row-1][col-1];
        // reverse and add m to a subsequence
        for (ull j= seq_len-1; j >= 0; j--)
          g[row][k++]= g[row-1][seq_beg+j] + m;
      }
      m += m; //or m *= 2; - for the next row
    }
}
```

4. Algorithm for generating the vectors of the Boolean cube ordered by weights and with minimal change

The **correctness** of Algorithm A follows from Theorem 5, given definitions and comments.
When $n \leq 64$, then unsigned integers are represented in one 64-bit (or less-bit) computer word. Then:

- The **time complexity** of the algorithm is $\Theta(2^n)$.
- The **space complexity** is $\Theta(n.2^n)$, but it can be reduced to $\Theta(2^n)$.

Therefore, the **average cost** of generating a single integer is $\Theta(1)$.

4. Algorithm for generating the vectors of the Boolean cube ordered by weights and with minimal change

Algorithm A and the given definitions were used in the creation of the sequence A351939 in the OEIS.

Table 1: Results obtained by Algorithm A, for $n = 1, 2, \ldots, 5$.

n	G_n
1	0, 1
2	0, 1, 2, 3
3	0, 1, 2, 4, 3, 6, 5, 7
4	0, 1, 2, 4, 8, 3, 6, 5, 12, 10, 9, 7, 13, 14, 11, 15
5	0, 1, 2, 4, 8, 16, 3, 6, 5, 12, 10, 9, 24, 20, 18, 17, 7, 13, 14, 11, 25, 26, 28, 21, 22, 19, 15, 27, 30, 29, 23, 31

4. Algorithm for generating the vectors of the Boolean cube ordered by weights and with minimal change

Example 6

Let us consider G_4: $\underbrace{0}_{g_{4,0}}, \underbrace{1,2,4,8}_{g_{4,1}}, \underbrace{3,6,5,12,10,9}_{g_{4,2}}, \underbrace{7,13,14,11}_{g_{4,3}}, \underbrace{15}_{g_{4,4}}$.

Table 2: G_4, the characteristic vectors and corresponding subsets of $A = \{a, b, c, d\}$ ordered by cardinalities and with minimal change

G_4	α	$S \subseteq A$	G_4	α	$S \subseteq A$
0	(0,0,0,0)	$\emptyset$	12	(1,1,0,0)	$\{a, b\}$
1	(0,0,0,1)	$\{d\}$	10	(1,0,1,0)	$\{a, c\}$
2	(0,0,1,0)	$\{c\}$	9	(1,0,0,1)	$\{a, d\}$
4	(0,1,0,0)	$\{b\}$	7	(0,1,1,1)	$\{b, c, d\}$
8	(1,0,0,0)	$\{a\}$	13	(1,1,0,1)	$\{a, b, d\}$
3	(0,0,1,1)	$\{c, d\}$	14	(1,1,1,0)	$\{a, b, c\}$
6	(0,1,1,0)	$\{b, c\}$	11	(1,0,1,1)	$\{a, c, d\}$
5	(0,1,0,1)	$\{b, d\}$	15	(1,1,1,1)	$\{a, b, c, d\}$

Conclusions

Algorithm A is of the same type as the algorithm that generates the vectors of the Boolean cube in WLO. Both algorithms:

- can be used to generate all subsets of a given set in the corresponding ordering.

Conclusions

Algorithm A is of the same type as the algorithm that generates the vectors of the Boolean cube in WLO. Both algorithms:

- can be used to generate all subsets of a given set in the corresponding ordering.
- are iterative algorithms, use binary representation of integers and perform only additions of integers when $n \leq 64$.

Conclusions

Algorithm A is of the same type as the algorithm that generates the vectors of the Boolean cube in WLO. Both algorithms:

- can be used to generate all subsets of a given set in the corresponding ordering.
- are iterative algorithms, use binary representation of integers and perform only additions of integers when $n \leq 64$.
- work similarly and have the same type of complexity, which is proportional to the number of integers (vectors, subsets) generated.

Conclusions

Algorithm A is of the same type as the algorithm that generates the vectors of the Boolean cube in WLO. Both algorithms:

- can be used to generate all subsets of a given set in the corresponding ordering.
- are iterative algorithms, use binary representation of integers and perform only additions of integers when $n \leq 64$.
- work similarly and have the same type of complexity, which is proportional to the number of integers (vectors, subsets) generated.

The difference between these orderings is substantial, but the difference between the algorithms is very small, as is the formal difference between the corresponding orderings – if we ignore the reflection in point 3 of Definition 1 (for Gray code), we get a definition of a lexicographic ordering of the Boolean cube vectors.

Thanks

Thank you for your attention!