Greedy generation of some restricted classes of binary words

GASCom 2024

Laboratoire d'Informatique de Bourgogne & Macao Polytechnic University

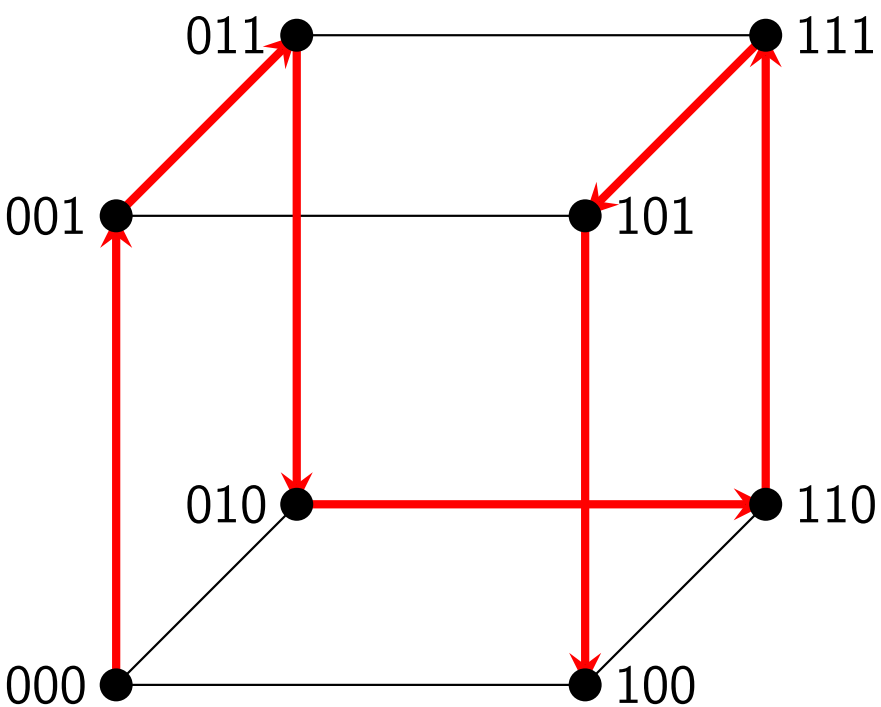
Nathanaël Hassler, Vincent Vajnovszki, Dennis Wong

Overview

The greedy Gray code algorithm

A Gray code for a class of combinatorial objects is a list that contains each object from the class exactly once, such that any two consecutive objects in the list differ only by a 'small change'.
[Torsten Mütze, Combinatorial Gray codes – an updated survey, 2023]

Gray codes



$00\overline{0}$
$0\overline{0}1$
$01\overline{1}$
$\overline{0}10$
$11\overline{0}$
$1\overline{1}1$
$10\overline{1}$
100

Homogeneous transposition

0100110000101

Homogeneous transposition

0100110000101

Non homogeneous transposition

Homogeneous transposition

0100110000101 — Homogeneous transposition

0100110000101 — Non homogeneous transposition

Let S be a set of same length and same weight binary words.

Definition

A homogeneous Gray code for S is a list containing every words of S, such that two consecutive words differ by a homogeneous transposition.

The greedy Gray code algorithm

S : set of same length and same weight binary words

Algorithm

1. Initialize the list $\mathcal{L}$ with a particular word in S.
2. For the last word in $\mathcal{L}$, homogeneously transposes the leftmost possible 1 with the leftmost possible 0, such that the obtained word is in S but not in $\mathcal{L}$.
3. If at point 2. a new word is obtained, then append it to the list $\mathcal{L}$ and return to point 2.

This definition is a specialisation of that introduced in
[Aaron Williams, The greedy Gray code algorithm, 2013]

Example 1

Let S be the set of length four binary words with two 1's.

1001

Example 1

Let S be the set of length four binary words with two 1's.

1001

Example 1

Let S be the set of length four binary words with two 1's.

1001

0101

Example 1

Let S be the set of length four binary words with two 1's.

1001

0101

Example 1

Let S be the set of length four binary words with two 1's.

$$1001$$
$$0101$$
$$0011$$

Example 1

Let S be the set of length four binary words with two 1's.

$$1001$$
$$0101$$
$$0011$$
$$\mathcal{L} = [1001, 0101, 0011].$$

Example 2

0011

Example 2

0011

Example 2

0011

1001

Example 2

0011

1001

Example 2

0011

1001

0101

Example 2

0011

1001

0101

Example 2

0011

1001

0101

0110

Example 2

0011

1001

0101

0110

Example 2

0011

1001

0101

0110

1010

Example 2

0011

1001

0101

0110

1010

Example 2

0011

1001

0101

0110

1010

1100

Example 2

0011
1001
0101
0110
1010
1100

Questions

1. Which classes of binary words this algorithm generates ?

2. Which first words generate the whole class ?

Fibonacci words

Definition

Let $F_n(2,k)$ be the set of length n and weight k binary words that do not have two consecutive 1's.

$$|F_n(2,k)| = \binom{n-k+1}{k}.$$

Example:
$F_5(2,2) = \{10100, 01010, 00101, 10010, 01001, 10001\}$.

Fibonacci words and the greedy algorithm

Definition

For $\alpha \in F_n(2,k)$, we denote by $\mathcal{F}(\alpha)$ the list obtained by applying the greedy algorithm for $F_n(2,k)$, starting with α.

Example : $\mathcal{F}(01010)$ is the list

$$[01010,\ 10010,\ 10100,\ 10001,\ 01001,\ 00101].$$

Last word in $\mathcal{F}(\alpha)$

Let

$$\gamma_{n,k} := 0^{n-2k}(01)^k \in \mathcal{F}_n(2,k).$$

Lemma
For any $\alpha \in F_n(2,k)$ with $\alpha \neq \gamma_{n,k}$, the last word of $\mathcal{F}(\alpha)$ is $\gamma_{n,k}$.

Some generators of $F_n(2, k)$

Definition

We denote by $GenF(n, k)$ the set of *generators*, i.e. words $\alpha \in F_n(2, k)$ such that $\mathcal{F}(\alpha)$ is a homogeneous Gray code for $F_n(2, k)$.

Remark:
By definition of the greedy algorithm, $\alpha \in GenF(n, k)$ if and only if $\mathcal{F}(\alpha)$ contains every word of $F_n(2, k)$.

Some generators of $F_n(2, k)$

Example : $01010 \in GenF(5, 2)$, while $10010 \notin GenF(5, 2)$.

$\mathcal{F}(01010)$ is the list

$$[01010,\ 10010,\ 10100,\ 10001,\ 01001,\ 00101].$$

$\mathcal{F}(10010)$ is the list

$$[10010,\ 01010,\ 01001,\ 10001,\ 00101].$$

Some generators of $F_n(2, k)$

For n, k with $n \geq 2k$, let $\alpha^i_{n,k} = 0^i 1 (01)^{k-1} 0^{n-2k+1-i}$.

Theorem

Let $n \in \mathbb{N}^\star$. Then for all k with $n \geq 2k$, we have

1. $\alpha^i_{n,k} \in GenF(n, k)$ for all $0 \leq i \leq n - 2k$, and $\mathcal{F}(\alpha^i_{n,k})$ is a suffix partitioned list, with $\gamma_{n,k}$ as last word.
2. $\gamma_{n,k} \in GenF(n, k)$, $\mathcal{F}(\gamma_{n,k})$ is a suffix partitioned list and its last word is
   1. $\alpha^0_{n,k}$ if k is even,
   2. $\alpha^{n-2k}_{n,k}$ if k is odd.

$\mathcal{F}(\alpha)$ is suffix partitioned

Lemma
For any $\alpha \in F_n(2, k)$, $\mathcal{F}(\alpha)$ is a suffix partitioned list.

The generators of $F_n(2, k)$

Theorem

Let $n \geq 2k - 1$. Then

$$GenF(n, k) = \{0^i 1(01)^{k-1} 0^{n-2k+1-i} \mid 0 \leq i \leq n - 2k + 1\}.$$

In particular, $|GenF(n, k)| = n - 2k + 2$.

Generalised Dyck prefixes

Definition

Let $C_n(p,k)$ be the set of length n and weight k binary words with the property that any prefix contains at least p times as many 0's as 1's.

$$|C_n(p,k)| = \binom{n}{k} - p\binom{n}{k-1}.$$

Example:

- $C_n(0,k)$ is the set of length n binary words of weight k,
- $C_{2n}(1,n)$ is the set of length $2n$ Dyck words,
- $C_{3n}(2,n)$ is in bijection with size $3n$ ternary trees.

The generators of $C_n(p,k)$

Theorem

Let $n \geq (p+1)k$. If $p \geq 1$ then

$$Gen_p(n,k) = \bigcup_{j=1}^{k} \{0^{pj-i}1^{j-1}0^i1(0^p1)^{k-j}0^{n-(p+1)k} \mid 0 \leq i \leq p-1\}$$
$$\cup \{0^i1^k0^{n-i-k} \mid pk+1 \leq i \leq n-k\}.$$

If $p = 0$ then $Gen_0(n,k) = \{0^i1^k0^{n-i-k} \mid 0 \leq i \leq n-k\}$. In particular,

$$|Gen_p(n,k)| = n-k+1-p.$$

Recursive tail partitioned lists

$\mathcal{L}$ is a recursive tail partitioned list if it has the form

$$\mathcal{L} = \mathcal{L}_1 \cdot 01^u, \mathcal{L}_2 \cdot 01^{u+1}, \mathcal{L}_3 \cdot 01^{u+2}, \cdots, \mathcal{L}_{\ell+1} \cdot 01^{u+\ell}$$

or the form

$$\mathcal{L} = \mathcal{L}_1 \cdot 01^{u+\ell}, \mathcal{L}_2 \cdot 01^{u+\ell-1}, \mathcal{L}_3 \cdot 01^{u+\ell-2}, \cdots, \mathcal{L}_{\ell+1} \cdot 01^{u}$$

for some $u, \ell \geq 0$, and each list $\mathcal{L}_i$, is in turn recursive tail partitioned.

Recursive tail partitioned lists

Theorem

If the list $\mathcal{L}$ is a homogeneous and suffix partitioned Gray code for a set of (same length and same weight) binary words, then $\mathcal{L}$ is an r-t partitioned list.

CAT generation for a homogeneous Gray code for $C_n(p,k)$

procedure pref(m, j)
  if $m = (p+1)j$ **then**
    if $p = 0$ **then return**
    end if
    $m \leftarrow m-1\,;\, j \leftarrow j-1$
  end if
  if $S_j < m$ **then** *# Increasing tail*
    for $i = 0$ **to** $j-1$ **do** *# i is the number of 1's in the tail*
      pref$(m-i-1, j-i)$
      $S_{j-i} \leftarrow m-i$
      print(S)
  end if
  if $S_j = m$ **then** *# Decreasing tail*
    for $i = j-1$ **downto** 0 **do** *# i is the number of 1's in the tail*
      $S_{j-i} \leftarrow \max(S_{j-i-1}+1, (p+1)(j-i))$
      print(S)
      pref$(m-i-1, j-i)$
  end if
end procedure

- the main call is pref(n, k), it generates $C_n(p, k)$
- S_i is the position of the ith 1 in the word
- table S and the parameter p are global
- S is initialized as $S_i \leftarrow (p+1)i$ for $1 \leq i \leq k$

The list $\mathcal{D}_1(01010101)$ obtained by the call of pref$(8,4)$ with $p = 1$, which is an homogeneous Gray code for $C_8(1,4)$.

The list $\mathcal{D}_1(01010101)$ obtained by the call of pref$(8,4)$ with $p = 1$, which is an homogeneous Gray code for $C_8(1,4)$.

[01010101, 00101011,
00110101, 00110011,
00101101, 01010011,
01001101, 01000111,
00011101, 00100111,
00011011, 00010111,
01001011, 00001111]

The list $\mathcal{D}_1(01010101)$ obtained by the call of pref$(8,4)$ with $p = 1$, which is an homogeneous Gray code for $C_8(1,4)$.

[01010101, 00101011,
00110101, 00110011,
00101101, 01010011,
01001101, 01000111,
00111101, 00100111,
00011011, 00010111,
01001011, 00001111]

The list $\mathcal{D}_1(01010101)$ obtained by the call of pref$(8,4)$ with $p = 1$, which is an homogeneous Gray code for $C_8(1,4)$.

[01010101, 00110101, 00101101, 01001101, 00011101, 00011011, 01001011, 00101011, 00110011, 01010011, 01000111, 00100111, 00010111, 00001111]

The list $\mathcal{D}_1(01010101)$ obtained by the call of pref$(8,4)$ with $p = 1$, which is an homogeneous Gray code for $C_8(1,4)$.

[01010101, 00101011,
00110101, 00110011,
00101101, 01010011,
01001101, 01000111,
00011101, 00100111,
00011011, 00010111,
01001011, 00001111]

The list $\mathcal{D}_1(01010101)$ obtained by the call of pref$(8,4)$ with $p = 1$, which is an homogeneous Gray code for $C_8(1,4)$.

$$
\begin{aligned}
&[01010101, && 00101011,\\
&\ 00110101, && 00110011,\\
&\ 00101101, && 01010011,\\
&\ 01001101, && 01000111,\\
&\ 00011101, && 00100111,\\
&\ 00011011, && 00010111,\\
&\ 01001011, && 00001111]
\end{aligned}
$$

The list $\mathcal{D}_1(01010101)$ obtained by the call of pref$(8,4)$ with $p = 1$, which is an homogeneous Gray code for $C_8(1,4)$.

[01010101, 00101011,
00110101, 00110011,
00101101, 01010011,
01001101, 01000111,
00011101, 00100111,
00011011, 00010111,
01001011, 00001111]

Algorithm analysis

With a classical complexity analysis, we can obtain the following result

Proposition
The call pref(n, k) generates the homogeneous greedy Gray code for $C_n(p, k)$ efficiently.

See [Frank Ruskey, Combinatorial Generation Book].

Biblio


A. Bultena and F. Ruskey.
An Eades-McKay algorithm for well-formed parentheses strings.
Information Processing Letters, 68 :255–259, 1998.


T. Mütze.
Combinatorial Gray codes – an updated survey.
Electronic Journal of Combinatorics, (Dynamic Survey DS26), 2023.


F. Ruskey.
Combinatorial Generation.
Book in preparation.


A. Williams.
The greedy Gray code algorithm.
In *Algorithms and Data Structures*, page 525–536, Berlin, Heidelberg, 2013.


D. Wong and V. Vajnovszki.
Greedy Gray codes for Dyck words and ballot sequences.
In *COCOON 2023*, 2023.