Algorithm for the generation of complement-free sets*

Dániel Fülöp, Carolin Hannusch

Faculty of Informatics, University of Debrecen, Hungary
fulop.daniel9623@gmail.com
hannusch.carolin@inf.unideb.hu

Submitted: February 6, 2019
Accepted: March 29, 2019
Published online: April 6, 2019

Abstract

We introduce an algorithm for the generation of complement-free sets of binary \( m \)-tuples, where \( m \) is even. We also provide an implementation for this algorithm for \( m = 12 \). Such complement-free sets are needed for the generation of a new class of error-correcting codes, which were introduced by Hannusch and Lakatos. These codes build the fundamental improvement in the cryptographic system of Dömösi, Hannusch and Horváth. Therefore the generation of complement-free sets will be important for cryptographic applications. In the end of the paper we give some interesting facts about complement-free sets as combinatorial objects.

Keywords: algorithmic computation, discrete sets

MSC: 03D32, 97N70

1. Introduction and notation

Let \( m \) be an even number, thus \( m = 2k \) for some \( k \in \mathbb{N} \). Then let \( X \) be the set of all binary \( m \)-tuples with exactly \( k \) pieces of 1-s and \( k \) pieces of 0-s.

Definition 1.1. Let \( x \in X \) be an arbitrary element. Further we denote the whole-1 tuple of length \( m \) by \( 1 \). Then we say that \( 1 - x \) is the complement of \( x \).

*This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.
**Definition 1.2.** Let $Y \subset X$, such that $y \in Y$ implies $1 - y \notin Y$. Then $Y$ is called a complement-free subset of $X$. If $Y$ has order $\frac{1}{2}\binom{m}{k}$, then we say that $Y$ is a maximal complement-free subset.

In this paper, we give an algorithm for generating a maximal complement-free set randomly. Such sets are used in [3] for the construction of self-dual error-correcting codes of length $2^m$ and with minimum distance $2^k$. These codes are called HL-codes and they are used in the cryptographic system of Dömösi, Hannusch and Horváth in [1]. In order to develop an effective implementation of the DHH-cryptosystem [2], it is necessary to generate a complement-free set effectively.

The DHH-cryptosystem is using the HL-code for $m = 12$, therefore we provide an implementation of our algorithm for $m = 12$ in C++ under the following link: https://arato.inf.unideb.hu/hannusch.carolin/alg.cpp

**2. The algorithm**

We fix $m = 2k$.

**Input:** number $l$ with $0 \leq l \leq \frac{1}{2}\binom{m}{k} - 1$

**Output:** maximal complement-free set $Y$

Step 1:

- Let $A$ be the list of all binary $m$-tuples with $k$ pieces of 1-s, where the first coordinate is 1.
- Let $B$ be the list of all binary $m$-tuples where $B[i] = 1 - A[i]$.

Step 2: for $i$ from 1 to $\frac{1}{2}\binom{m}{k} + l - 1 \mod \frac{1}{2}\binom{m}{k}$ do

$i := 0$ or 1 randomly; end for;

Step 3: if $i = 0$ then $Y[i] := A[i]$; else $Y[i] := B[i]$. end for;

Continue Step 2 until $\text{order}(Y) = \frac{1}{2}\binom{m}{k}$.

This algorithm provides one possibility to create a complement-free set. Further research step will be the use of this algorithm (esp. the implementation) in an implementation of the DHH-cryptosystem. A fast algorithm with low memory-need is a necessary part of a competitive DHH-cryptosystem. The provided algorithm generates 100 complement-free sets of order 462 in 2.7 seconds and 1000 complement-free sets of order 462 in 15.8 seconds on Intel(R) Core(TM)2 Duo CPU at 2.93 GHz.
3. Additional facts about complement-free sets

The ordering of the list $A$ in Step 1 of the algorithm introduced in Section 2 should be kept secret. This will improve the security of the algorithm when it is used in Cryptography. For $m = 12$ the list $A$ has 462 elements, which means there are $462!$ possible orders of the elements of $A$ and since

$$462! > 10^{1032},$$

this cannot be brute-forced.

So, let us now assume that $A$ is secret. For the random value of $i$ in Step 2 of the algorithm we need a random generator with almost 50% possibility that if $i = 0$, then $i + 1 = 1$ and vice versa. Applying such a random generator we have a probability of $\left(\frac{1}{2}\right)^{462}$ that we generate the same complement-free set twice. A good random generator can be found e.g. in [4].

Some more interesting things can be investigated in relation to complement-free sets if we have a more detailed look at one set itself. Given a complement-free set $Y$, each element $y \in Y$ consists of $m$ coordinates. We will count the 1-s in a fixed coordinate for all $y \in Y$. For example, let $Y = \{(1, 1, 0, 0), (1, 0, 0, 1), (0, 1, 0, 1)\}$. Then we have two 1-s in each four positions. Thus we will say that $Y$ is of type $(2, 2, 2, 2)$ according to the following definition:

**Definition 3.1.** We say that the complement-free set $Y$ is of type $\nu = (n_1, \ldots, n_m)$, if

$$n_i = \sum_{y \in Y} y_i,$$

i.e. $n_i$ is the number of 1-s in the $i$-th coordinate of all binary strings in $Y$.

**Remark 3.2.** We have $\sum_{i=1}^{m} n_i = k \cdot \frac{1}{2}\left(\begin{array}{c} m \\ k \end{array}\right)$. Let us denote $\sum_{i=1}^{m} n_i$ by $N$. Then it is clear, that if $\nu$ is the type of a complement-free set, then $\nu$ is also a partition of $N$. This statement is not true in the other way, since e.g. for $m = 6$ we have $N = 30$ and $(7, 7, 5, 3, 3, 1)$ is a partition, but there is no complement-free set of such a type.

**Proposition 3.3.** For fix $m = 2k$ there exist at least $\frac{1}{4}\left(\begin{array}{c} m \\ k \end{array}\right) + 1$ different types of complement-free sets.

**Proof.** We may assume $n_1 \geq n_2 \geq \cdots \geq n_m$. Then there exists exactly one type with $n_1 = \frac{1}{2}\left(\begin{array}{c} m \\ k \end{array}\right)$ (namely the complement-free set consists of all elements of the list $A$ in this case). Now imagine, that we change one element of the set from $A[i]$ to $B[i]$. Thus the new complement-free set has type $n_1 = \frac{1}{2}\left(\begin{array}{c} m \\ k \end{array}\right) - 1$. We continue this step until the descending order $n_1 \geq n_2 \geq \cdots \geq n_m$ can be fulfilled. Since $k \cdot \frac{1}{2}\left(\begin{array}{c} m \\ k \end{array}\right)$ is divisible by $m$ there exists exactly one type with $n_1 = \frac{1}{4}\left(\begin{array}{c} m \\ k \end{array}\right)$.  

Computations of all types of complement-free sets for small values of $m$ let us conjecture that the distribution of types with $\frac{1}{4}\left(\begin{array}{c} m \\ k \end{array}\right) \leq n_1 \leq \frac{1}{2}\left(\begin{array}{c} m \\ k \end{array}\right)$ is close to Gaussian distribution. Further, it turns out that computing all types of complement-free
sets for \( m = 8 \) needs a lot of computation and cannot be done fast. Thus we come to the following open problems.

**Problem 3.4.** Determine all types of complement-free sets for fix \( m! \)

**Problem 3.5.** Show the distribution of complement-free sets with respect to the largest value in the type! (Is it Gaussian distribution?)

**References**

[1] P. Dömösi, C. Hannusch, G. Horváth: A cryptographic system based on a new class of binary error-correcting codes, submitted.

[2] P. Dömösi, C. Hannusch, G. Horváth: Public key cryptographic method and apparatus for data encryption and decryption based on error-correcting codes, Budapest: Hungarian Intellectual Property Office, patent application, P1800038, 2018.

[3] C. Hannusch, P. Lakatos: Construction of self-dual binary \([2^{2k}, 2^{2k-1}, 2^k]-codes\), Algebra and Discrete Mathematics 21.1 (2016), pp. 59–68.

[4] T. Herendi: Construction of uniformly distributed linear recurring sequences modulo powers of 2, Uniform distribution theory 13.1 (2018), pp. 109–129, doi: 10.1515/udt-2018-0006.