Counting Sequences, Gray Codes and Lexicodes

Counting Sequences, Gray Codes and Lexicodes

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op maandag 22 mei 2006 om 10.00 uur

door

I Nengah SUPARTA

Master of Science in Mathematics
Bandung Institute of Technology,
Bandung - Indonesia

geboren te Buleleng, Bali - Indonesië

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. A.J. van Zanten

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. A.J. van Zanten,	Universiteit Maastricht, promotor
Prof. dr. S.M. Dodunekov,	Bulgarian Academy of Sciences, Bulgarije
Prof. dr. A.A. Evdokimov,	State University Novosibirsk, Rusland
Prof. dr. H. Aydinian,	Bielefeld University, Duitsland
Prof. dr. H.C.A. van Tilborg,	Technische Universiteit Eindhoven
Prof. dr. ir. C. Roos,	Technische Universiteit Delft
Prof. dr. G.J. Olsder,	Technische Universiteit Delft

Counting sequences, Gray codes and Lexicodes.
Dissertation at Delft University of Technology.

This work has been carried out under the terms of the Memorandum of Understanding between the Ministry of Research and Technology of the Republic of Indonesia and the Ministry of Education, Culture and Science of The Netherlands on Cooperation in the field of Research, Science and Technology.

ISBN 90-8559-176-7

Keywords : Complete graph, counting sequences, Gray codes, greedy algorithm, lexicodes, linearity, maximum counting sequences, self-orthogonality, self-duality, separability, uniform counting sequences.

Printed in The Netherlands by Optima Grafische Communicatie

To my parents, the late **Siwa** and **Nita**,
to my wife **Susantini**, and
to my sons **Ananda K.S.** and **Rama B.S.**

Acknowledgments

It was prof. dr. A.J. van Zanten who showed me a path to open and to enter the gate of Delft University of Technology. He introduced me into the preliminary knowledge of coding theory which constitutes a basic tool for my current research. His willingness to give me an opportunity to do PhD research under his supervision is a great thing. I very much appreciate his patience when guiding me in doing my research. His detailed comments and valuable suggestions were predominant factors for the completion of my thesis. I express my deep gratitude to prof. dr. A.J. van Zanten for his immeasurable support.

I am indebted to the chief of Mathematics Education Department, to the dean of MIPA, and to the rector of IKIP Negeri Singaraja, for giving me permission to be absent for about four years. I am grateful to the Koninklijke Nederlandse Academie van Wetenschappen (KNAW) for their financial support to conduct and complete my PhD research. My appreciations go especially to prof. dr. R.K. Sembiring, dr. E. Tri Baskoro and to drs. Paul Althuis who have shown great flexibility in their assistance to make my stay at Delft University of Technology possible.

My special thanks go to Durk Jellema, Rene Tamboer, and Manon Post, who have been a great help for me to arrange all kinds of administrative and practical things since the beginning of my PhD period.

I owe much gratitude to Ir. X. Wu for always being friendly when assisting me in overcoming computer problems, and to Mrs. Evelyn Sharabi for all her help which always was accompanied by a kind smile. Many thanks also go to Diderich and Veselin for their kindness to share room HB.10.320. I also enjoyed the nice ambiance during the last year of my PhD period with my two nice roommates Alex and Alina. Much of their time was taken by my questions which sometimes may have sounded rather stupid. I also express my appreciation to all other people of floor 5 for sharing all facilities as well as their company.

I would like to express my gratitude to Budi, Julius, Hartono, Loeky and Antony, my housemates at Nassaulaan 149. I apologize for anything which might have bothered them. The same goes for Jacopo, my housemate at Zusterlaan 176.

As for my neighbors in the Nassaulaan: Alfond and family, Jeroen and Annemarieke, and Bill I thank you for your hospitality, and I will miss our chats about the weather, the small garden, traveling, bicycling, and even taxation.

Furthermore, I am indebted to: H. de Groot Heupner - I. Simon, Willem - Ketut, Patrict - Silvia, and Tom - Lina. *Lambaian nyiur, hamparan bukit, laut biru, buah nangka, pisang, durian, nasi goreng, ayam goreng, sayur urap, sayur asem, pecel* etc. were usually hot topics. Our discussions sometimes made us aware of the richness of

the archipelago known as Indonesia, that rich country with its many poor habitants.

My fellow-Indonesians, also those who have not been mentioned here, have generated a real nice atmosphere which made it possible for me not to forget my nationality, since in almost every meeting Bahasa Indonesia was the language of conversation. It was their presence that made my homesickness less painful. I thank Pujianiki for her great support and advice. I especially thank Adi S., Ari S., Ari T., Bib, Ferry, Julfikar, Julius, Jusuf, Nelson, Saoer and Tenov, for all their help they offered and also for the heart-warming conversations we had. Hopefully, there will be many more occasions in the future to talk about the wind which used to blow so strongly in wintertime.

My thanks also go to Dwi R. and Diah C. for their most helpful assistance during the process of finishing my thesis. I also thank Ardie and Yoga for their delicious mixed *bubur kacang hijau* and *durian*.

Finally, I would like to extend my deepest gratitude and all my respect to my late beloved mother, Ni Nengah Nita, and to my late beloved father, I Ketut Siwa. This thesis is a tribute to them. I wish I could have been with them at the moments they really missed me. "I do miss your sincere, cool smiles."
To my dear sons Ananda Kusuma and Rama Bhawana, and to my beloved wife Susantini, I express my deep respect and great appreciations. I apologize for having been a far-away father and husband at a distance of more than 10.000 kilometers, for a period of about four years. During that time we had to bury deeply our desire to talk, to walk, or to have dinner together. At the end, my special thanks go to my brothers, sisters, nephews and nieces, for all their best wishes.

Delft, some day in March 2006,

I N. Suparta
isuparta@yahoo.com

Contents

PART ONE

COUNTING SEQUENCES

And

GRAY CODES

1 Preliminaries

1.1 Introduction

This thesis contains two main topics: Counting sequences and Gray codes, and Lexicodes. The discussion of the first topic is referred to as the first part of this thesis. The second part contains the remaining topic.

Counting sequences have applications in logic-circuits. Sometimes it is desirable to have a counting sequence such that the number of bit changes from one codeword to its successor is as *large* as possible, for example when testing a physical circuit for reliable behavior in worst-case conditions (see e.g. [32, Exercise 67, p. 35]). In particular, balanced counting sequences are of considerable interest in combinatorial logic circuits.

Binary Gray codes which constitute a special type of counting sequences is a well-known topic. Although this type of code has been named after its inventor Frank Gray from Bell Laboratories, the code itself actually was demonstrated already by the French engineer Émile Baudot in 1878 in a telegraph device(cf. [26]). Among all kinds of Gray codes, the binary reflected Gray code, also known as the standard Gray code, is the best known(cf. [59, 86]). This code was a patented invention due to Gray in 1953, and was used to reduce the coding errors in a pulse code communication system [23]. Its name was a tribute to its inventor.

The usefulness of the binary reflected Gray code and its widespread appearance are undisputed, for instance in algebraic coding theory (cf. [84]), in the design of combinatorial algorithms (cf. [59]), while its optimality with respect to various applications has proved itself frequently(cf. [2]). For certain applications however, sometimes additional properties of Gray codes are requested. For instance, when designing experiments, or when designing and testing electrical circuits and informa-

tion systems, balanced Gray codes are needed(cf. [3, 39, 41, 42, 87, 88]), whereas applications of the *N-ary n-cube* can be found in the design of several concurrent computers including the Ametek 2020, the J-Machine, the Mosaic, the iWarp, and the Cray T3D(see [6] and references therein). This constitutes one reason why the topic of N-ary Gray codes is interesting. Moreover, if Q is a power of a prime, Q-ary Gray codes also have applications in determining the weight distribution of a linear code(see [24]).

There are still many other types of Gray codes depending on their application(cf. e.g. [21, 22, 41]). For a more extended survey on Gray codes we refer to [61].

As commencement of the first part, we shall introduce in this chapter some basic definitions and notations with respect to counting sequences, and in particular to Gray codes. For introductory remarks to the second part we refer to the introductions of Chapters 6 and 7.

1.2 Binary counting sequences

Let $\mathcal{O}(n|p)$ be a sequence of p distinct binary n-tuples. We call p the *period* of $\mathcal{O}(n|p)$. If $p = 2^n$, we call the sequence $\mathcal{O}(n|p)$ a *counting sequence* of length n, or shortly a *counting sequence* n, and we denote it by $\mathcal{O}(n)$ (cf. [60]). In Figure 1.1 we show two examples of counting sequences of length 4. In the sequel, binary n-tuples will be called *codewords* of length n. Codewords in a sequence will be denoted by boldface letters like $\mathbf{g}$, $\mathbf{h}$, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{v}$, or $\mathbf{w}$. Furthermore, we shall index codewords in $\mathcal{O}(n|p)$ from 0 until $p-1$. So, the j-th codeword in the sequence $\mathcal{O}(n|p)$ will be denoted for instance by $\mathbf{x}_j$, $0 \leq j < p$. Bit positions will be counted from 1 until n going from right to left. For example, the j-th codeword in a sequence $\mathcal{O}(n|p)$, with respect to its components, is written as $\mathbf{x}_j = x_{jn} \cdots x_{j2} x_{j1}$. If we consider the sequence $\mathcal{O}(n|p)$ as a closed or a cyclic sequence, then we identify the codeword of index p and the one of index 0. As usual, the *Hamming distance* between two codewords $\mathbf{x}$ and $\mathbf{y}$ of the same length, denoted by $d_H(\mathbf{x}, \mathbf{y})$, is defined to be the number of bit positions where they differ. A counting sequence $\mathcal{O}(n)$ of length n which has the property that any two successive codewords in the list, including the last and the first codeword, have the same Hamming distance is called *uniform*. The list shown in Fig. 1.1.a is an example of a uniform counting sequence of length 4 and Hamming distance 1.

The number of changes in bit position i, $1 \leq i \leq n$, in a binary counting sequence $\mathcal{O}(n|p)$, denoted by $TC_{\mathcal{O}(n|p)}(i)$, is called the *transition count* of bit position i in the list of $\mathcal{O}(n|p)$. The transition count distribution $TC_{\mathcal{O}(n|p)} = (TC_{\mathcal{O}(n|p)}(1), TC_{\mathcal{O}(n|p)}(2), ..., TC_{\mathcal{O}(n|p)}(n))$ of a sequence $\mathcal{O}(n|p)$ is called the *transition count spectrum* of $\mathcal{O}(n|p)$. A sequence satisfying the condition that $|TC_{\mathcal{O}(n|p)}(i) - TC_{\mathcal{O}(n|p)}(j)| \leq 2$ is called a *balanced* sequence, and it is called *totally balanced* if $|TC_{\mathcal{O}(n|p)}(i) - TC_{\mathcal{O}(n|p)}(j)| = 0$, for every $1 \leq i, j \leq n$.

It should be mentioned that occasionally authors prefer to consider the distribution $(\frac{TC_{\mathcal{O}(n|p)}(1)}{2^n}, \frac{TC_{\mathcal{O}(n|p)}(2)}{2^n}, ..., \frac{TC_{\mathcal{O}(n|p)}(n)}{2^n})$ instead of $TC_{\mathcal{O}(n|p)}$, which is referred to as the *bit error probabilities* of $\mathcal{O}(n|p)$ (see e.g. [41, 64]).

0000	1100	0000	0110
0001	1101	1111	1001
0011	1111	0001	0111
0010	1110	1110	1000
0110	1010	0011	0101
0111	1011	1100	1010
0101	1001	0010	0100
0100	1000	1101	1011
a.		b.	

Figure 1.1: a. A uniform counting sequence 4; b. a maximum counting sequence 4.

Let $\mathbf{x}_i = x_{in} \cdots x_{i2} x_{i1}$ be the i-th codeword of $\mathcal{O}(n|p)$ and let $s_i := \{j | x_{(i-1)j} \neq x_{ij}\}$. For every i, $1 \leq i \leq p$, s_i is called the *transition* from the codeword $\mathbf{x}_{i-1}$ to $\mathbf{x}_i$. The *transition sequence* $S_{\mathcal{O}(n|p)}$ of a sequence $\mathcal{O}(n|p)$ is an ordered sequence of transitions s_i for all i, $1 \leq i \leq p-1$. If we are dealing with a cyclic counting sequence, the transition sequence of the cyclic counting sequence can be completed with the transition s_p of the last and the first codeword of the sequence. This last transition is occasionally called the *closing transition* of the transition sequence. The transition sequence of a cyclic sequence which is completed with its closing transition is called the *complete* transition sequence, or occasionally, just the transition sequence. A non-complete transition sequence is simply called *non-complete* transition sequence.

In Figure 1.1 sequence a. has complete transition sequence

$$\{1\},\{2\},\{1\},\{3\},\{1\},\{2\},\{1\},\{4\},\{1\},\{2\},\{1\},\{3\},\{1\},\{2\},\{1\},\{4\},$$

whereas sequence b. has complete transition sequence

$$\{1,2,3,4\},\{2,3,4\},\{1,2,3,4\},\{1,3,4\},\{1,2,3,4\},\{2,3,4\},\{1,2,3,4\},\{1,2,4\},$$

$$\{1,2,3,4\},\{2,3,4\},\{1,2,3,4\},\{1,3,4\},\{1,2,3,4\},\{2,3,4\},\{1,2,3,4\},\{1,2,4\}.$$

We shall omit the curly brackets in the transition sequence, when its components are all singleton sets. Counting sequences which have a transition sequence the components of which are all singleton sets are called *Gray codes*. If the closing transition is also a singleton, we speak of a *cyclic* Gray code.

Very often, especially for binary Gray codes, the transition sequence of a counting sequence is a compact tool for studying properties of the sequence (see e.g. [21, 22, 39, 46]). For instance, the transition count of bit position i in a counting sequence is equal to the number of occurrences of the integer i in the transition sequence of the counting sequence. The construction of special Gray codes is often based on the structure of their transition sequences (see e.g. [3, 20, 58, 60, 66, 70, 77]). For

constructing single-error-correcting unit-distance counting codes, Kautz in [29] manipulates the transition sequences of binary standard Gray codes. In [24], the transition sequences of q-ary Gray codes constitute a basis for an algorithm for generating the (Hamming) weight distribution of an arbitrary linear error correcting code over $GF(q)$.

The *average distance* d_A of a sequence $\mathcal{O}(n|p)$ is the average Hamming distance between the p pairs of successive codewords. Here, the successor of $\mathbf{x}_{p-1}$ is $\mathbf{x}_0$. It follows immediately that the following relation exists between the average Hamming distance and the transition counts in any $\mathcal{O}(n|p)$

$$\sum_{i=1}^{n} TC_{\mathcal{O}(n|p)}(i) = p \cdot d_A. \tag{1.1}$$

The following theorem was proved in [60]. Here, we shall give a slightly different proof.

Theorem 1.2.1. *The average distance d_A of a counting sequence $\mathcal{O}(n)$, $n > 1$, is bounded according to $1 \leq d_A \leq n - \frac{1}{2}$.*

Proof. It is clear that the lower bound is sharp, since for a cyclic Gray code d_A is equal to 1. Again, it is clear that for every two n-bit codewords $\mathbf{x}$ and $\mathbf{y}$, we have $d(\mathbf{x}, \mathbf{y}) \leq n$, and the equality occurs only if $\mathbf{x}$ and $\mathbf{y}$ are complementary codewords. It is obvious that there are precisely 2^{n-1} pairs of complementary codewords of length n. Any other pair of codewords has a mutual distance of at most $n - 1$. Hence, in any counting sequence of length n we shall have

$$1 \leq d_A \leq \frac{n2^{n-1} + (n-1)2^{n-1}}{2^n} = n - \frac{1}{2}.$$

□

Later we shall see that for every length n, a closed sequence of length n having average Hamming distance $n - \frac{1}{2}$ exists. Such a sequence will be called a *maximum counting sequence* or shortly *maximum sequence* (cf. [60]). So, the upper bound of d_A is also sharp.

1.3 Binary Gray codes

It is well known that uniform counting sequences exist in which any two successive codewords have Hamming distance 1, including the first and the last codeword. These special counting sequences are called *cyclic Gray codes*(cf. Section 1.2). From now on, any binary sequence of length n with period p, $1 \leq p \leq 2^n$, and such that any two successive codewords in the sequence have Hamming distance 1, will be called a *Gray sequence*. We call a Gray sequence *cyclic*, if also the last codeword of the sequence differs in only one bit from the first codeword.

The binary reflected Gray code, also known as the *standard Gray code* is the best known Gray code (cf. [59, 86]). A characteristic property of the binary standard Gray code is that the second half of the list of codewords can be obtained from the first half by reflection, i.e. by writing the first half upside down and replacing the front zero by one. This property leads to the name of binary *reflected* Gray code. More formally, a binary reflected Gray code of length $n+1$ can be obtained from the binary reflected Gray code of length n according to the following rules. Let $G_{ref}(n)$ be the binary reflected Gray code of length n. Then, the list of the binary reflected Gray code $G_{ref}(n+1)$ of length $n+1$ is obtained by listing all codewords of $G_{ref}(n)$ and adding a prefix 0 to each codeword, followed by reflecting the list of $G_{ref}(n)$ with an additional prefix 1 in front of each codeword. For instance, $G_{ref}(1) = 0, 1$, $G_{ref}(2) = 00, 01, 11, 10$, and $G_{ref}(3) = 000, 001, 011, 010, 110, 111, 101, 100$.

Many researchers have studied binary Gray codes because of a wide range of applications such as circuit testing, experimental designs and signal processing and communication systems (see e.g. [2, 28, 31, 34, 36, 50, 56]), or just for interesting mathematical properties (see e.g. [12, 27, 33]).

In the sequel, when discussing binary Gray codes, we shall simplify some notations. For example, the complete transition sequence of a cyclic Gray code of length n will be denoted by $\bar{S}(n)$, and the notation $S(n)$ will be reserved for the corresponding non-complete transition sequence. Furthermore, the addition between binary codewords in $GF(2)^n$ will be denoted by the symbol $\oplus$.

1.4 N-ary Gray codes

A well-known generalization of binary Gray codes is an *N-ary Gray code*, $N > 0$, of length n. This is an ordered list of all N^n codewords of length n over the set of integers $\{0, 1, 2, ..., N-1\}$, such that each codeword differs from the previous one in exactly one bit position. One applies such codes in analogue-to-digital conversion of data, where the adjacency property of successive codewords reduces both the likelihood and the effect of errors [10]. The natural number N is called the *radix* of the Gray code (cf. [15]). As usual, if the last codeword of the list differs in one position from the first codeword, one speaks of *cyclic* N-ary Gray code. In this case, the Hamming distance of any codeword to its two immediate neighbors in the list is equal to one, where the list is considered to be a cyclic list. More specifically, in a cyclic list one can require that if $\mathbf{x}_i$ is the i-th codeword in the list and $\mathbf{x}_{i+1}$ the (i+1)-st codeword, then one either has $x_{i+1j} = x_{ij} + 1$ or $x_{i+1j} = x_{ij} - 1$, mod N, for all values of i, $0 \leq i \leq N^n - 1$ and all values of j, $1 \leq j \leq n$. Such a code is defined as a *minimal-change* N-ary Gray code. One also could say that codewords which are neighbors in this list are at *Lee distance* 1 from each other(cf. [35],[53, p. 1750]). In this thesis, the term N-ary Gray code applies to this type of cyclic codes.

The *list distance* of two codewords $\mathbf{x}_i$ and $\mathbf{x}_j$, denoted by $d_L(\mathbf{x}_i, \mathbf{x}_j)$, in an N-ary Gray code of length n is defined to be the absolute value of the difference of i and j. So,

$$d_L(\mathbf{x}_i, \mathbf{x}_j) = |j - i|. \tag{1.2}$$

In eq. (1.2) the list of codewords is considered to be linear, i.e. non-cyclic. But in practice, we are quite frequently dealing with a cyclic list. In that case the above definition is not convenient. Therefore, it is natural to introduce the *cyclic* list distance for a cyclic list, which is defined as

$$D(\mathbf{x}_i, \mathbf{x}_j) = min\{|j - i|, N^n - |j - i|\}. \tag{1.3}$$

(cf. also [57]). Throughout this thesis, the terms list and Gray code are interchangeable, and the list of codewords will be written occasionally as a matrix the rows of which are the codewords.

1.4.1 N-ary reflected Gray codes

A well-known N-ary Gray code, generalizing the binary reflected Gray code, is the

000	000	122	200
001	001	121	201
011	002	120	202
010	012	110	212
110	011	111	211
111	010	112	210
101	020	102	220
100	021	101	221
	022	100	222
a.	*b.*		

Figure 1.2: a. Binary reflected Gray code of length 3; b. Ternary reflected Gray code of length 3.

N-ary reflected Gray code $G_{ref}(n, N)$ of length n, which is recursively defined as

$$
\begin{aligned}
G_{ref}(n,N) &= \begin{pmatrix} 0 & G_{ref}(n-1,N) \\ 1 & G_{ref}(n-1,N)^R \\ 2 & G_{ref}(n-1,N) \\ \vdots & \vdots \\ N-1 & G_{ref}(n-1,N)^* \end{pmatrix} \\
G_{ref}(1,N) &= \begin{pmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ N-2 \end{pmatrix}
\end{aligned}
\tag{1.4}
$$

where $G_{ref}(n-1,N)^R$ stands for the list $G_{ref}(n-1,N)$ in reversed ordered. The symbol * in the last row of the matrix $G_{ref}(n,N)$ stands for R only when N is even, otherwise it should be deleted (cf. [15]).

The code $G_{ref}(n,N)$ is also called the *standard N-ary Gray code*. In Fig. 1.2: a. and b. we show the lists of the binary and of the ternary reflected Gray code of length 3. As we can see, the 3-bit binary reflected Gray code is *cyclic*, whereas the ternary one is not. Theorem 1.4.1 formulates this property in general.

The following theorem is obvious(cf. [15]).

Theorem 1.4.1. *For every $n > 1$, the N-ary reflected Gray code $G_{ref}(n,N)$ of length n is cyclic if N is even, and it is non-cyclic otherwise.*

2

Separability in N-ary Gray Codes

In this chapter we shall discuss the so-called separability problem for several types of N-ary Gray codes. This problem roughly deals with the relationship between the Hamming distance between any two codewords and their list distance, expressed by the separability function of the code. We shall derive upper and lower bounds for these functions and compare them with respect to their separability power. In this context we shall introduce a *near optimal* binary Gray code the separability of which is almost optimal.

2.1 The separability of the N-ary reflected Gray code $G_{ref}(n, N)$

In a Gray code, or in any ordered code, a question of theoretical as well as of practical relevance is the following. If two codewords in a code differ in m positions, how far are they separated from each other in the list of codewords? The larger this list distance of the code, the smaller the number of bit errors will be when transmitting codewords by means of analog signals (cf. [86]). Stated more precisely, when we index the codewords in the list from 0 until $2^n - 1$, and if two codewords $\mathbf{x}_i$ and $\mathbf{x}_j$ have Hamming distance $d_H(\mathbf{x}_i, \mathbf{x}_j) = m$, can we find an integral-valued *bounding function* b such that the list distance satisfies $d_L(\mathbf{x}_i, \mathbf{x}_j) \geq b(m)$, for $1 \leq m \leq n$? Of course, the most interesting bounding function is a function giving sharp lower bounds for all values of m, i.e. such that for every m-value there exists at least one pair of codewords with list distance $b(m)$. The question of finding this uniquely determined function is called the separability problem (cf. [90, 86]). We shall use the term *separability function* for a function b - occasionally written as $b(m)$ - yielding sharp lower bounds for $1 \leq m \leq n$. In [90] Yuen solved the separability problem for the binary reflected Gray code. The separability function in this case appears to be $\lceil \frac{2^m}{3} \rceil$. The derivation

of this expression is accomplished by making use of the *index system* of the reflected Gray code, i.e. the relationship between a codeword $\mathbf{g}_i$ and its index i, $0 \leq i \leq 2^n - 1$ (cf. e.g. [59]). Along similar lines, Cavior in [9] derived a sharp upper bound for the list distance in this code, being $2^n - \lceil \frac{2^m}{3} \rceil$, $1 \leq m \leq n$. In both papers the list of codewords is interpreted as a linear (non-cyclic) list, which implies that $d_L(\mathbf{x}_i, \mathbf{x}_j)$ is defined as $|i - j|$. Now, it is well known that the reflected Gray code is a cyclic Gray code(see Section 1.2). With respect to this notion the results of Yuen and Cavior can be combined in the following implication

$$d_H(\mathbf{x}_i, \mathbf{x}_j) = m \rightarrow D(\mathbf{x}_i, \mathbf{x}_j) \geq \lceil \frac{2^m}{3} \rceil. \tag{2.1}$$

We call this implication the separability property of the standard binary Gray code. In the next we shall derive a more general separability property which holds for N-ary reflected Gray codes, $N \geq 2$. Although an index system for this code is known (cf. [18]), it will appear that such a system is not needed to prove the result. Throughout this thesis, the terms list and Gray code (which is represented by that list) are interchangeable. The columns of this list are numbered from right to left by $1, 2, \ldots, n$. Furthermore, as already announced in Section 1.4, we shall also use matrix notation for these lists where the rows of a matrix represent the codewords of the list.

2.1.1 Equivalence of ordered codes

Let $V_{n,N}$ denote the set of all cyclic minimal-change N-ary Gray codes of length n. Let G be some code in $V_{n,N}$. We shall introduce a number of transformations mapping G to some other (possibly the same) element of $V_{n,N}$:

(i) if a is a permutation of the integers $1, 2, \ldots, n$, then aG is the code of length n obtained by permuting the columns of G according to a;
(ii) if b is the cyclic permutation $(0, 1, 2, \ldots, N-1)$, then b_iG is the code of length n obtained by permuting the integers in the i-th column according to b, for some $i \in \{1, 2, \ldots, n\}$;
(iii) if c is equal to the permutation $(0, N-1)(1, N-2)\ldots(\frac{N-2}{2}, \frac{N}{2})$, for N even, or equal to the permutation $(0, N-1)(1, N-2)\ldots(\frac{N-3}{2}, \frac{N+1}{2})$, for N odd, and $N > 2$, then c_iG is the code of length n obtained by permuting the integers in the i-th column according to c, for some $i \in \{1, 2, \ldots, n\}$ (cf. [20] and also [10, Ch. 2]).

It will be clear that all these transformations define mappings of $V_{n,N}$ onto itself, and also that these transformations generate a group of order $n!(2N)^n$. We remark that the subgroup generated by the transformations (ii) is isomorphic to the translation group $G \rightarrow G + \mathbf{v}, \mathbf{v} \in \{0, 1, \ldots, N-1\}^n$. Furthermore, applying transformation (iii) to column n in case of $G_{ref}(n, N)$, yields the reversed code $G_{ref}(n, N)^R$ for N even.

Whereas for N odd, applying transformation (iii) to all columns of $G_{ref}(n, N)$ also yields $G_{ref}(n, N)^R$.

Definition 2.1.1. *Minimal-change Gray codes which can be transformed into each other by applying one or more of the transformations (i) - (iii) are called equivalent codes.*

The relevance of this definition will become clear from the following proposition.

Proposition 2.1.1. *Equivalent codes satisfy the same separability property.*

The proof is immediate by observing that Hamming distances and list distances are not effected by the transformations (i) - (iii).

2.1.2 Contractions of ordered codes

Let G be some code in $V_{n,N}$. Take two k-strings $\mathbf{a} := a_1 a_2 \dots a_k \in \{0, 1, \dots, N-1\}^k$, and $\mathbf{i} := i_1 i_2 \dots i_k$, with $1 \leq i_1 < i_2 < \dots < i_k \leq n$, for some fixed k-value, $1 \leq k \leq n$. The string $\mathbf{a}$ will be called a *bit pattern* and $\mathbf{i}$ a *position vector*. We now consider the sublist of G consisting of all codewords which have a_j on position i_j, for $1 \leq j \leq k$. Leaving out the common bit pattern $\mathbf{a}$ from these codewords provides us with an ordered code of codeword length $n - k$. We call this code the *contraction* of G with respect to the pair $(\mathbf{a}, \mathbf{i})$, and we write $G(\mathbf{a}, \mathbf{i})$. In particular, we can contract the standard N-ary Gray code $G_{ref}(n, N)$ with respect to some pair $(\mathbf{a}, \mathbf{i})$. The resulting code will be denoted by $G_{ref}(n, N; \mathbf{a}, \mathbf{i})$.

Proposition 2.1.2. *Let $G_{ref}(n, N)$ be the N-ary reflected Gray code, of length $n > 1$. Then for any pair $(\mathbf{a}, \mathbf{i})$, the contraction $G_{ref}(n, N; \mathbf{a}, \mathbf{i})$ is a Gray code equivalent to the reflected Gray code $G_{ref}(n - k, N)$ of length $n - k$.*

Proof. We shall only give the proof for the case N is even, and omit the proof for N odd, which is completely similar.

Since N is even, $G_{ref}(n, N)$ is cyclic. We shall prove the Proposition by applying mathematical induction to n.
(i) The statement is true for $n = 2$, as can be verified by inspection.
(ii) Assume the statement holds for all codeword lengths less than n. Consider the sublist of all codewords of $G_{ref}(n, N)$ containing pattern $\mathbf{a}$ on position $\mathbf{i}$. If $i_k = n$, this sublist is either part of a sublist $a_k G_{ref}(n-1, N)$ or of a sublist $a_k G_{ref}(n-1, N)^R$. In the first case the code $G_{ref}(n - k, N; \mathbf{a}, \mathbf{i})$ can be considered as the contraction $G_{ref}(n-1, N; \mathbf{a}', \mathbf{i}')$, with $\mathbf{a}' = a_1 a_2 \dots a_{k-1}$ and $\mathbf{i}' = i_1 i_2 \dots i_{k-1}$. By the induction assumption, this last code is equivalent to $G_{ref}(n - 1 - k + 1, N) = G_{ref}(n - k, N)$. In the second case we proceed similarly, making use of the equivalence of $G_{ref}(n - k, N)$ and $G_{ref}(n - k, N)^R$ (cf. the remark prior to Definition 2.1.1). If $i_k \neq n$, the contraction process yields a code of type

$$G_{ref}(n,N;\mathbf{a},\mathbf{i}) = \begin{pmatrix} 0 & G_{ref}(n-1,N,\mathbf{a},\mathbf{i}) \\ 1 & G_{ref}(n-1,N,\mathbf{a},\mathbf{i})^R \\ 2 & G_{ref}(n-1,N,\mathbf{a},\mathbf{i}) \\ \vdots & \vdots \\ N-1 & G_{ref}(n-1,N,\mathbf{a},\mathbf{i})^R \end{pmatrix}. \tag{2.2}$$

Again by the induction assumption $G_{ref}(n-1,N;\mathbf{a},\mathbf{i})$ is equivalent to the reflected code $G_{ref}(n-1-k,N)$. Applying Definition (2.1.1) shows that $G_{ref}(n,N;\mathbf{a},\mathbf{i})$ is equivalent to $G_{ref}(n-k,N)$. □

2.1.3 The separability function of the N-ary reflected Gray code

We are ready now to prove our main results of Section 2.1. We start with radix N is even.

Theorem 2.1.3. *Let $G_{ref}(n,N)$ be the N-ary reflected Gray code of length n, and let N be even. If the Hamming distance between two codewords $\mathbf{g}$ and $\mathbf{h}$ satisfies $d_H(\mathbf{g},\mathbf{h}) = m$, then the list distance between $\mathbf{g}$ and $\mathbf{h}$ satisfies $D(\mathbf{g},\mathbf{h}) \geq \lceil \frac{N^m}{N^2-1} \rceil$. Moreover, this lower bound is sharp for all m-values with $1 \leq m \leq n$.*

Proof. We prove the Theorem in two steps.
A. First we take $m = n$. In addition to the statement of the Theorem we shall also prove that there is a pair of codewords at minimum distance, such that the shortest path connecting them in the list $G_{ref}(n,N)$ contains the first codeword as well as the last codeword of the list (1.4). For $n = 1$ and $n = 2$ all above statements are trivial. Assume all these statements are true for all values less than $n > 2$. Let $\mathbf{g}$ and $\mathbf{h}$ be two codewords with $d_H(\mathbf{g},\mathbf{h}) = n$. If we write $\mathbf{g} = g_n g_{n-1} v$ and $\mathbf{h} = h_n h_{n-1} w$, it follows that $g_n \neq h_n$, $g_{n-1} \neq h_{n-1}$ and $d_H(\mathbf{v},\mathbf{w}) = n-2$. From (3) it follows that $\mathbf{v}$ and $\mathbf{w}$ can be considered as codewords of $G_{ref}(n-2,N)$ or of $G_{ref}(n-2,N)^R$. It also follows that $\mathbf{g}$ and $\mathbf{h}$ are separated from each other by at least a number $p(\geq 1)$ of complete blocks $G_{ref}(n-2,N)$ or $G_{ref}(n-2,N)^R$ of size N^{n-2}. So $D(\mathbf{g},\mathbf{h})$ is equal to pN^{n-2} plus a term due to the positions of $\mathbf{v}$ and $\mathbf{w}$ in their respective blocks $G_{ref}(n-2,N)$ or $G_{ref}(n-2,N)^R$. It will be obvious that $D(\mathbf{g},\mathbf{h})$ is minimal if both contributions can be minimized simultaneously. This is indeed possible by taking $p = 1$ and by selecting codewords $\mathbf{v}$ and $\mathbf{w}$, which are both in a block $G_{ref}(n-2,N)$ or both in a block $G_{ref}(n-2,N)^R$ for odd p-values, as described in the beginning of this proof. Due to the induction assumptions $D(\mathbf{g},\mathbf{h})$ is minimal for this choice of $\mathbf{v}$ and $\mathbf{w}$ and its value is equal to

$$N^{n-2} + \lceil \frac{N^{n-2}}{N^2-1} \rceil = \lceil \frac{N^n}{N^2-1} \rceil.$$

Therefore, the Theorem also holds for n. In particular we can take $\mathbf{g} = \mathbf{0}$ and $\mathbf{h} = c1c1c\cdots$, with $c = N-1$, showing that also the additional induction requirement(cf. the first lines of the proof) is satisfied again. By the principle of mathematical induction the Theorem has been proved now for the case $m = n$.

B. If $m < n$, then $\mathbf{g}$ and $\mathbf{h}$ are equal in $k := n - m$ positions, indicated by some position vector $\mathbf{i} = i_1 i_2 \cdots i_k$. The corresponding values of the coordinates will be given by $\mathbf{a} = a_1 a_2 \cdots a_k$. Now, we consider the contraction $G_{ref}(n, N; \mathbf{a}, \mathbf{i})$. Let $\mathbf{v}$ and $\mathbf{w}$ be the codewords in this contraction which correspond to $\mathbf{g}$ and $\mathbf{h}$ respectively. So, we have $d_H(\mathbf{v}, \mathbf{w}) = m$. Since $G_{ref}(n, N; \mathbf{a}, \mathbf{i})$ is equivalent to $G_{ref}(m, N)$, it follows, by Proposition 2.1.1 and part A of this proof, that $D(\mathbf{v}, \mathbf{w}) \geq \lceil \frac{N^m}{N^2-1} \rceil$ in the contracted code. Hence, we have a *fortiori* the same inequality for $D(\mathbf{g}, \mathbf{h})$, since in $G_{ref}(n, N)$ the codewords corresponding to codewords of $G_{ref}(n, N; \mathbf{a}, \mathbf{i})$ will, in general, be interlaced by codewords which have no counterpart in $G_{ref}(n, N; \mathbf{a}, \mathbf{i})$. Finally, one can easily prove that this bound is sharp by applying mathematical induction to $n \geq m$, and using part A for the case $n = m$. □

Corollary 2.1.4. *(Yuen, Cavior) The separability function of the standard binary Gray code is equal to* $\lceil \frac{2^m}{3} \rceil$.

As is known from Theorem 1.4.1 the N-ary reflected Gray code of length $n > 1$ is not cyclic for odd values of radix N. Hence, we use the non-cyclic list distance when dealing with the separability of these codes, and formulate the following theorem.

Theorem 2.1.5. *Let $G_{ref}(n, N)$ be the N-ary reflected Gray code of length n, and let N be odd. If the Hamming distance between two codewords $\mathbf{g}$ and $\mathbf{h}$ satisfies $d_H(\mathbf{g}, \mathbf{h}) = m \geq 1$, then the list distance between $\mathbf{g}$ and $\mathbf{h}$ satisfies $d_L(\mathbf{g}, \mathbf{h}) \geq N^{m-2} + 1$. Moreover, this bound is sharp for all m-values with $1 \leq m \leq n$.*

Proof. For $m = 1$ the Theorem is trivial. For $m \geq 2$, we shall distinguish two cases: $m = n$ and $m < n$, and present a proof for the case $m = n$. The case $m < n$ can be dealt with by a same procedure as we used for case B of Theorem 2.1.3.
Let us take $m = n \geq 2$. We rewrite the reflected Gray code in the following way.

$$G_{ref}(n, N) = \begin{pmatrix} 0 & 0 & G_{ref}(n-2, N) \\ 0 & 1 & G_{ref}(n-2, N)^R \\ \vdots & \vdots & \vdots \\ 0 & N-1 & G_{ref}(n-2, N) \\ 1 & N-1 & G_{ref}(n-2, N)^R \\ 1 & N-2 & G_{ref}(n-2, N) \\ \vdots & \vdots & \vdots \\ 1 & 0 & G_{ref}(n-2, N)^R \\ \vdots & \vdots & \vdots \\ N-1 & 0 & G_{ref}(n-2, N) \\ \vdots & \vdots & \vdots \\ N-1 & N-1 & G_{ref}(n-2, N) \end{pmatrix}, \tag{2.3}$$

with

$$G_{ref}(1,N) \;=\; \begin{pmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ N-1 \end{pmatrix}.$$

Furthermore, we write $\mathbf{g} = g_n g_{n-1}\mathbf{v}$ and $\mathbf{h} = h_n h_{n-1}\mathbf{w}$. In order that $\mathbf{g}$ and $\mathbf{h}$ have Hamming distance $m = n$, we must have $g_n \neq h_n$ and $g_{n-1} \neq h_{n-1}$. This occurs when these codewords are separated by at least one block $G_{ref}(n-2,N)$ or $G_{ref}(n-2,N)^R$, both of which have size N^{n-2}. Hence, the list distance of the codewords $\mathbf{g}$ and $\mathbf{h}$ is at least $N^{n-2}+1$. For detecting how close these two codewords are, let us consider the following arbitrary triple of consecutive blocks in eq. (2.3)

$$\begin{array}{ccc} i-1 & j+(-1)^i & G_{ref}(n-2,N)^* \\ i-1 & j & G_{ref}(n-2,N)^{**} \\ i & j & G_{ref}(n-2,N)^* \end{array}$$

where $0 \leq i,j \leq N-1$. The symbol * (resp. **) stands for R if i is odd (resp. i is even), otherwise it should be deleted. Since N is odd, the Hamming distance between the first and the last codeword of $G(n-2,N)$, or of $G(n-2,N)^R$, is equal to $n-2$. Thus, the last codeword of block $i-1 \;\; j+(-1)^i \;\; G(n-2,N)^*$ and the first codeword of block $i \;\; j \;\; G(n-2,N)^*$ have Hamming distance n. Furthermore, it is clear that these codewords have a list distance which is exactly equal to $N^{n-2}+1$. This last statement also indicates that the lower bound is sharp. □

2.2 On a class of cyclic N-ary Gray code G_N

According to Theorem 1.4.1, it is known that the N-ary reflected Gray codes are cyclic only when N is even, otherwise they are non-cyclic. In this section we are concerned with the construction of a class of N-ary Gray codes which are always cyclic, i.e. for all values of radix N.

We introduce a recursive procedure for generating such codes. We should mention that our procedure for constructing these cyclic Gray codes differs only slightly from the procedure introduced by Sharma and Khanna in [63]. Our procedure however, is simpler and more efficient in terms of the codeword length.

Moreover, we shall focus on equivalence classes of all n-bit N-ary words with respect to their *weight* modulo N. More in particular, we arrange all elements in a class such that the ordering satisfies a *minimal-change* property. As is shown in [74] for the binary case, the ordered list of elements of a class all having the same weight modulo N, is said to satisfy the minimal-change property if any two successive codewords differ in precisely two bit positions. Any such list which satisfies this type of minimal-change is also referred to as Gray code, or better as a *constant-weight Gray code*.

Further study of these codes enhances the separability problem and the *ranking problem* or *index system problem*(shortly *index problem*), i.e the problem of the relationship between a codeword and its index in the code list. A solution of the index problem of an ordered code will help us to determine the position of a given word or vice versa, to determine the codeword on a given position, without generating the whole list.

2.2.1 A recursive construction of Sharma and Khanna

Let $L(n, N)$ be the naturally ordered list of N-ary numbers of length n. Sharma and Khanna in [63] presented a bijective mapping converting the codewords of $L(n, N)$ onto the codewords of an N-ary Gray code with codeword length n, denoted by $G_N(n)$. Let $a_n a_{n-1} \dots a_1$ be a word of length n in the N-ary number system representing the value $\sum_{i=1}^{n} a_i N^{i-1}$. The codeword $g_n g_{n-1} \dots g_1$ in the N-ary Gray code $G_N(n)$ generated by this mapping is defined as

$$\begin{cases} g_n = a_n, \\ g_i = a_i - a_{i+1}(\bmod N), \quad \text{for } i = 1, 2, \dots, n-1. \end{cases} \tag{2.4}$$

Conversely, we have

$$\begin{cases} a_n = g_n, \\ a_i = g_i + a_{i+1}, (\bmod N), \quad \text{for } i = 1, 2, \dots, n-1. \end{cases} \tag{2.5}$$

Bose, Broeg, Kwon, and Ashir in [6] proved that the resulting Gray codes produced by applying (2.4) are always cyclic. We notice that the construction of N-ary Gray codes using (2.4) can also be found in [5]. Further observation concerning how many distinct Gray codes could be produced by repeatedly applying the mapping in (2.4) can be seen in [38, 40]. We discuss below an alternative construction, also introduced by Sharma and Khanna in [63], to generate the same codes.

In the next, the transpose of the matrix A will be denoted by A^T. It can easily be seen that (2.4) produces $G_N(1) = (0\ 1\ \cdots\ N-1)^T$. Let

$$G_N(1)^{[i]} = (N-i\ \ N-i+1\ \ \cdots\ \ N-i-1)^T \tag{2.6}$$

be obtained by shifting cyclically the elements of $G_N(1)$ over i places to the right. Notice that $G_N(1)^{[i]}$ is the N-ary Gray code of length 1 with the first entry $N-i$. We have that $G_N(1)^{[0]} = G_N(1) = (0\ 1\ \cdots\ N-1)^T$. Then $G_N(2)$ is constructed as follows

$$G_N(2) = \begin{pmatrix} 0 & G_N(1)^{[0]} \\ 1 & G_N(1)^{[1]} \\ \vdots & \vdots \\ N-1 & G_N(1)^{[N-1]} \end{pmatrix}. \tag{2.7}$$

To extend this procedure for obtaining $G_N(n)$ in general, the following notations are introduced. First we define

$$G_N(n)^{[0]} = G_N(n) = \begin{pmatrix} \mathbf{g}_0(n) \\ \mathbf{g}_1(n) \\ \vdots \\ \mathbf{g}_{N^n-1}(n) \end{pmatrix}. \tag{2.8}$$

where $\mathbf{g}_i(n)$ is the i-th codeword of length n of $G_N(n)$, whereas $\mathbf{g}_0(n)$ is the all-zero codeword of length n, and

$$G_N(n)^{[k]} = \begin{pmatrix} \mathbf{g}_0(n:k) \\ \mathbf{g}_1(n:k) \\ \vdots \\ \mathbf{g}_{N^n-1}(n:k) \end{pmatrix}. \tag{2.9}$$

Here, we define

$$\mathbf{g}_i(n:k) = \mathbf{g}_i(n) + k\mathbf{g}_{N^n-1}(n)(\text{mod } N), \tag{2.10}$$

where the product of a codeword with an integer $k \leq N-1$, is considered to be the componentwise multiplication of two elements of $\mathbf{Z}_N = \{0, 1, \ldots, N-1\}$, the set of integers modulo N. In terms of the above notation $G_N(n+1)$ is constructed in the following manner:

$$G_N(n+1) = \begin{pmatrix} 0 & G_N(n)^{[0]} \\ 1 & G_N(n)^{[1]} \\ \vdots & \vdots \\ N-1 & G_N(n)^{[N-1]} \end{pmatrix}. \tag{2.11}$$

In [63] Sharma and Khanna remark that $G_N(n)^{[k]}$, as defined by (2.9) and (2.10), is identical with an ordinary cyclic shift of the elements of $G_N(n)$ over k positions when $n = 1$. In general, when $n = l(l \geq 1)$, $G_N(l)$ has N blocks of length N^{l-1} each. The list $G_N(l)^{[k]}$ is the list obtained by cyclically shifting blocks over k block positions, and by simultaneously carrying out k cyclic shifts within each block. It follows that for generating a code of codeword length $n+1$, this procedure involves $\frac{N(N^n-1)}{N-1}$ cyclic shifts to blocks in the code of codeword length n. So, the number of shifts increases exponentially as function of n.

2.2.2 An efficient procedure to construct $G_N(n)$

Let us make a further observation especially with respect to relation (2.10). Observe that

$$\mathbf{g}_{N^n-1}(n:N-1) = \mathbf{g}_{N^n-1}(n) + (N-1)\mathbf{g}_{N^n-1}(n) = \mathbf{0}(\text{ mod } N),$$

for every $n \geq 1$. So, the last codeword of $G_N(n)^{[N-1]}$ is equal to the all-zero codeword of length n. It implies that the last codeword of $G_N(n+1)$, as defined in (2.11), is a codeword with zeros in the first n bits from the right, and with $N-1$ in the $(n+1)$-th bit. Because of this fact, calculations in (2.10) will not affect the first n columns of $G_N(n+1)$ when calculating $G_N(n+1)^{[k]}$ for all k, $0 \leq k \leq N-1$. What actually takes place in (2.10) is merely the addition of the number $k(N-1) \equiv N-k \pmod{N}$ to every bit in the last column of $G_N(n)$ when generating $G_N(n)^{[k]}$.
Furthermore, let p be the cyclic permutation $(0\ N-1 \cdots 1)$, and let p_iG be the code obtained by applying the permutation p to the i-th column of code G, and let p^k stand for applying k times permutation p. We can easily see that $G_N(n)^{[k]} = p_n^k G_N(n)$. In the next, we shall write by $G_N(n)^k$ for $p_n^k G_N(n)$. Based on this observation we introduce a slightly different procedure to obtain the same codes as defined in (2.11). We define

$$G_N(n+1) = \begin{pmatrix} 0 & G_N(n)^0 \\ 1 & G_N(n)^1 \\ \vdots & \vdots \\ N-1 & G_N(n)^{N-1} \end{pmatrix}, \tag{2.12}$$

with

$$G_N(1)^0 = G_N(1) = (0\ 1\ \cdots\ N-1)^T.$$

To produce the code of length $n+1$, our method only requires $N-1$ cyclic permutations $p = (0\ N-1 \cdots 1)$ to the n-th column of the previous code of length n. Notice that the last codeword of $G_N(n)^k$ is the same as the first codeword of $G_N(n)^{k+1}$, $0 \leq k < N-1$. Since $G_N(n)^k$ is a Gray code for each k, $0 \leq k \leq N-1$, $G_N(n+1)$ is also a Gray code. Moreover, it is also obvious that the first codeword of $G_N(n)^0$ and the last codeword of $G_N(n)^{N-1}$ are all-zero codewords. Therefore, the first and the last codewords of $G_N(n+1)$ have Hamming distance and Lee distance 1, like all other pairs of successive codewords. It follows that the code $G_N(n+1)$ is a cyclic code.

Example 2.2.1. Let $n = 3$, and $N = 3$. It is clear that $G_3(1)^0 = (012)^T$, $G_3(1)^1 = (201)^T$, and $G_3(1)^2 = (120)^T$. From (2.12) it follows that

$$G_3(2)^0 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 2 \\ 1 & 2 \\ 1 & 0 \\ 1 & 1 \\ 2 & 1 \\ 2 & 2 \\ 2 & 0 \end{pmatrix}, G_3(2)^1 = \begin{pmatrix} 2 & 0 \\ 2 & 1 \\ 2 & 2 \\ 0 & 2 \\ 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 2 \\ 1 & 0 \end{pmatrix}, G_3(2)^2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 2 & 2 \\ 2 & 0 \\ 2 & 1 \\ 0 & 1 \\ 0 & 2 \\ 0 & 0 \end{pmatrix}.$$

Finally we obtain

$$G_3(3) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 2 & 2 \\ 0 & 2 & 0 \\ 1 & 2 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 1 & 0 & 2 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 0 \\ 2 & 2 & 1 \\ 2 & 0 & 1 \\ 2 & 0 & 2 \\ 2 & 0 & 0 \end{pmatrix}$$

Remark Applying the procedure for radix $N = 2$ yields the binary reflected Gray code. However, for $N > 2$, the code $G_N(n)$ produced by this procedure is not equivalent to the N-ary reflected Gray code.

2.2.3 Constant weight codes

Let b be some element of $\mathbf{Z}_N := \{0, 1, \ldots, N-1\}$. The *weight* of b, denoted by $wt(b)$, is defined as its value as integer of $\mathbb{N}$. According to this definition, $wt(0) = 0$ and $wt(N-1) = N-1$. By $(\mathbf{Z}_N)^n$ we shall denote the set of all vectors of length n over $\mathbf{Z}_N$. The concept of weight associated with elements of $\mathbf{Z}_N$ will be extended to the elements of $(\mathbf{Z}_N)^n$. The weight of a vector $\mathbf{b} \in (\mathbf{Z}_N)^n$, denoted by $wt(\mathbf{b})$, is defined as the sum of the weights of its n components, modulo N. Thus, if $\mathbf{b} = b_n b_{n-1} \ldots b_1 \in (\mathbf{Z}_N)^n$, then $wt(\mathbf{b}) = \sum_{i-1}^{n} b_i (\bmod N)$ (cf. [63]).

Throughout this thesis, a code the codewords of which have a constant weight w will be called a *constant w-weight code*. Below we shall see how one can obtain a

constant w-weight code as a subcode of $C_N(n)$ which also satisfies a minimal-cange property.

Let us consider the sequence $w, w+N, w+2N, \ldots, w+(N^{n-1}-1)N$ of N^{n-1} integers in natural increasing order. When generating the constant w-weight code $C_N(w;n)$ of length n, Sharma and Khanna in [63] applied (2.4) to the N-ary representation of integers in the sequence. Here we define for all w, $0 \leq w \leq N-1$, a procedure for generating the same code as follows.

$$C_N(w;1) = w, \text{ for all } w, \text{ for } n = 1, \tag{2.13}$$

and

$$C_N(w;n) = C_N(0;n)^{N-w}, \text{ for } n > 1, \tag{2.14}$$

with

$$C_N(0;n) = \begin{pmatrix} 0 & C_N(0;n-1) \\ 1 & C_N(N-1;n-1) \\ \vdots & \vdots \\ N-1 & C_N(1;n-1) \end{pmatrix}, \tag{2.15}$$

By applying induction to n and taking into account eqs. (2.12) and (2.13)-(2.15), we can show that $C_N(w;n)$ is a subcode (sublist) of $G_N(n)$. That $C_N(w;n)$ satisfies minimal-change property is shown by Corollary 2.2.3 below.

Example 2.2.2. Let $N = 3$, and $n = 3$. We have $C_3(0;1) = 0$, $C_3(1;1) = 1$, $C_3(2;1) = 2$;

$$C_3(0;2) = \begin{pmatrix} 0 & 0 \\ 1 & 2 \\ 2 & 1 \end{pmatrix}, C_3(1;2) = \begin{pmatrix} 1 & 0 \\ 2 & 2 \\ 0 & 1 \end{pmatrix}, C_3(2;2) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \\ 1 & 1 \end{pmatrix};$$

and

$$C_3(0;3) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \\ 2 & 2 & 2 \\ 2 & 0 & 1 \end{pmatrix}, C_3(1;3) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 2 & 0 \\ 2 & 0 & 2 \\ 2 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{pmatrix}, C_3(2;3) = \begin{pmatrix} 2 & 0 & 0 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \\ 1 & 0 & 2 \\ 0 & 0 & 2 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 2 & 2 \\ 1 & 0 & 1 \end{pmatrix}.$$

In Example 2.2.2 we can observe that columns $n, n-1, \ldots, 2$ of $C_3(w;n)$ constitute the N-ary cyclic Gray code $G_3(n)$. The following Theorem states that this phenomenon is true in general.

Theorem 2.2.1. *The constant weight code $C_N(0;n+1)$, defined by* (2.13) - (2.15)*, can be written as*

$$C_N(0;n+1) = \begin{pmatrix} \mathbf{g}_0 & a_0 \\ \mathbf{g}_1 & a_1 \\ \vdots & \vdots \\ \mathbf{g}_{N^n-1} & a_{N^n-1} \end{pmatrix},$$

where $\mathbf{g}_i$ is the i-th codeword of $G_N(n)$, and with numbers a_i satisfying $\sum_{j-1}^{n} g_{ij} + a_i = 0$ (mod N), for $0 \le i \le N^n - 1$.

Proof. We prove the Theorem using induction to n.

1. By inspecting Example 2.2.2, we can establish that the Theorem is true for $n = 1$, 2 and 3.

2. Assume that the Theorem is true for all codeword lengths less than n. It means that

$$C_N(0;k+1) = \begin{pmatrix} \mathbf{g}_0 & b_0 \\ \mathbf{g}_1 & b_1 \\ \vdots & \vdots \\ \mathbf{g}_{N^n-1} & b_{N^n-1} \end{pmatrix} = \begin{pmatrix} & b_0 \\ G_N(k) & b_1 \\ & \vdots \\ & b_{N^n-1} \end{pmatrix},$$

where $\mathbf{g}_i$ is the i-th codeword of $G_N(k)$ for all $k < n$, and with numbers b_i satisfying $\sum_{j-1}^{k} g_{ij} + b_i = 0 \pmod N$, for $0 \le i \le N^k - 1$. From (2.14) it follows that

$$\begin{aligned} C_N(l;k+1) = C_N(0;k+1)^{N-l} &= \begin{pmatrix} & b_0 \\ G_N(k) & b_1 \\ & \vdots \\ & b_{N^n-1} \end{pmatrix}^{N-l} \\ &= \begin{pmatrix} & b_0 \\ G_N(k)^{N-l} & b_1 \\ & \vdots \\ & b_{N^n-1} \end{pmatrix}, \end{aligned}$$

for all l, $0 \le l \le N-1$. If we put $k = n-1$ in the last equation, we have the following relation $\sum_{j=1}^{n-1} g_{ij} + b_i = l \pmod N$, for $0 \le i \le N^{n-1} - 1$ and $0 \le l \le N-1$. Then we obtain

$$
\begin{aligned}
C_N(l;n+1) &= \begin{pmatrix} 0 & C_N(0;n) \\ 1 & C_N(N-1;n) \\ 2 & C_N(N-2;n) \\ \vdots & \vdots \\ N-1 & C_N(1;n) \end{pmatrix} && (\text{ by (2.15)}) \\
&= \begin{pmatrix} 0 & C_N(0;n) \\ 1 & C_N(0;n)^1 \\ 2 & C_N(0;n)^2 \\ \vdots & \vdots \\ N-1 & C_N(0;n)^{N-1} \end{pmatrix} && (\text{ by (2.14)}) \\
&= \begin{pmatrix} & & b_0 \\ 0 & G_N(n-1)^0 & \vdots \\ & & b_{N^{n-1}-1} \\ & & b_0 \\ 1 & G_N(n-1)^1 & \vdots \\ & & b_{N^{n-1}-1} \\ \vdots & \vdots & \vdots \\ & & b_0 \\ N-1 & G_N(n-1)^{N-1} & \vdots \\ & & b_{N^{n-1}-1} \end{pmatrix} && (\text{ by assumption }).
\end{aligned}
$$

Finally, from (2.12) we may conclude that the Theorem is true. □

Next, applying (2.14), Theorem 2.2.1 can be generalized for all w, $0 \leq w \leq N-1$, as follows,

Corollary 2.2.2. *The constant weight code $C_N(w;n+1)$, defined by* (2.13) - (2.15)*, can be written as*

$$
C_N(w;n+1) = \begin{pmatrix} \mathbf{g}_0 & a_0 \\ \mathbf{g}_1 & a_1 \\ \vdots & \vdots \\ \mathbf{g}_{N^n-1} & a_{N^n-1} \end{pmatrix},
$$

where $\mathbf{g}_i$ is the i-th codeword of $G_N(n)^{N-w}$, and with numbers a_i satisfying $\sum_{j=1}^{n} g_{ij} + a_i = w \pmod N$, for all $w, 0 \leq w \leq N-1$ and for all $i, 0 \leq i \leq N^n - 1$.

Now, we can use Corollary 2.2.2 as an alternative definition of $C_N(w;n+1)$. Furthermore, from Theorem 2.2.1 and Corollary 2.2.2 we have the following results.

Corollary 2.2.3. *For all i, $0 \leq i < N^{n-1} - 1$, $d_H(\mathbf{g}_i, \mathbf{g}_{i+1}) = 2$, for all $\mathbf{g}_i, \mathbf{g}_{i+1} \in C_N(w;n)$. Moreover, the Hamming distance of the last codeword and the first codeword in the list is equal to* 2*.*

Notice that the cyclic permutation b introduced in Subsection 2.1.1, is the inverse of the permutation p. So, we can immediately infer that $G_N(n)^l$ is equivalent to $G_N(n)$. Now, let $W_{n,N}$ denote the set of the constant weight, modulo N, codes of length n. We shall extend the transformations a and b in Subsection 2.1.1 to the elements of $W_{n,N}$. It is also clear that these transformations define mappings of $W_{n,N}$ onto itself. The transformation c will also define a mapping of $W_{n,N}$ onto itself, if the transformation is applied simultaneously to all columns of a code in $W_{n,N}$. As a consequence of Definition 2.1.1 we have

Proposition 2.2.4. *For all $v, w \in \{0, 1, \ldots, N-1\}$, the codes $C_N(v; n)$ and $C_N(w; n)$ are equivalent.*

The proof of Proposition 2.2.4 is immediate from Definition 2.1.1 and the construction in eqs. (2.13)-(2.15).

Now, let $\mathbf{a}$ be a bit pattern $a_1 a_2 \ldots a_k \in \{0, 1, \ldots, N-1\}^k$, and let $\mathbf{i}$ be a position vector $i_1 i_2 \ldots i_k$ with $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ (see Subsection 2.1.2 or [63, 79]). Let $G(n)$ be some ordered code of codeword length n, and let $G(n; \mathbf{a}, \mathbf{i})$ be an n-bit Gray code obtained from $G(n)$ by restricting ourselves to the sublist of codewords containing pattern $\mathbf{a}$ on the positions indicated by $\mathbf{i}$, followed by deleting these positions. This process provides us with an ordered code of codeword length $n - k$. We shall call the code $G(n; \mathbf{a}, \mathbf{i})$ the *contraction* of $G(n)$ with respect to the pair $(\mathbf{a}, \mathbf{i})$. We now have the following Proposition.

Proposition 2.2.5. *For any bit pattern $\mathbf{a}$ and any position vector $\mathbf{i}$, both of length k, the contraction $G_N(n; \mathbf{a}, \mathbf{i})$ is a cyclic Gray code equivalent to the cyclic Gray code $G_N(n - k)$.*

Proof. We shall prove the Proposition by applying mathematical induction to n. By inspection we can verify that the statement is true for $n = 2$. Now, assume the Proposition holds for all codeword lengths less than n. The assumption has as consequence that for all m, $1 \leq m < n$, $G_N(m; \mathbf{a}, \mathbf{i})^l$ is equivalent to $G_N(m - k)^l$, for $1 \leq k \leq m$, and $0 \leq l \leq N - 1$.
Consider the subcode of $G_N(n)$ containing pattern $\mathbf{a}$ on position $\mathbf{i}$. If $i_k = n$, such a subcode is part of a subcode $a_n G_N(n-1)^l$ for some l, $0 \leq l \leq N - 1$. Therefore, the code $G_N(n; \mathbf{a}, \mathbf{i})$ can be considered as the contraction of $G_N(n-1; \mathbf{a}, \mathbf{i})^l$ for some l, $0 \leq l \leq N-1$, with $\mathbf{a} = a_1 a_2 \ldots a_{k-1}$ and $\mathbf{i} = i_1 i_2 \ldots i_{k-1}$. According to the induction assumption, we have that $G_N(n; \mathbf{a}, \mathbf{i})$ is equivalent to $G_N(n - k)^l$. By making use of the equivalence of $G_N(n - k)^l$ and $G_N(n - k)$ as remarked earlier in this section, we infer that $G_N(n; \mathbf{a}, \mathbf{i})$ is equivalent to the cyclic code $G_N(n - k)$. If $i_k \neq n$, the contraction yields a code of type

$$G_N(n; \mathbf{a}, \mathbf{i}) = \begin{pmatrix} 0 & G_N(n-1; \mathbf{a}, \mathbf{i})^0 \\ 1 & G_N(n-1; \mathbf{a}, \mathbf{i})^1 \\ \vdots & \vdots \\ N-1 & G_N(n-1; \mathbf{a}, \mathbf{i})^{N-1} \end{pmatrix},$$

Again by the induction assumption, $G_N(n-1;\mathbf{a},\mathbf{i})^l$ is equivalent to $G_N(n-k-1)^l$ for all $l, 0 \leq l \leq N-1$. Because of (2.12), we conclude that $G_N(n;\mathbf{a},\mathbf{i})$ is equivalent to $G_N(n-k)$. □

2.2.4 The separability function for $G_N(n)$ and $C_N(w;n)$

Observe that applying transformation (i), (ii) or (iii) to G preserves the Hamming distance as well as the list distance in G. It implies that these transformations preserve the separability property of G. So, we have the following Proposition which was also stated in [79].

Proposition 2.2.6. *Equivalent codes satisfy the same separability property.*

An immediate simple result following from Propositions 2.2.4 and 2.2.6 is the following.

Corollary 2.2.7. *For all $v, w \in \{0, 1, \ldots, N-1\}$, the constant weight codes $C_N(v;n)$ and $C_N(w;n)$ have the same separability.*

The following theorem solves the separability problem of the cyclic Gray code $G_N(n)$ introduced in Subsection 2.2.2.

Theorem 2.2.8. *Let $G_N(n)$ be the cyclic Gray code produced by construction (2.12). If the Hamming distance between two codewords $\mathbf{g}$ and $\mathbf{h}$ satisfies $d_H(\mathbf{g},\mathbf{h}) = m$, then the list distance of these codewords satisfies $D(\mathbf{g},\mathbf{h}) \geq \lceil \frac{N^m}{N^2-1} \rceil$.*

Proof. Let us again consider the code $G_N(n)$ defined in Section 3. The complete proof will be accomplished in two steps.

1. Let $m = n$ and let codewords $\mathbf{g}$ and $\mathbf{h}$ in $G_N(n)$ have a Hamming distance $m = n$. Let also $\mathbf{g} = g_n g_{n-1}\mathbf{v}$ and $\mathbf{h} = h_n h_{n-1}\mathbf{w}$, where $\mathbf{v}$ and $\mathbf{w}$ are in $G_N(n-2)^l$ and $G_N(n-2)^{l'}$ respectively, for some l and l', with $0 \leq l, l' \leq N-1$. In order that $g_n \neq h_n$ and $g_{n-1} \neq h_{n-1}$, codewords beginning with $g_n g_{n-1}$ and $h_n h_{n-1}$ respectively, are separated by at least one block of type $G_N(n-2)^l$ of size N^{n-2}. Let $\mathbf{v} = v_{n-2}v_{n-3}\mathbf{x}$ and $\mathbf{w} = w_{n-2}w_{n-3}\mathbf{y}$ with $\mathbf{x}$ and $\mathbf{y}$ in $G_N(n-4)^s$ and $G_N(n-4)^s$ respectively, for some s and s', $0 \leq s, s' \leq N-1$. By similar arguments as above, we infer that if $v_{n-2} \neq w_{n-2}$ and $v_{n-3} \neq w_{n-3}$, codewords in $G_N(n-2)^l$ beginning with $v_{n-2}v_{n-3}$ and $w_{n-2}w_{n-3}$ respectively, are separated by at least one block of type $G_N(n-4)^s$ of size N^{n-4} . Continuing in this way, we finally infer that codewords $\mathbf{g}$ and $\mathbf{h}$ at Hamming distance $m = n$, must be separated at least by a series of blocks of size $N^{n-2}, N^{n-4}, \ldots, N($ or $1)$ if n odd (or n even). Thus, the list distance of $\mathbf{g}$ and $\mathbf{h}$ is at least equal to $N^{n-2} + N^{n-4} + \ldots + N^{\xi} + 1$, where $\xi = 1$ or 0, which is equal to $\lceil \frac{N^n}{N^2-1} \rceil$.

2. Assume that $m < n$. Let $\mathbf{g}$ and $\mathbf{h}$ differ in m positions. Thus, $\mathbf{g}$ and $\mathbf{h}$ have $k = n - m$ bits in common, indicated by a position vector $\mathbf{i} = i_1 i_2 \ldots i_k$. The corresponding values of the coordinates will be given by $\mathbf{a} = a_1 a_2 \ldots a_k$. Now, we consider the contraction $G_N(n;\mathbf{a},\mathbf{i})$. Let $\mathbf{v}$ and $\mathbf{w}$ be the codewords in this

contraction corresponding to the codewords $\mathbf{g}$ and $\mathbf{h}$ respectively. So, we have $d_H(\mathbf{v}, \mathbf{w}) = m$. Since $G_N(n; \mathbf{a}, \mathbf{i})$ is equivalent to $G_N(m)$, by Proposition 2.2.6 and part 1 of this proof, we infer that the list distance of $\mathbf{v}$ and $\mathbf{w}$ satisfies $D(\mathbf{v}, \mathbf{w}) \geq \lceil \frac{N^m}{N^2-1} \rceil$ in the contracted code. In $G_N(n)$ therefore, we certainly have also $D(\mathbf{g}, \mathbf{h}) \geq \lceil \frac{N^m}{N^2-1} \rceil$.

□

From the proof of part 1 it also follows that for $n = m$ the lower bound is sharp. For $n \geq m$ this property can then easily be proved by induction to n. We shall give now an alternative proof, based on the index system of $G_N(n)$.

Theorem 2.2.9. *For all $0 < m \leq n$, there exists at least one pair of codewords in $G_N(n)$ with Hamming distance m and list distance equal to $\lceil \frac{N^m}{N^2-1} \rceil$.*

Proof. Let $B = N - 1$ and $M = B - 1$. Let $\mathbf{g}_i = 0 \ldots 01B0 \ldots 0$ with 1 as the m-th coordinate and $\mathbf{g}_j = 0 \ldots 0MB1B1 \ldots 1B$ with M as the $(m-1)$-st coordinate. It is obvious that the Hamming distance of $\mathbf{g}_i$ and $\mathbf{g}_j$ satisfies $d_H(\mathbf{g}_i, \mathbf{g}_j) = m$. Using eqs. (2.5), we have that the indices of $\mathbf{g}_i$ and $\mathbf{g}_j$ in the related N-ary number system are $\mathbf{i} = 0 \ldots 010 \ldots 0$ and $\mathbf{j} = 0 \ldots 0MBMB \ldots MB$, if m is even, or $\mathbf{j} = 0 \ldots 0MBM \ldots BM$, if m is odd. Here, we shall only proceed for m is even, and leave the m is odd case to the reader. The list distance $D(\mathbf{g}_i, \mathbf{g}_j)$ is equal to

$$\begin{aligned} |i-j| &= |N^{m-1} - (MN^{m-2} + BN^{m-3} + \ldots + MN + B)| \\ &= |B(N^{m-2} + N^{m-3} + \ldots N + 1) + 1 - (MN^{m-2} + BN^{m-3} + \ldots + MN + B)| \\ &= |(N^{m-2} + N^{m-4} + \ldots N + 1) + 1| \\ &= \lceil \frac{N^m}{N^2 - 1} \rceil. \end{aligned}$$

So, the Theorem has been proved. □

The above Theorem indicates that the lower bound $\lceil \frac{N^m}{N^2-1} \rceil$ is a sharp bound for all m, $1 \leq m \leq n$. Therefore, this lower bound is the separability function of $G_N(n)$. Since $G_N(n)^l$ and $G_N(n)$ are equivalent, according to Proposition 2.2.6 the separability function of $G_N(n)^l$, $0 \leq l \leq N-1$, is the same as the one of $G_N(n)$. As remarked in Section 3, the cyclic Gray code $G_2(n)$ is the standard binary Gray code. So, we have the following Corollary

Corollary 2.2.10 (Yuen [90], Cavior [9]). *The separability function of the binary reflected Gray code $G_2(n)$ is equal to $\lceil \frac{2^m}{3} \rceil$ with $1 \leq m \leq n$.*

The separability function for $C_N(w; n)$ now immediately follows from Theorem 2.2.8 and Theorem 2.2.9.

Theorem 2.2.11. *For all m, $1 < m \leq n$, the list distance of codewords $\mathbf{g}_i$ and $\mathbf{g}_j$ in $C_N(w; n)$ with Hamming distance $d_H(\mathbf{g}_i, \mathbf{g}_j) = m$, satisfies $D(\mathbf{g}_i, \mathbf{g}_j) \geq \lceil \frac{N^{m-1}}{N^2-1} \rceil$, for all w, $0 \leq w \leq N-1$. Moreover, this lower bound is sharp for all relevant m-values.*

We remark that the case $m = 1$ is not relevant for the formulation of Theorem 2.2.11, since different codewords of a constant equivalent weight code always differ in at least two positions.

2.2.5 Index system of $C_N(w; n)$

Equations (2.4) and (2.5) are actually the solution of the index problem of $G_N(n)$. Since we have Theorem 2.2.1 and Corollary 2.2.2, the index system of $C_N(w; n)$ can be derived immediately.
Let $\mathbf{g}_i = g_{in} \dots g_{i2} g_{i1}$ be the i-th codeword of $C_N(w; n)$. From Corollary 2.2.2 we infer that the codeword $g_{in} \dots g_{i2}$ has the same index i in $G_N(n-1)^{N-w}$. Using (2.5), we can immediately determine the index of $\mathbf{g}_i$ by calculating the index of $g_{in} \dots g_{i2}$ in $G_N(n-1)^{N-w}$ or of $\gamma_{in} g_{i(n-1)} \dots g_{i2}$ in $G_N(n-1)$ with $\gamma_{in} = g_{in} - w$.

Example 2.2.3. The index of the codeword 2301 in $C_4(2; 4)$ has the same index as the codeword 230 in $G_4(3)^{4-2} = G_4(3)^2$. This codeword is equal to the codeword 030 in $G_4(3)$. Using (2.5), the index of the codeword 030 has quaternary representation 033 which stands for the value $3.4 + 3 = 15$. So, the codeword 2301 in $C_4(2; 4)$ has index 15.

Conversely, if the index of a codeword in $C_N(w; n)$ is given, say i, we first determine the codeword with the same index i in $G_N(n-1)$. Using (2.4) the codeword can immediately be obtained, say $\mathbf{g}_i = g_{in} \dots g_{i2}$. The corresponding codeword in $G_N(n-1)^{N-w}$ is $\gamma_{in} g_{i(n-1)} \dots g_{i2}$, with $\gamma_{in} = g_{in} + w \pmod N$. Hence, the corresponding codeword in $C_N(w; n)$ is $\gamma_{in} g_{i(n-1)} \dots g_{i2} g_{i1}$ with $g_{i1} = w - (\gamma_{in} + g_{i(n-1)} + \dots + g_{i2}) = -(g_{in} + g_{i(n-1)} + \dots + g_{i2}) \pmod N$. We now have completely solved the index problem of $C_N(w; n)$ for $0 \le w \le N - 1$.

Example 2.2.4. Let us determine the codeword in $C_5(3; 4)$ with index 45. This index number has 5-ary representation 140. Using (2.4), we have that the codeword in $G_5(3)$ with this index is equal to 131. The corresponding codeword in $G_5(3)^{5-3} = G_5(3)^2$ is 431. In order that the codeword $431s$ is in $C_5(3; 4)$, we must have $s = -(1 + 3 + 1) \pmod 5 = 0$. Thus the codeword with index 45 in $C_5(3; 4)$ is the codeword 4310.

2.3 A binary Gray code with high separability capacity

In this section we shall only consider binary Gray codes.

From Corollary 2.1.4 we know that if $\mathbf{g}_i$ and $\mathbf{g}_j$ are two codewords in the reflected Gray code of length n with indices i and j respectively. Then we have

$$d_H(\mathbf{g}_i, \mathbf{g}_i) = m \to D(\mathbf{g}_i, \mathbf{g}_j) \ge \lceil \frac{2^m}{3} \rceil. \tag{2.16}$$

This implication is called the *separability property* of the binary reflected Gray code. As for the separability capacity of Gray codes, one can pose the natural question of the existence of Gray codes with a better separability capacity.

Problem 2.3.1. *Does there exist a Gray code and a bound $b(m)$, such that if $d_H(\mathbf{g}_i, \mathbf{g}_j) = m$ then $D(\mathbf{g}_i, \mathbf{g}_j) > b(m) \geq \lceil \frac{2^m}{3} \rceil$ for all m-values with $2 < m \leq n$?*

A weaker version of the above requirement yields the following problem.

Problem 2.3.2. *Does there exist a Gray code and a bound $b(m)$, such that if $d_H(\mathbf{g}_i, \mathbf{g}_j) = m$, then $D(\mathbf{g}_i, \mathbf{g}_j) \geq b(m) \geq \lceil \frac{2^m}{3} \rceil$, $2 \leq m \leq n$, whereas at least for one m-value $D(\mathbf{g}_i, \mathbf{g}_j) > \lceil \frac{2^m}{3} \rceil$, for all pairs $\mathbf{g}_i$ and $\mathbf{g}_j$ with $d_H(\mathbf{g}_i, \mathbf{g}_j) = m$?*

Park and Bose in [52] constructed a new class of Gray codes and proved that this class of codes has the following separability properties

$$d_H(\mathbf{g}_i, \mathbf{g}_i) = m \rightarrow D(\mathbf{g}_i, \mathbf{g}_j) \geq \begin{cases} \lceil \frac{4}{15} 2^m \rceil, & \text{if } m \text{ is odd,} \\ \lceil \frac{7}{15} 2^m \rceil, & \text{if } m \text{ is even.} \end{cases} \tag{2.17}$$

We can see immediately that this class of Gray codes has a better separability capacity than the standard Gray codes if m even and > 4, whereas for m odd and > 4, their separability is worse.

In this section we are concerned with Gray codes which have a better separability capacity than the reflected Gray codes. We shall introduce another class of such Gray codes in Section 2.3.1, which are constructed by using a modification of the *transition sequence* of the reflected Gray code. As for these transition sequences, in this section we shall use the following notations:

1. T^n stands for the non-complete transition sequence of the reflected Gray code of length n;
2. T_n stands for the sequence which is obtained from T^n by interchanging the integers $n-1$ and n;
3. $T^1 = T_1 = 1$ and $T^0 = T_0 =$ empty.

Notice that the transition sequence T_n is equal to $T^{n-2}, n, T^{n-2}, n-1, T^{n-2}, n, T^{n-2}$, for $n \geq 2$.

It is well known that

$$\begin{aligned} T^1 &= 1, \\ T^n &= T^{n-1} n T^{n-1}, \qquad n > 1. \end{aligned}$$

For instance we have $T^2 = 1, 2, 1$, $T^3 = 1, 2, 1, 3, 1, 2, 1$, and $T^4 = 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1$.

The remaining part of this section will be organized as follows. The construction of a class of Gray codes, the so-called *nearly optimal Gray codes*, is discussed in Subsection 2.3.1. We prove an optimality property of this class of Gray codes in 2.3.2. In 2.3.3 we discuss the index system of the Gray codes constructed in 2.3.1. Finally, some conclusions are described in 2.3.4.

2.3.1 Construction of nearly optimal Gray codes

This subsection is devoted to Problem 2.3.2 raised in the previous section. It is obvious that there is no bound b as required such that $b(m) > \lceil \frac{2^m}{3} \rceil$ for $m = 1$ and $m = 2$. Our conjecture is that neither functions b can be constructed such that $b(m) > \lceil \frac{2^m}{3} \rceil$ for $m = 3$. To discuss b-values for $m \geq 4$, we introduce the notion of nearly optimal. We start with the following definition.

Definition 2.3.1. We call a cyclic Gray code of length n nearly optimal (with respect to the separation property) if it satisfies the following requirements:

1. There exists an integer m_0, $0 < m_0 \leq n$, such that for $0 < m < m_0$ and $0 \leq i, j \leq 2^n - 1$, $d_H(\mathbf{g}_i, \mathbf{g}_j) = m$ implies $D(\mathbf{g}_i, \mathbf{g}_j) \geq \lceil \frac{2^m}{3} \rceil$; and

2. There exists some integer m_1, $m_0 \leq m_1 \leq n$ such that if $d_H(\mathbf{g}_i, \mathbf{g}_j) = m_1$, then there exists an integer $b(m_1)$ such that $D(\mathbf{g}_i, \mathbf{g}_j) \geq b(m_1) > \lceil \frac{2^{m_1}}{3} \rceil$.

We shall show in the next that for any integer $n \geq 4$, a nearly optimal Gray code for $m_0 = n$, can be constructed.

Construction 2.1.

Start from a codeword (usually the zero codeword) of length $n \geq 4$. Generate the next $2^n - 1$ codewords using the sequence

$$T(n) := T^{n-2}, n-1, T_{n-2}, n, T^{n-2}, n-1, T_{n-2}, n$$

Notice that the sequence $T(n)$ in Construction 2.1 is a slight modification of the transition sequence of the binary standard Gray codes. In this case, the integers $n-2$ and $n-3$ are interchanged after each occurrence of the integer $n-1$.
We shall prove later that $T(n)$ is the complete transition sequence of a cyclic Gray code for every $n \geq 1$.

Example 2.3.1. We know that $T^2 = 1, 2, 1$ and $T_2 = 2, 1, 2$. Therefore we obtain $T(4) = 1, 2, 1, 3, 2, 1, 2, 4, 1, 2, 1, 3, 2, 1, 2, 4$. The resulting Gray code is listed in Figure 2.1.

We can verify that for every pair of codewords $\mathbf{g}_i$ and $\mathbf{g}_j$, with $d_H(\mathbf{g}_i, \mathbf{g}_j) = m$, $1 \leq m \leq 4$, we have that $D(\mathbf{g}_i, \mathbf{g}_j) \geq \lceil \frac{2^m}{3} \rceil$. Furthermore, we also have that $D(\mathbf{g}_i, \mathbf{g}_j) = 8 > \lceil \frac{2^4}{3} \rceil$, for every pair of codewords with Hamming distance 4.

Using Construction 2.1 we obtain that the transition sequence of the new Gray code of length five is $T(5) = T^3, 4, T_3, 5, T^3, 4, T_3 =$

$$1, 2, 1, 3, 1, 2, 1, 4, 1, 3, 1, 2, 1, 3, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 3, 1, 2, 1, 3, 1.$$

Each pair of complementary codewords in this code has list distance at least $\lceil \frac{2^5}{3} \rceil + 2^2 > \lceil \frac{2^5}{3} \rceil$. Again by inspection we can verify that the code generated by

0000	1111
0001	1110
0011	1100
0010	1101
0110	1001
0100	1011
0101	1010
0111	1000

Figure 2.1: Example of a 4-bit nearly optimal Gray code

transition sequence $T(5)$ is also a Gray code satisfying the requirements 1 and 2 of Definition 2.3.1 for all relevant m-values. The question now is, whether the resulting Gray codes are also nearly optimal for all lengths $n \geq 4$.

First we shall prove that $T(n)$ really constitutes a transition sequence of an n-bit cyclic Gray code. To this end, we refer to the following lemma which is due to Gilbert in [20, Section 2].

Let Q_n be the n-dimensional cube, or shortly n-cube, i.e. the graph the vertices of which are binary strings of length n, while the edges are all pairs of vertices which differ in exactly one position. The reflected Gray code $G_{ref}(n)$ of length n is a Hamiltonian cycle in Q_n.

Lemma 2.3.1. *An L-tuple $T = a_1, a_2, \ldots, a_L, \quad a_i \in \{1, 2, \ldots, n\}$ is the transition sequence of a cycle in Q_n if and only if every non-empty subsequence of length less than L contains at least one digit an odd number of times while T itself contains every digit an even number of times.*

According to Lemma 2.3.1, establishing that $T(n)$ is a transition sequence of a cyclic Gray code of length n, comes down to showing that every subsequence of $T(n)$ of length $1, 2, \ldots,$ or $2^n - 1$ contains at least one digit an odd number of times, while the complete sequence itself contains every digit an even number of times.

Theorem 2.3.2. *The sequence $T(n)$ in Construction 2.1 is a transition sequence of a cyclic Gray code of length n.*

Proof. Consider the complete sequence $T(n) = T^{n-2}, n-1, T^{n-2}, n, T^{n-2}, n-1, T^{n-2}, n,$ produced by the Construction 2.1. It is obvious that every subsequence of length 1 contains one digit an odd number of times and the sequence $T(n)$ itself contains every digit an even number of times. The proof is completed by showing that each subsequence S of T^{n-2} of length more than two contains at least one digit an odd number of times. Notice that $T_{n-2} = T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}$.

Since T^{n-4} is a subsequence of the transition sequence of the binary reflected Gray code, any S which is a subsequence of T^{n-4} contains some digits an odd number of times. Now, let S not be a subsequence of T^{n-4}. Thus, S contains integer $n-2$ or $n-3$. If S contains $n-3$, then S contains at least the integer $n-3$ an odd number of times, otherwise S contains the integer $n-2$ an odd number of times. □

2.3.2 A proof of near-optimality

Let $n \geq 4$ and consider the transition sequences T^n and $T(n)$ described below.

1.
$$\begin{aligned} T^n := \; & T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-1, \\ & T^{n-4}, \mathbf{n-3}, T^{n-4}, \mathbf{n-2}, T^{n-4}, \mathbf{n-3}, T^{n-4}, n, \\ & T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-1, \\ & T^{n-4}, \mathbf{n-3}, T^{n-4}, \mathbf{n-2}, T^{n-4}, \mathbf{n-3}, T^{n-4}, \end{aligned}$$

2.
$$\begin{aligned} T(n) := \; & T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-1, \\ & T^{n-4}, \mathbf{n-2}, T^{n-4}, \mathbf{n-3}, T^{n-4}, \mathbf{n-2}, T^{n-4}, n, \\ & T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-1, \\ & T^{n-4}, \mathbf{n-2}, T^{n-4}, \mathbf{n-3}, T^{n-4}, \mathbf{n-2}, T^{n-4}. \end{aligned}$$

We shall show that the code produced by Construction 2.1 is nearly optimal with $m_0 = n$. We distinguish two cases: $m \leq n-1$, and $m = n$.

Case 1. Let us consider the Hamming distance $m \leq n-1$. Since each T^{n-4} contains $n-4$ distinct integers, we are able to determine a shortest subsequence S containing m distinct integers an odd number of times in the subsequence $T^{n-4}, n-i, T^{n-4}, n-j, T^{n-4}, n-k, T^{n-4}$, of $T(n)$ for $i, j, k = 0, 1, 2$ or 3 and i, j, k all different. Furthermore, in T^n the shortest subsequence of type S is contained in the subsequence $T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n, T^{n-4}$, or in $T^{n-4}, n, T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}$, which really has the same pattern and also the same length as the subsequence $T^{n-4}, n-i, T^{n-4}, n-j, T^{n-4}, n-k, T^{n-4}$. Hence, using Yuen's lower bound, we conclude that the length of the subsequence S is at least $\lceil \frac{2^m}{3} \rceil$.

Case 2. Let $m = n$. To compose a subsequence S containing $m = n$ distinct integers an odd number of times, we need a subsequence either of type

$$\begin{aligned} & T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-1, \\ & T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}, n, T^{n-4}, \end{aligned}$$

or

$$\begin{aligned} & T^{n-4}, n-1, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-2, \\ & T^{n-4}, n, T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, \end{aligned}$$

or

$$\begin{aligned} & T^{n-4}, n, T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}, n-3, \\ & T^{n-4}, n-1, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}. \end{aligned}$$

Notice that to produce a subsequence S in T^n, we need a subsequence either of type

$$T^{n-4}, n-3, T^{n-4}, n-1, T^{n-4}, n-3, T^{n-4}, n-2,$$
$$T^{n-4}, n-3, T^{n-4}, n, T^{n-4},$$

or

$$T^{n-4}, n, T^{n-4}, n-3, T^{n-4}, n-2, T^{n-4}, n-3,$$
$$T^{n-4}, n-1, T^{n-4}, n-3, T^{n-4}.$$

So, a sequence in $T(n)$ needed to produce such a sequence S, has a greater length than a similar sequence in T^n. The difference in length is equal to the length of the subsequence of type $n-i, T^{n-4}, n-j, T^{n-4}$, which equals $2 \cdot 2^{n-4} = 2^{n-3}$. Since we know that the length of a subsequence in T^n containing m distinct integers an odd number of times is at least equal to $\lceil \frac{2^n}{3} \rceil$, the length of the subsequence S is at least $\lceil \frac{2^n}{3} \rceil + 2^{n-3}$.

In the sequel, a Gray code of length n constructed by using Construction 2.1 will be denoted by $G_{nop}(n)$. So, we have proven the main result formulated in the following theorem.

Theorem 2.3.3. *For all $n \geq 4$, the Gray code $G_{nop}(n)$ of length n is nearly optimal. Moreover, the list distance of complementary codewords in $G_{nop}(n)$ is at least $\lceil \frac{2^n}{3} \rceil + 2^{n-3}$.*

One immediate consequence is the following corollary.

Corollary 2.3.4. *The Gray code $G_{nop}(4)$ of length 4 has optimal separability capacity.*

Proof. Suppose that a Gray code $\bar{G}(4)$ of length 4 exists such that for all $\mathbf{x}, \mathbf{y}, \mathbf{v}, \mathbf{w}$ in $\bar{G}(4)$ $d(\mathbf{v}, \mathbf{w}) = 4$ implies $D(\mathbf{v}, \mathbf{w}) = 8$, and $d(\mathbf{x}, \mathbf{y}) = 3$ implies $D(\mathbf{x}, \mathbf{y}) > 3$. Consider the zero codeword $\mathbf{x}_0 = 0000$. It follows that $\mathbf{x}_8 = 1111$. The codewords which have Hamming distance 3 to $\mathbf{x}_0$ are $\mathbf{a}_1 = 0111$, $\mathbf{a}_2 = 1011$, $\mathbf{a}_3 = 1101$, and $\mathbf{a}_4 = 1110$. Evidently, there exist two $\mathbf{a}_i$'s which have list distance 5 to 0000, and the other two $\mathbf{a}_j$'s have list distance 7 to 0000. Let $\mathbf{a}_{ij}$ be a vector of length 4 such that $d(\mathbf{a}_i, \mathbf{a}_{ij}) = 1 = d(\mathbf{a}_{ij}, \mathbf{a}_j)$, for $i, j \in \{1, 2, 3, 4\}$. Then we can see that $d(\mathbf{a}_{ij}, \mathbf{a}_{kl}) = 4$, where $i, j, k, l \in \{1, 2, 3, 4\}$ and $\{i, j\} \cap \{k, l\} = \emptyset$. Assume that $\mathbf{x}_5 = \mathbf{a}_i$ and $\mathbf{x}_{11} = \mathbf{a}_k$. This implies $\mathbf{x}_7 = \mathbf{a}_j$ and $\mathbf{x}_9 = \mathbf{a}_l$. Thus we have $\mathbf{x}_6 = \mathbf{a}_{ij}$ and $\mathbf{x}_{10} = \mathbf{x}_{kl}$. But $D(\mathbf{x}_6, \mathbf{x}_{10}) = D(\mathbf{a}_{ij}, \mathbf{a}_{kl}) = 4 < 8$. This contradicts the assumption that any two complementary codewords in $\bar{G}(4)$ have list distance 8. So, Gray codes of length 4 where complementary codewords are at list distance 8, have the property that there exist at least two codewords with Hamming distance 3 and list distance 3. This implies that $G_{nop}(4)$ has optimal separability capacity. □

2.3.3 Index system

Of course, interchanging the positions of the integers $n-3$ and $n-2$ in the transition sequence of a Gray code of length n will influence the $(n-3)$-rd and $(n-2)$-nd

bits of codewords in the Gray code. Since the transition sequence of $G_{nop}(n)$ is related to the transition sequence of the corresponding $G_{ref}(n)$, there must exist a close relationship between them. For an arbitrary value $n \geq 4$, we can observe that the difference between T^n and $T(n)$ merely concerns the order of occurrence of the integers $n-2$ and $n-3$. By observing the occurrences of these integers, we infer that differences come in whenever one of the following conditions holds.

1. The integers $n-3$ and $n-2$ have equal parity and the integer $n-1$ has odd parity;
2. The integers $n-3$ and $n-2$ have different parity and the integer n has odd parity.

Equivalently, we can conclude that differences occur as soon as one of the following conditions holds.

1. The $(n-3)$-rd and the $(n-2)$-nd bit are the same and the $(n-1)$-st bit is 1;
2. The $(n-3)$-rd and the $(n-2)$-nd bit are different and the n-th bit is 1.

In case 1 as well as in case 2, the conversion of some nearly optimal Gray codeword into the corresponding reflected Gray codeword is accomplished merely by adding the codeword $00110\ldots0$, since the difference is caused by the interchanging of the integers $n-3$ and $n-2$. Thus, if $\mathbf{w}$ is a codeword in a nearly optimal Gray code and if $\mathbf{g}$ is the corresponding codeword in the related reflected Gray code, then we have the following relation

$$\mathbf{g} = \begin{cases} \mathbf{w} \oplus 00110\ldots0, & \text{if } \mathbf{w} \text{ satisfies condition 1 or 2,} \\ \mathbf{w}, & \text{otherwise,} \end{cases} \tag{2.18}$$

or, vice versa,

$$\mathbf{w} = \begin{cases} \mathbf{g} \oplus 00110\ldots0, & \text{if } \mathbf{w} \text{ satisfies condition 1 or 2,} \\ \mathbf{g}, & \text{otherwise.} \end{cases} \tag{2.19}$$

For example, the converted codewords of 01001010 and 100101101 are 01111010 and 101001101 respectively. Now, we are ready to determine the one - one correspondence between nearly optimal codewords and their indices. To do this, we need the well-known bijective mapping between codewords and indices for the binary reflected Gray code(see e.g. [15, 18, 63, 38, 90]).

Let $\mathbf{g}_i = g_{in}g_{in-1}\ldots g_{i1}$ be a codeword in the reflected Gray code with index i, and let the binary representation of i be $i_n i_{n-1}\ldots i_1$. The mapping and its inverse are as follows

$$g_{ij} = \begin{cases} i_n, & j = n \\ i_j \oplus i_{j+1}, & 1 \leq j < n \end{cases} \tag{2.20}$$

$$i_j = \begin{cases} g_{in}, & j = n \\ g_{ij} \oplus i_{j+1}, & 1 \leq j < n. \end{cases} \tag{2.21}$$

Now, we have a one-one correspondence between the indices $i(\mathbf{g})$ of codewords $\mathbf{g}$, in $G_{ref}(n)$ and indices $i(\mathbf{w})$ of codewords $\mathbf{w}$, in the related Gray code $G_{nop}(n)$ as follows,

$$i(\mathbf{w}) = i(\mathbf{g}), \tag{2.22}$$

whenever $\mathbf{w}$ and $\mathbf{g}$ satisfy the conditions (2.18) or (2.19).

Example 2.3.2. To determine the index of the codeword $\mathbf{g} = 010100$ in Gray code $G_{nop}(6)$ of length 6 we carry out the following procedure. First, convert $\mathbf{g}$ to its counterpart in the reflected Gray codeword. Since $\mathbf{g}$ does not satisfy condition 1 or 2, we have $\mathbf{g} = \mathbf{w} = 010100$. Thus $i(\mathbf{w}) = i(\mathbf{g}) = 011000$, which stands for the value $2^4 + 2^3 = 24$.

Example 2.3.3. Now let us determine the codeword $\mathbf{w}$ of $G_{nop}(6)$ with index $i = 37$. First, we calculate $\mathbf{g}_{37}$. The binary representation of 37 is 100101, so $\mathbf{g}_{37} = 110111$. Notice that $\mathbf{g}_{37}$ satisfies condition 2, and hence $\mathbf{w}_{37} = \mathbf{g}_{37} \oplus 001100 = 111011$.

2.3.4 Conclusion

In Subsection 2.3.1, we presented Construction 2.1 for producing a nearly optimal code with $m_0 = n$. An interesting question is whether it is possible to find the smallest value m_0 which is less than n. However, our conjecture is that such an m_0-value does not exist.

3

Transition Count Spectra of Gray Codes

Throughout this chapter we shall only deal with *binary* Gray codes. A necessary condition for the existence of Gray codes with respect to a given transition count spectrum is proved. Moreover, a sufficient condition is conjectured for the existence of Gray codes with a given transition count spectrum. A proof is given for the existence of so-called balanced Gary codes, i.e. Gray codes which have a transition count spectrum which is as uniform as possible. The proof is more straightforward than the one of Bhat and Savage in [3] and leads to a simple construction of balanced Gray codes. Moreover, our construction can yield balanced Gray codes which can not be produced by using Bhat and Savage's method in [3], nor by using Bakos' method in [1]. A simple constructive proof for the existence of exponentially balanced Gray codes is also derived. This proof is much simpler than an earlier proof presented in [77]. All these proofs are based on the Gray construction in [32, Theorem D] which is an independent reformulation of the Gray construction of Bakos in [1].

3.1 Introduction

The usefulness of the binary reflected Gray code and its widespread appearance are undisputed, while its optimality with respect to various applications has proved itself frequently(cf. [2]). For certain applications however, sometimes additional properties of Gray codes are requested. A codacon spectrograph for instance, uses Gray codes with large minimum *run length*, which is the distance between two successive changes of the same bit(cf. [21]). In experimental design, one is interested in Gray codes which possess balanced distribution of changes(see e.g. [39, 41, 42, 87]). For an extended survey we refer to [61].

Let $TC_n(i)$ be the transition count of integer i. With respect to the list of codewords, $TC_n(i)$ refers to the number of times bits in column i change from 0 to 1 or from 1 to 0. If $G(n)$ is a cyclic code, then it will be clear that $TC_n(i)$, $1 \leq i \leq n$, is even and moreover, that $\sum_{i=1}^{n} TC_n(i) = 2^n$. The reflected Gray code $G_{ref}(n)$ of length n has the following transition counts

$$TC_n(i) = \begin{cases} 2^{n-i}, & \text{if } 1 \leq i \leq n-1, \\ 2, & \text{if } i = n. \end{cases} \tag{3.1}$$

A necessary condition for the existence of Gray codes w.r.t. its transition count spectrum is formulated in the following theorem.

Theorem 3.1.1. *Let $(c_1, c_2, \ldots, c_n)$ be the transition count spectrum of a cyclic Gray code $G(n)$ of length n which is ordered in non-decreasing way, i.e. $c_i \leq c_{i+1}$, for all i, $1 \leq i \leq n-1$. Then, one has $\sum_{i=1}^{k} c_i \geq 2^k$, for all k, $1 \leq k \leq n$.*

For a very short proof of the Theorem we refer to [32, p. 85]. Here, we shall give a more detailed proof, which actually is an extended version of the proof in [32]

Proof. Let $G(n)$ be a cyclic Gray code with transition count spectrum TC_n and with complete transition sequence $\bar{S}(n) := s_1, s_2, ..., s_{2^n}$. Suppose that there exist integers k such that $\sum_{i=1}^{k} c_i < 2^k$, and that K is the largest k satisfying this inequality. Then $K < n$, otherwise $G(n)$ is not a Gray code. Let $L := n - K$. It is obvious that $\sum_{i=1}^{K} c_i = 2^n - \sum_{j=1}^{L} c_{K+j}$. Now consider the list of codewords of $G(n)$.
Since $TC_n(n) = c_n$, there are c_n pairs of consecutive codewords in $G(n)$ which differ in bit position n (the first $n-1$ bits are the same). Assume that $I_l := \{i \in [2^n] | s_i = l\}$, $\forall l \in [n]$. Notice that $c_l = |I_l|$, $\forall l \in [n]$. We take the following steps.

For codeword $\mathbf{x}_i$ with index i in $G(n)$ we write $\mathbf{x}_i = x_{in}\mathbf{x}_i^1$. Then for all $i \in I_n$, we have that $\mathbf{x}_{i-1}^1 = \mathbf{x}_i^1$. Remove all c_n codewords $\mathbf{x}_i$, $i \in I_n$ from $G(n)$. It is easy to see that for all $i \neq j$, $i, j \in I_n$, one has $\mathbf{x}_i^1 \neq \mathbf{x}_j^1$, otherwise we would have $\mathbf{x}_j = \mathbf{x}_i$ or $\mathbf{x}_{i-1}$. Then leave out the n-th bit from the $2^n - c_n$ remaining codewords, and call the resulting list $G^1(n-1)$. We emphasize that we shall keep the original indices for the punctured codewords in $G^1(n-1)$. Notice that all distinct codewords of length $n-1$ are in $G^1(n-1)$.

We follow the same procedure for the list $G^1(n-1)$ of length $n-1$. Let $\mathbf{x}_i^1 = x_{in-1}\mathbf{x}_i^2$, for every codeword $\mathbf{x}_i^1$ of index i in $G^1(n-1)$. Remove all c_{n-1} codewords $\mathbf{x}_i^1$, $\forall i \in I_{n-1}$, from $G^1(n-1)$. Again we infer that $\mathbf{x}_i^2 \neq \mathbf{x}_j^2$, for all $i, j \in I_{n-1}$, $i \neq j$. The number of the remaining codewords is equal to $2^n - c_n - c_{n-1}$. Call $G^2(n-2)$ the list consisting of these codewords after removing the bits in position $n-1$. We conclude that $2^n - c_n - c_{n-1} \geq 2^{n-2}$, since $G^2(n-2)$ should contain all 2^{n-2} distinct codewords.

Repeating the same procedure L times, we arrive at the list $G^L(n-L)$ which contains at least 2^{n-L} codewords and exactly 2^{n-L} of these codewords are distinct. However, the number of remaining codewords is equal to $2^n - \sum_{j=1}^{L} c_{K+j} = \sum_{i=1}^{K} c_i <$

$2^K = 2^{n-L}$, and so, $\sum_{i=1}^{k} c_i \geq 2^k$, which contradicts the assumption in the beginning of our proof. □

An obvious corollary of Theorem 3.1.1 is the following

Corollary 3.1.2. $\lceil \frac{2^i}{i} \rceil \leq c_i \leq \frac{2^n - 2^{i-1}}{n+1-i}$ *for all* $i, 1 \leq i \leq n$.

Proof. The lower bound for c_i is trivial. The maximal value of c_i will be reached when $\sum_{j=1}^{i-1} c_j$ reaches its minimum 2^{i-1}. We conclude that the maximal value of c_i equals $\frac{2^n - 2^{i-1}}{n+1-i}$. □

We can extend Theorem 3.1.1 to the non-cyclic case as follows.

Theorem 3.1.3. *Let* $(c_1, c_2, \ldots, c_n)$ *be the transition count spectrum of a non-cyclic Gray code* $G(n)$ *of length* n *which is ordered in non-decreasing way. Then, one has* $\sum_{i=1}^{k} c_i \geq 2^k - 1$, *for all* k, $1 \leq k \leq n$.

A sufficient condition for the existence of cyclic Gray codes w.r.t. the transition count spectrum is conjectured by Evdokimov [16] and Knuth [32] as follows.

Conjecture 3.1.4. [**Evdokimov**, **Knuth**] *Let* $\sum_{i=1}^{n} c_i = 2^n$ *where* $c_i \leq c_{i+1}$, $1 \leq i \leq n-1$, *and* c_j *is even for all* j, $1 \leq j \leq n$. *If* $\sum_{i=1}^{k} c_i \geq 2^k$, *for all* $k, 1 \leq k < n$, *then there exists a Gray code of length* n *which has transition count spectrum* $(c_1, c_2, \ldots, c_n)$.

The second problem posed by Bhat and Savage in [3, Section 4] also corresponds to Conjecture 3.1.4.

It seems that the Gray construction introduced by Ludman and Sampson in [42] might be used to construct Gray codes with required transition count spectra. Unfortunately the validity of the construction is not proved yet for large codeword length n.

In the following sections we focus on balanced and on exponentially balanced Gray codes, including totally balanced Gray codes. All these discussions are based on Bakos' Gray construction in [1], which is independently reformulated in [32, Theorem D].

3.2 Balanced Gray codes

In this section we introduce a straightforward technique to prove the existence of balanced Gray codes. This technique gives rise to a simple procedure for the construction of such codes, which can be applied under weaker conditions than the methods of Bakos [1], Bhat and Savage [3], and of Robinson and Cohn [60].

Let $G(n)$ be a Gray code of length n, and let $TC_n := (TC_n(1), TC_n(2), ..., TC_n(n))$ be its transition count spectrum. As is defined in Chapter 1, an n-bit Gray code with transition counts satisfying $|TC_n(i) - TC_n(j)| \leq 2$ for every $1 \leq i, j \leq n$, is called

a *balanced* code, and it is called a *totally balanced* code if $TC_n(i) = TC_n(j)$ for all i and j. Since $\sum_{i=1}^{n} TC_n(i) = 2^n$, a necessary condition for a Gray code to be totally balanced is that n is equal to a power of 2. We can easily verify that the reflected Gray codes of length $1, 2$ and 3 are balanced and, moreover, those of length 1 and 2 are totally balanced. However, for $n \geq 4$ the reflected Gray code $G_{ref}(n)$ is not balanced.

Because of the importance of balanced Gray codes (cf. [3, 39, 41, 42, 87, 88]), for instance for designing experiments or for designing and testing electrical circuits and information systems, there is a considerable interest in this type of Gray codes. Recently, Liu and Schrack in [39] introduced a heuristic construction to construct balanced Gray codes. This method is applicable in cases when the codeword length n is rather small, just like the methods introduced in [41, 42, 87]. A similar remark can be made with respect to the construction of Wagner and West in [88] for totally balanced Gray codes.

Two more constructions for balanced Gray codes, in [1, 60], are worthwhile to be mentioned. These two constructions are of advantage for producing balanced Gray codes for large values of the codeword length. In [60] the claim was made - but not completely proved - that balanced Gray codes exist for every codeword length $n \geq 1$. For obtaining a balanced Gray code of length n, Robinson and Cohn's approach in [60] requires a special sequence of integers taken from the transition sequence of an $(n-2)$-bit balanced Gray code. Bhat and Savage in [3] proved that such a sequence always exists, thus completing the proof of Robinson and Cohn for the existence of balanced Gray codes for all values of $n \geq 1$. A similar but shorter proof which is introduced by Knuth in [32, p. 15 and p. 85] is due to Robinson and Cohn in [60]. Actually, this result was already proved by Bakos in [1] in the context of truth functions, and unnoticed by other authors of articles on the subject of balanced or uniform Gray codes. In [72, 77] we combined the approaches of Robinson and Cohn in [60] and of Bakos in [1] to obtain a straightforward technique for the construction of balanced Gray codes.

Although Bakos' Gray construction in [1] and the one of Robinson and Cohn in [60] look quite similar, the first construction can be carried out under weaker requirements. In this section we focus on the construction of balanced Gray codes based on [32, Theorem D] which constitutes a reformulation of Bakos' Gray construction. We slightly modify and extend our technique introduced in [72] to adjust it to this construction. Our modified technique gives rise to a simple procedure for the construction of balanced Gray codes for any codeword length $n \geq 4$.

The remaining part of this section is organized as follows. In Subsection 3.2.1, we discuss a Gray construction which is a variation of Bakos' method. A slight modification of the technique introduced in [72] for balancing Gray codes, is discussed in Subsection 3.2.2. In Subsection 3.2.3, we summarize the discussion of Subsection 3.2.2 to derive a simple procedure for the construction of balanced Gray codes. At the end of that Section we present some examples to illustrate our method. We emphasize that the codes produced in Examples 3.2.5 and 3.2.6 cannot be constructed by the methods discussed in [1, 3, 60], while the construction itself is simpler and more

straightforward than the method in [1]. Some remarks concerning two questions posed by Bhat and Savage in [3, Section 4] are discussed in Subsection 3.2.4.

3.2.1 A Gray code construction

In this Chapter the word *subsequence* (of some sequence S) stands for what sometimes is called a *contiguous* or *consecutive* subsequence, i.e. all elements of this subsequence are consecutive in S. Let $\bar{S}(n)$ be the complete transition sequence of a Gray code of codeword length n, and let u be a subsequence of $\bar{S}(n)$ which may be empty. We denote by u^R the sequence obtained from u by reversing its order. For instance, if $u = 1, 3, 2, 4, 1$, then $u^R = 1, 4, 2, 3, 1$. For the sake of efficiency, if n is a positive integer, then the set $\{1, 2, \ldots, n\}$ is frequently denoted by $[n]$.

The following lemma is introduced in [69]. For the sake of simplicity, in this lemma we shall write $ab\mathbf{x}_i$ instead of $a_i b_i \mathbf{x}_i$. So, ab in $ab\mathbf{x}_i$ and ab in $ab\mathbf{x}_j$, $i \neq j$ do not have to be the same.

Lemma 3.2.1. *Let $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_m$ be m codewords of length n, with $\mathbf{y}_i = ab\mathbf{x}_i$ for all $1 \leq i \leq m$, and $ab \in \{00, 01, 11, 10\}$, and let $s_i \in [n]$ be the transition between $\mathbf{y}_{i-1}$ and $\mathbf{y}_i$, $2 \leq i \leq m$. Let $u := s_1, s_2, \ldots, u_{m-1}$, then the codewords corresponding to the transition sequence $u, n-1, u^R, n, u$, or $u, n, u^R, n-1, u$, have the form $cd\mathbf{x}$ where $\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ and $cd \in \{00, 01, 11, 10\}$.*

Proof. Let us consider the transition sequence $u, n-1, u^R, n, u$. Starting at $\mathbf{y}_1$, it is clear that the leftmost u in this transition sequence generates codewords $ab\mathbf{x}_1$, $ab\mathbf{x}_2$, ..., $ab\mathbf{x}_m$, with $ab \in \{00, 01, 11, 10\}$. When the transition $n-1$ occurs, the next generated codeword is $a(b \oplus 1)\mathbf{x}_m$. Here, $\oplus$ means addition modulo 2 without carry. So, $a(b \oplus 1)$ is also in $\{00, 01, 11, 10\}$. The sequence u^R then generates codewords $a(b \oplus 1)\mathbf{x}_{m-1}$, $a(b \oplus 1)\mathbf{x}_{m-2}$, ..., $a(b \oplus 1)\mathbf{x}_1$. The next codeword generated by transition n is $(a \oplus 1)(b \oplus 1)\mathbf{x}_1$. Again it is clear that $(a \oplus 1)(b \oplus 1) \in \{00, 01, 11, 10\}$. Furthermore, the last transition sequence u will generate codewords $(a \oplus 1)(b \oplus 1)\mathbf{x}_2$, $(a \oplus 1)(b \oplus 1)\mathbf{x}_3$, ..., $(a \oplus 1)(b \oplus 1)\mathbf{x}_m$. Thus, the codewords generated by the transition sequence $u, n-1, u^R, n, u$ are

$$ab\mathbf{x}_1, ..., ab\mathbf{x}_m, a(b \oplus 1)\mathbf{x}_m, a(b \oplus 1)\mathbf{x}_{m-1}, ..., a(b \oplus 1)\mathbf{x}_1,$$

$$(a \oplus 1)(b \oplus 1)\mathbf{x}_1, (a \oplus 1)(b \oplus 1)\mathbf{x}_2, ..., (a \oplus 1)(b \oplus 1)\mathbf{x}_m.$$

Using the same arguments, the codewords generated by the transition sequence $u, n, u^R, n-1, u$ are

$$ab\mathbf{x}_1, ..., ab\mathbf{x}_m, (a \oplus 1)b\mathbf{x}_m, (a \oplus 1)b\mathbf{x}_{m-1}, ..., (a \oplus 1)b\mathbf{x}_1,$$

$$(a \oplus 1)(b \oplus 1)\mathbf{x}_1, (a \oplus 1)(b \oplus 1)\mathbf{x}_2, ..., (a \oplus 1)(b \oplus 1)\mathbf{x}_m.$$

The Lemma has been proved now. □

The following theorem is from [32, Theorem D] which is a reformulation of Bakos' Theorem in [1]. The differences in notation and the opposite parity of l compared with the formulation in [32] are due to our convention with respect to the labelling of bit positions and the indexing of codewords as introduced in Chapter 1.

Theorem 3.2.2. *Let $\bar{S}(n-2) := u_0, s_{j_1}, u_1, s_{j_2}, ..., u_{l-1}, s_{j_l}, u_l, s_{2^n}$ be the transition sequence of an $(n-2)$-bit Gray code, where each u_k is a possibly empty sequence of transitions, and l is even. Then the sequence*

$$\begin{aligned} u_0, s_{j_0}, u_1, ..., s_{j_l}, u_l, n-1, \\ u_l^R, n, u_l, n-1, u_l^R, s_{j_l}, \\ u_{l-1}^R, n-1, u_{l-1}, n, u_{l-1}^R, s_{j_{l-1}}, \\ \vdots \\ u_1^R, n-1, u_1, n, u_1^R, s_{j_1}, \\ u_0^R, n, u_0, n-1, u_0^R, n, \end{aligned}$$

is the transition sequence of an n-bit Gray code.

Proof. A proof of the Theorem can be accomplished using Lemma 3.2.1. □

For a complete proof we refer to [32, p. 14].

Example 3.2.1. Consider the complete transition sequence $\bar{S}(3) = 1\ ,\underline{2}, 1,\ 3,\ 1,\ \underline{2}, 1, 3,$ of the 3-bit reflected Gray code. Take $u_0 = 1$, $s_{j_1} = 2$, $u_1 = 1, 3, 1$, $s_{j_2} = 2$, and $u_2 = 1$. Applying construction in Theorem 3.2.2 we obtain a 5-bit Gray code with the complete transition sequence

$$1, 2, 1, 3, 1, 2, 1, 4, 1, 5, 1, 4, 1, 2, 1, 3, 1, 4, 1, 3, 1, 5, 1, 3, 1, 2, 1, 5, 1, 4, 1, 5.$$

In the sequel, the sequence of transitions $s_{j_1}, s_{j_2}, ..., s_{j_l}$, in Theorem 3.2.2 will be denoted by T. Thus, the length of the sequence T is equal to l. We emphasize that the sequence T does not include the closing transition s_{2^n} of $\bar{S}(n)$, as is evident from the notation in Theorem 3.2.2. Hence, we also say that $S(n)$ which is defined by the sequence $\bar{S}(n)$ in the Theorem constitutes the basis of the construction in Theorem 3.2.2.

One can easily derive that the Gray code of length n constructed by applying Theorem 3.2.2, has transition count spectrum $(TC_n(1), TC_n(2), ..., TC_n(n))$, with

$$TC_n(i) := \begin{cases} l+2, & \text{if } i = n-1, n, \\ 4TC_{n-2}(i) - 2b(i), & \text{if } i \in \{1, ..., n\} \backslash \{s_{2^{n-2}}\}, \\ 4(TC_{n-2}(i) - 1) - 2b(i), & \text{if } i = s_{2^{n-2}}. \end{cases} \tag{3.2}$$

where $b(i)$ is the number of times the integer i occurs in the sequence T. Notice that the sum of all $b(i)$, $1 \leq i \leq n-2$, is equal to l, the length of T.

It should be mentioned here that the Gray code constructions in [32, Theorem D], and in [72, 77], which are an extended version of the construction in [60], are all modified versions of Bakos' Gray construction in [1]. Bakos proved his results in quite a different context, and his work was unnoticed for long by authors of articles in the field of ordered codes[1]. Furthermore, we remark that an extension of

[1] We are indebted to prof. dr. A.A. Evdokimov for drawing our attention to Bakos' contribution to this topic.

[32, Theorem D] which holds for the opposite parity of l, was introduced in [32, Exercise 50] and was also proved in [67, Construction 1] (See Theorem 4.2.3 in Chapter 4.).

The list $(p_1, p_2, ..., p_m)$, $m > 0$, is called an *m-partition* of the integer 2^n, if p_i is a positive integer for all i, $1 \leq i \leq m$, $p_1 \leq p_2 \leq \cdots \leq p_m$, and $\sum_{i=1}^{m} p_i = 2^n$. An m-partition of 2^n will be denoted by $P_m(2^n)$. If p_i is even for all i, $1 \leq i \leq m$, the partition $P_m(2^n) = (p_1, p_2, ..., p_m)$ is called an *even m-partition* of 2^n, and it is called *balanced* if for every i, j, $1 \leq i, j \leq n$, we have $|p_i - p_j| \leq 2$.

Theorem 3.2.3. *Let $\bar{S}(n-2)$ be the transition sequence of an $(n-2)$-bit Gray code $G(n-2)$, with transition counts $TC_{n-2}(i)$, $1 \leq i \leq n-2$, where $TC_n(i) \leq TC_n(i+1)$, $1 \leq i \leq n-3$. Let furthermore $P_n(2^n) = (p_1, p_2, ..., p_n)$ be an even n-partition of 2^n. Applying Theorem 3.2.2 yields an n-bit Gray code with transition count spectrum $P_n(2^n) = (p_1, p_2, ..., p_n)$ if and only if*

(i) $p_k = p_{k+1}$ *for some* $k \in [n-1]$;
(ii) $2TC_{n-2}(i) \leq p_i \leq 4TC_{n-2}(i)$ *for every* $i \in [k-1]$;
$2TC_{n-2}(i) \leq p_{i+2} \leq 4TC_{n-2}(i)$ *for every* i, $k \leq i \leq n-2$;
(iii) *there is at least one* $i_0 \in [n-2]$ *such that either* $p_{i_0} \leq 4(TC_{n-2}(i_0)-1)$, $i_0 \in [k-1]$ *or* $p_{i_0+2} \leq 4(TC_{n-2}(i_0)-1)$, $k \leq i_0 \leq n-2$.

Proof. We start by proving the only-if-part of the Theorem. Let $G(n)$ be the n-bit Gray code which has transition count spectrum $P_n(2^n) = (p_1, p_2, ..., p_n)$. According to (3.2), the transition counts $TC_n(n)$ and $TC_n(n-1)$ are equal. It implies $p_k = p_{k+1}$ for $k = n-1 \in [n-1]$. Since $0 \leq b(i) \leq TC_{n-2}(i)$, for all i, $1 \leq i \leq n-2$, it is easy to see that (3.2) implies (ii). Now, let $j = s_{2^{n-2}}$(the closing transition of $\bar{S}(n-2)$). Since Theorem 3.2.2 only uses the subsequence $S(n-2)$ defined by $\bar{S}(n-2)$, the number of integers j occurring in $S(n-2)$ is equal to $TC_{n-2}(j) - 1$. Hence, equality (3.2) implies (iii) of the Theorem.
Now, we prove the if-part of the Theorem. For all $i \in [n-2] \backslash \{i_0\}$, we define

$$b(i) = \begin{cases} \frac{4TC_{n-2}(i)-p_i}{2}, & i \in [k-1], \\ \frac{4TC_{n-2}(i)-p_{i+2}}{2}, & k \leq i \leq n-2, \end{cases} \tag{3.3}$$

where the numbers p_i are from the given partition $P_n(2^n)$. For i_0 we define

$$b(i_0) = \begin{cases} \frac{4(TC_{n-2}(i_0)-1)-p_{i_0}}{2}, & \text{if } i_0 \in [k-1], \\ \frac{4(TC_{n-2}(i_0)-1)-p_{i_0+2}}{2}, & \text{if } k \leq i_0 \leq n-2. \end{cases} \tag{3.4}$$

From (ii) and (iii) it follows that $0 \leq b(i) \leq TC_{n-2}(i)$ for all i, $1 \leq i \leq n-2$. From (3.3) and (3.4) we obtain

$$\sum_{i=1}^{n-2} b(i) = \sum_{i=1, i\neq i_0}^{k-1} 4TC_{n-2}(i) - p_i + \sum_{i=k}^{n-2} 4TC_{n-2}(i) - p_{i+2} + 4(TC_{n-2}(i_0) - 1) - p_{i_0}, \quad (3.5)$$

if $i_0 \in [k-1]$, or

$$\sum_{i=1}^{n-2} b(i) = \sum_{i=1}^{k-1} 4TC_{n-2}(i) - p_i + \sum_{i=k, i\neq i_0}^{n-2} 4TC_{n-2}(i) - p_{i+2} + 4(TC_{n-2}(i_0) - 1) - p_{i_0+2}, \quad (3.6)$$

if $k \leq i_0 \leq n-2$. It is easy to check that (3.5) as well as (3.6) implies $\sum_{i=1}^{n-2} b(i) = p_k - 2$. Let l be an integer in $[2^{n-2}]$ such that $s_l = i_0$ in $\bar{S}(n-2)$. Shift $\bar{S}(n-2)$ over l positions to the left, in cyclic sense. The resulting transition sequence $\bar{S}(n-2)^l$ has i_0 as its closing transition. Now consider the transition sequence $S(n-2)$ which is defined from the new transition sequence $\bar{S}(n-2) := \bar{S}(n-2)^l$. Take a sequence T consisting of $b(i)$ integers i in $S(n-2)$, for all i, $1 \leq i \leq n-2$, and apply Theorem 3.2.2. Then the resulting Gray code $G(n)$ of length n will have transition count spectrum $(p_1, p_2, ..., p_n)$. □

Example 3.2.2. The transition count spectrum of $G_{ref}(2)$ is $(2, 2)$. Theorem 3.2.3 guarantees that Gray codes of length 4 exist with the following transition count spectra.

$$(2, 2, 4, 8), \qquad (2, 4, 4, 6), \qquad (4, 4, 4, 4).$$

Example 3.2.3. We know that the transition count spectrum of $G_{ref}(3)$ is equal to $(2, 2, 4)$. Due to Theorem 3.2.3, we can conclude that Gray codes of length 5 with the following transition count spectra exist:

$$\begin{array}{lllll} (2,2,4,8,16), & (2,4,4,6,16), & (4,4,4,4,16), & (2,4,4,8,14), & (2,4,6,6,14), \\ (4,4,4,6,14), & (2,2,8,8,12), & (4,4,4,8,12), & (4,4,6,6,12), & (2,4,8,8,10), \\ (4,4,6,8,10), & (4,6,6,6,10), & (4,4,8,8,8), & (4,6,6,8,8), & (6,6,6,6,8). \end{array}$$

It is obvious that any cyclic Gray code of length 3 has transition count spectrum $(2, 2, 4)$. Therefore, we may conclude that Theorem 3.2.2 can not be applied to produce any Gray code of length 5 with transition count spectrum $(2, 2, 6, 6, 16)$ since the requirement (iii) can not be satisfied. Neither do Gray codes with transition count spectrum $(2, 4, 6, 8, 12)$ because of Condition (i).

Example 3.2.4 (Balanced Gray codes). We can easily verify with Theorem 3.2.3 that, starting from balanced Gray codes of length 2 and 3, balanced Gray codes of length n, $2 \leq n \leq 10$, exist with the following transition count spectra:

$$\begin{array}{ccc} (2,2), & (6,6,6,6,8), & (32,32,32,32,32,32,32,32), \\ (2,2,4), & (10,10,10,10,12,12), & (56,56,56,56,56,58,58,58,58), \\ (4,4,4,4), & (18,18,18,18,18,18,20), & (102,102,102,102,102,102,102,102,104,104). \end{array}$$

Examples 3.2.5 and 3.2.6 in Section 3.2.3 give the details for the construction of balanced Gray codes of length 9 and 10 starting from balanced Gray codes of length 7 and 8, respectively.

3.2.2 A proof for the existence of balanced Gray codes

Bakos' method in [1], for establishing the existence of a balanced Gray code for any codeword length $n \geq 1$, is based on a graphical approach. Here, we shall introduce a proof for the same statement using an algebraic approach.

In the next, a partition $P_m(2^n) = (p_1, p_2, ..., p_m)$ of 2^n, which was defined in the previous section, will be called a *balanced even* m-partition if it satisfies both p_i is even and $|p_i - p_j| \leq 2$, for all $1 \leq i, j \leq m$. First, we shall show that a balanced even n-partition of the integer 2^n, for every $n \geq 1$, exists.

Let $2^n = qn + r$, $0 \leq r < n$. We distinguish between the cases q is even and q is odd.
Case I. q is even. This implies r is even. We define a set $Q \subseteq [n-2]$ which consists of the last $\frac{r}{2}$ elements of $[n-2]$, and define the integers p_i, $1 \leq i \leq n$, according to

$$p_i := \begin{cases} q, & i \in [n] \backslash Q, \\ q+2, & i \in Q. \end{cases} \tag{3.7}$$

Remark 3.2.1. *If n is a power of two, then the value of r is zero. This implies that the value of p_i is equal to q for all i, $1 \leq i \leq n$.*

Case II. q is odd. Now, the integer $n + r$ is even. Here, we define $Q \subseteq [n-2]$ which consists of the last $\frac{n+r}{2}$ elements of $[n-2]$, and define

$$p_i := \begin{cases} q-1, & i \in [n] \backslash Q, \\ q+1, & i \in Q. \end{cases} \tag{3.8}$$

It is easy to verify in each case that $\sum_{i=1}^{n} p_i = 2^n$ and that $|p_i - p_j| \leq 2$ for every $1 \leq i, j \leq n$. Thus, the partition $(p_1, p_2, ..., p_n)$ is really a balanced even n-partition of 2^n. Moreover, one can easily verify that a balanced even n-partition of the integer 2^n is unique, or equivalently, if $(\underbrace{p, ..., p}_{k}, \underbrace{p+2, ..., p+2}_{n-k})$ is a balanced even n-partition of 2^n, then p and k are uniquely determined. This implies that a balanced Gray code of length n must have a transition count spectrum $(p_1, p_2, ..., p_n)$, $1 \leq i \leq n$, where p_i is defined by (3.7) or (3.8), and hence we have that $\lfloor \frac{2^n}{n} \rfloor - 1 \leq p_i \leq \lfloor \frac{2^n}{n} \rfloor + 2$, for all i.

Lemma 3.2.4. *If* $x = \lfloor \frac{2^{n-2}}{n-2} \rfloor$ *and* $y = \lfloor \frac{2^n}{n} \rfloor$, *then*

(*i*) $\frac{4(x-2)-(y+2)}{2} > 0$, *for all* $n \geq 9$;

(*ii*) $\frac{4(x+2)-(y-1)}{2} < (x-1)$, *for all* $n \geq 11$.

Proof. We shall only prove part (i) of the Lemma, and omit the similar proof of part (ii). Since $x = \lfloor \frac{2^{n-2}}{n-2} \rfloor$ and $y = \lfloor \frac{2^n}{n} \rfloor$, we have that

$$x \leq \frac{2^{n-2}}{n-2} < x+1 \text{ and } y \leq \frac{2^n}{n} < y+1.$$

This implies

$$\begin{aligned} 4(x-2)-(y+2) &= 4(x+1)-y-14 > 4\frac{2^{n-2}}{n-2} - \frac{2^n}{n} - 14 \\ &= \frac{2^{n+1}}{n(n-2)} - 14 > 0, \end{aligned}$$

for all $n \geq 9$. □

Lemma 3.2.5. *If there exists a balanced Gray code* $G(n-2)$ *of length* $n-2$, $n \geq 11$, *then there also exists a balanced Gray code* $G(n)$ *of length* n.

Proof. Let $(TC_{n-2}(1), TC_{n-2}(2), ..., TC_{n-2}(n-2))$ be the transition count spectrum of a balanced Gray code $G(n-2)$ of length $n-2$, and let $(p_1, p_2, ..., p_n)$ be a balanced even partition of the integer 2^n, $n \geq 11$. By considering (3.7) and (3.8), it will be clear that there is some k, $1 \leq k \leq n-1$, such that $p_k = p_{k+1}$. So, part (i) of Theorem 3.2.3 is satisfied. Furthermore, we have

$$\lfloor \frac{2^{n-2}}{n-2} \rfloor - 1 \leq TC_{n-2}(i) \leq \lfloor \frac{2^{n-2}}{n-2} \rfloor + 2, \tag{3.9}$$

for all $i \in [n-2]$, and

$$\lfloor \frac{2^n}{n} \rfloor - 1 \leq p_j \leq \lfloor \frac{2^n}{n} \rfloor + 2, \tag{3.10}$$

for all $j \in [n]$.
We shall show that $0 \leq \frac{4TC_{n-2}(i)-p_j}{2} \leq TC_{n-2}(i)$, for all $i \in [n-2]$ and $j \in [n]$, and hence, that part (ii) of Theorem 3.2.3 holds.
From (3.9) and (3.10) we obtain for all $i \in [n-2]$ and $j \in [n]$ that

$$4(\lfloor \frac{2^{n-2}}{n-2} \rfloor - 1) - (\lfloor \frac{2^n}{n} \rfloor + 2) \leq 4TC_{n-2}(i) - p_j \leq 4(\lfloor \frac{2^{n-2}}{n-2} \rfloor + 2) - (\lfloor \frac{2^n}{n} \rfloor - 1).$$

Due to Lemma 3.2.4, we can immediately conclude that $0 \leq \frac{4TC_{n-2}(i)-p_j}{2} \leq TC_{n-2}(i)$, for all $i \in [n-2]$ and $j \in [n]$.
More in particular, for every $n \geq 11$, part (i) of Lemma 3.2.4 implies that $0 < \frac{4(TC_{n-2}(i)-1)-p_j}{2}$, for all $i \in [n-2]$ and $j \in [n]$. Thus, part (iii) of Theorem 3.2.3 holds. From Theorem 3.2.3 we conclude that a balanced Gray code of length $n \geq 11$ exists which has a transition count spectrum $(p_1, p_2, ..., p_n)$. □

It is clear that $G_{ref}(1)$ is a totally balanced Gray code. Example 3.2.4 shows that balanced Gray codes of length n, $2 \leq n \leq 10$, exist. Therefore, because of Lemma 3.2.5 and Remark 3.2.1, we have proved now the following theorem.

Theorem 3.2.6 (Bakos, Robinson-Cohn, Bhat-Savage). *For all $n \geq 1$, there exists a balanced Gray code of length n, and if n is a power of 2, there exists a totally balanced Gray code of length n.*

The following theorem was stated in [77, Theorem 4].

Theorem 3.2.7. *Let $G(n)$ be a balanced Gray code. Then, $G(n)$ is totally balanced if and only if n is a power of 2.*

Proof. If $G(n)$ is a totally balanced Gray code, then for every bit position $i, 1 \leq i \leq n$, $TC_n(i) = \frac{2^n}{n}$. Because $TC_n(i)$ is an integer for every i, n must be a power of 2. Conversely, let $(p_1, p_2, \ldots, p_n)$ be the transition count spectrum of $G(n)$. Remark that p_i is even for all $i, 1 \leq i \leq n$, and moreover that $|p_j - p_i| \leq 2, 1 \leq i, j \leq n$. Let i be some fixed index value. Suppose that there are l transition counts p_j such that $p_j - p_i = 2$, with $1 \leq l < n = 2^k$. By summation over all j-values, $1 \leq j \leq n$, we obtain $np_i + 2l = 2^n$, and hence $l = 2^{n-1} - \frac{n}{2}p_i = 2^{n-1} - 2^{k-1}p_i = 2^{k-1}(2^{n-k} - p_i)$. Since $1 \leq l < 2^k$, we obtain

$$1 \leq 2^{k-1}(2^{n-k} - p_i) < 2^k \text{ or } \frac{1}{2^{k-1}} \leq 2^{n-k} - p_i < 2.$$

The number $2^{n-k} - p_i$ must be an integer, and hence $2^{n-k} - p_i = 1$. It implies that $p_i = 2^{n-k} - 1$ is an odd integer. This violates the fact that p_i is even. Hence, we may conclude that $p_i = p_j$, for all i and j. So, $G(n)$ is totally balanced. □

3.2.3 A procedure for constructing balanced Gray codes

In this subsection we shall introduce a procedure which exploits Theorem 3.2.2 and Theorem 3.2.3 to construct a balanced Gray code of length n. Our procedure starts with a balanced Gray code of length $n-2$.

Assume we have a balanced Gray code $G(n-2)$ of length $n-2$, with transition counts $TC_{n-2}(i)$, $1 \leq i \leq n-2$. Let $\bar{S}(n-2)$, $n \geq 4$, be the transition sequence of $G(n-2)$. We know that the transition count spectrum of a balanced Gray code $G(n)$ of length n must be equal to an even balanced n-partition $(p_1, p_2, ..., p_n)$ of the integer 2^n, where p_i, $1 \leq i \leq n$, can be calculated using (3.7) or (3.8). Since we remarked already that such an n-partition is unique up to permutations of the

integers p_i, $1 \leq i \leq n$, we may assume, without loss of generality, that $p_{n-1} = p_n$, for $n > 2$. Next, we define

$$b(i) := \begin{cases} \frac{4TC_{n-2}(i)-p_i}{2}, & 1 \leq i \leq n-2, \quad i \neq s_{2^{n-2}}, \\ \\ \frac{4(TC_{n-2}(i)-1)-p_i}{2}, & i = s_{2^{n-2}}. \end{cases} \tag{3.11}$$

Remark 2 For all $n \geq 11$, Lemma 3.2.4 guarantees that $0 \leq b(i) \leq TC_n(i)$, for all i, $1 \leq i \leq n$. Moreover, by inspection we can verify that such integers $b(i)$ can also be realized for all n, $4 \leq n \leq 10$ (cf. Example 3.2.4). Hence, our procedure will be valid for all $n \geq 4$, and is formulated as follows

Construction 3.1

1. Determine a balanced even n-partition $(p_1, p_2, ..., p_n)$ of the integer 2^n using (3.7) or (3.8). If $p_n > p_{n-1}$, permute the integers p_{n-2} and p_n such that in the resulting composition $p_{n-1} = p_n$(in this new composition we have that $p_1 = p_2 = \cdots = p_{n-3} < p_{n-2}$ and $p_{n-1} = p_n$).

2. Calculate $b(i)$ for all $i \in [n-2]$, using (3.11).

3. Define the sequence T consisting of $b(i)$ integers i, $i \in [n-2]$, in $S(n-2)$ which is defined by $\bar{S}(n-2)^k$ for some $k \in [2^{n-2}]$.

4. Apply Theorem 3.2.2, using sequence T defined at step 3.

We shall present some examples how to construct a balanced Gray code of higher dimension from some given balanced Gray code of lower dimension, using Construction 3.1. Examples 3.2.5 and 3.2.6 will give the details of the construction of balanced Gray codes of length 9 and 10 in Example 3.2.4, from balanced Gray codes of length 7 and 8, respectively.

Example 3.2.5. Assume that we want to construct a balanced Gray code of length 9. We start with a balanced Gray code $G(7)$ of length 7 with transition count spectrum $(18, 18, 18, 18, 20, 18, 18)$, and with transition sequence

$$\begin{aligned} \bar{S}(7) = \; & 2,1,3,1,2,1,3,4,3,5,3,4,3,1,4,5,2,1,5,1,4,1,3,4,5,1,2,5,2,4,2,6, \\ & 2,4,2,5,2,1,5,4,3,7,3,4,5,1,2,5,2,4,2,6,2,4,2,5,2,1,5,4,3,1,4,6, \\ & 4,7,4,1,5,7,5,6,5,1,6,7,2,5,7,5,6,5,4,6,7,1,3,7,3,6,3,4,3,5,6,5, \\ & 3,7,3,5,3,7,6,4,6,7,3,7,6,1,6,7,2,7,6,1,6,7,3,7,6,1,6,7,2,7,6,7. \end{aligned}$$

We see that a balanced even n-partition of the integer 2^9 is (56, 56, 56, 56, 56, 58, 58, 58, 58). This partition can be calculated using (3.7) since $2^9 = 9 \cdot 56 + 8$. Using (3.11) we may take $b(1) = \cdots = b(4) = 8$, $b(5) = 12$, $b(6) = 7$, and $b(7) = 5$. So, the length of sequence T is equal to $4 \cdot 8 + 12 + 7 + 5 = 56$. Let T consist of the first 8 occurrences of each of the integers $1, 2, 3$, and 4, the first 12 occurrences of the integer 5, the first

7 occurrences of the integer 6, and the first 5 occurrences of the integer 7 in $S(7)$. By applying Construction 1 with this sequence T as basis, we obtain a balanced Gray code of length 9 with transition count spectrum $(56, 56, 56, 56, 56, 58, 58, 58, 58)$, and with transition sequence

$$\begin{aligned}\bar{S}(9) = \; & 2,1,3,1,2,1,3,4,3,5,3,4,3,1,4,5,2,1,5,1,4,1,3,4,5,1,2,5,2,4,2,6,\\ & 2,4,2,5,2,1,5,4,3,7,3,4,5,1,2,5,2,4,2,6,2,4,2,5,2,1,5,4,3,1,4,6,\\ & 4,7,4,1,5,7,5,6,5,1,6,7,2,5,7,5,6,5,4,6,7,1,3,7,3,6,3,4,3,5,6,5,\\ & 3,7,3,5,3,7,6,4,6,7,3,7,6,1,6,7,2,7,6,1,6,7,3,7,6,1,6,7,2,7,6,8,\\ & 6,7,2,7,6,1,6,7,3,7,6,1,6,7,2,7,6,1,6,7,3,7,6,4,6,7,3,5,3,7,3,5,\\ & 6,5,3,4,3,6,3,7,3,1,7,9,7,1,3,7,3,6,3,4,3,5,6,5,3,7,3,5,3,7,6,4,\\ & 6,7,3,7,6,1,6,7,2,7,6,1,6,7,3,7,6,1,6,7,2,7,6,8,6,7,2,7,6,1,6,7,\\ & 3,7,6,1,6,7,2,7,6,1,6,7,3,7,6,4,6,7,3,5,3,7,3,5,6,5,3,4,3,6,3,7,\\ & 3,1,7,6,4,5,8,5,4,9,4,5,6,5,9,5,8,5,7,5,2,8,2,5,9,5,2,7,9,8,6,1,\\ & 5,8,5,1,9,1,5,6,5,9,5,8,5,7,8,9,5,1,4,9,4,1,8,1,4,7,4,8,4,9,4,6,\\ & 4,1,3,4,9,4,3,1,4,8,4,1,3,4,5,1,2,8,2,1,9,1,2,5,2,4,2,9,2,4,2,8,\\ & 2,4,2,6,2,4,2,8,2,4,2,9,2,4,2,5,2,1,9,1,2,8,2,1,5,4,8,4,9,4,3,9,\\ & 8,7,8,9,3,9,8,4,8,9,5,1,2,9,2,1,8,1,2,5,8,9,2,9,8,4,8,9,2,9,8,6,\\ & 8,9,2,9,8,4,8,9,2,9,8,5,8,9,2,9,8,1,8,9,5,9,8,4,8,9,3,9,8,1,8,9,\\ & 4,9,8,1,8,9,5,9,8,1,8,9,2,9,8,5,8,9,4,9,8,1,8,9,3,9,8,4,8,9,3,9,\\ & 8,5,8,9,3,9,8,4,8,9,3,9,8,1,8,9,2,9,8,1,8,9,3,9,8,1,8,9,2,9,8,9.\end{aligned}$$

Example 3.2.6. A balanced Gray code of length 8 with the following transition sequence $\bar{S}(8)$ is totally balanced with transition count spectrum (32, 32, 32, 32, 32, 32, 32, 32).

$$\begin{aligned}\bar{S}(8) = \; & 1,2,1,3,4,3,1,2,3,2,4,2,1,4,3,5,3,4,1,2,4,6,4,2,1,4,3,5,3,4,1,2,\\ & 4,2,3,5,3,6,3,2,6,5,1,5,6,3,4,6,4,5,4,3,5,6,1,6,5,2,5,6,1,6,5,7,\\ & 5,6,1,6,5,2,5,6,1,6,5,3,4,5,4,6,4,3,6,5,1,8,1,5,6,3,4,6,4,5,4,3,\\ & 5,6,1,6,5,2,5,6,1,6,5,7,5,6,1,6,5,2,5,6,1,6,5,3,4,5,4,6,4,3,6,5,\\ & 1,5,6,2,3,7,3,2,6,8,6,2,3,6,3,8,3,7,3,5,7,8,3,2,4,2,1,8,1,2,4,2,\\ & 7,2,4,2,1,4,7,8,3,8,7,5,7,8,3,8,7,4,1,2,7,2,1,8,1,2,4,8,7,6,7,8,\\ & 4,2,1,8,1,2,7,2,1,4,7,8,3,8,7,5,7,8,3,8,7,4,7,8,1,8,7,2,7,8,4,8,\\ & 7,2,7,8,3,8,7,2,7,8,1,8,7,3,7,8,4,8,7,3,7,8,1,8,7,2,7,8,1,8,7,8.\end{aligned}$$

Since $2^{10} = 10 \cdot 102 + 4$, according to (3.7) the even balanced 10-partition of 2^{10} is equal to $(102, 102, 102, 102, 102, 102, 102, 102, 104, 104)$. By applying (3.11), we may take $b(1) = \cdots = b(7) = 13$, and $b(8) = 11$. Here, the sequence T has length 102. Assume that T is chosen such that it consists of the first 13 occurrences in $S(8)$ of each of the integers $1, ..., 7$ and of the first 11 occurrences of the integer 8 in $S(8)$. By using this sequence T as the basis of Theorem 3.2.2, we obtain a Gray code of length 10 with the following transition sequence $\bar{S}(10)$. The letter a in $\bar{S}(10)$ stands for 10.

$$\begin{aligned}\bar{S}(10) = {} & 1,2,1,3,4,3,1,2,3,2,4,2,1,4,3,5,3,4,1,2,4,6,4,2,1,4,3,5,3,4,1,2,\\ & 4,2,3,5,3,6,3,2,6,5,1,5,6,3,4,6,4,5,4,3,5,6,1,6,5,2,5,6,1,6,5,7,\\ & 5,6,1,6,5,2,5,6,1,6,5,3,4,5,4,6,4,3,6,5,1,8,1,5,6,3,4,6,4,5,4,3,\\ & 5,6,1,6,5,2,5,6,1,6,5,7,5,6,1,6,5,2,5,6,1,6,5,3,4,5,4,6,4,3,6,5,\\ & 1,5,6,2,3,7,3,2,6,8,6,2,3,6,3,8,3,7,3,5,7,8,3,2,4,2,1,8,1,2,4,2,\\ & 7,2,4,2,1,4,7,8,3,8,7,5,7,8,3,8,7,4,1,2,7,2,1,8,1,2,4,8,7,6,7,8,\\ & 4,2,1,8,1,2,7,2,1,4,7,8,3,8,7,5,7,8,3,8,7,4,7,8,1,8,7,2,7,8,4,8,\\ & 7,2,7,8,3,8,7,2,7,8,1,8,7,3,7,8,4,8,7,3,7,8,1,8,7,2,7,8,1,8,7,9,\\ & 7,8,1,8,7,2,7,8,1,8,7,3,7,8,4,8,7,3,7,8,1,8,7,2,7,8,3,8,7,2,7,8,\\ & 4,8,7,2,7,8,1,8,7,4,7,8,3,8,7,5,7,8,3,8,7,4,1,2,7,2,1,8,1,2,4,8,\\ & a,8,4,2,1,8,1,2,7,2,1,4,7,8,3,8,7,5,7,8,3,8,7,4,7,8,1,8,7,2,7,8,\\ & 4,8,7,2,7,8,3,8,7,2,7,8,1,8,7,3,7,8,4,8,7,3,7,8,1,8,7,2,7,8,1,8,\\ & 7,9,7,8,1,8,7,2,7,8,1,8,7,3,7,8,4,8,7,3,7,8,1,8,7,2,7,8,3,8,7,2,\\ & 7,8,4,8,7,2,7,8,1,8,7,4,7,8,3,8,7,5,7,8,3,8,7,4,1,2,7,2,1,8,1,2,\\ & 4,8,7,6,9,6,a,6,7,a,9,8,4,2,1,9,1,2,4,a,4,2,1,8,1,2,a,2,1,9,1,2,\\ & 7,2,1,4,9,4,1,2,a,2,1,4,7,a,9,8,3,9,3,a,3,8,a,9,7,5,9,5,a,5,7,a,\\ & 9,8,3,9,3,a,3,8,a,9,7,4,1,2,4,2,9,2,4,2,1,4,a,4,1,2,4,2,7,2,4,2,\\ & 1,a,1,2,4,2,9,2,4,2,1,8,1,2,4,2,3,9,3,2,4,2,1,a,1,2,4,2,3,8,a,9,\\ & 7,5,3,9,3,5,a,5,3,7,3,a,3,9,3,8,3,6,3,2,6,9,6,2,3,6,3,a,3,6,3,2,\\ & 6,8,6,2,3,a,3,2,6,9,6,2,3,7,3,2,6,5,1,5,6,3,4,6,4,5,4,3,5,6,1,6,\\ & 5,9,5,6,1,6,5,3,4,5,4,6,4,3,6,5,1,5,6,2,3,a,3,2,6,5,1,5,6,3,4,6,\\ & 4,5,4,3,5,6,1,6,5,2,5,6,1,6,5,a,5,6,1,6,5,9,5,6,1,6,5,7,5,6,1,6,\\ & 5,9,5,6,1,6,5,a,5,6,1,6,5,2,5,6,1,6,5,3,4,5,4,6,4,3,6,5,1,a,1,5,\\ & 6,3,4,6,4,5,4,3,5,6,1,6,5,9,5,6,1,6,5,3,4,5,4,6,4,3,6,5,1,8,9,a,\\ & 1,5,6,3,4,6,4,5,a,5,4,6,4,3,6,5,9,5,6,3,4,6,4,5,4,9,a,3,5,a,5,9,\\ & 5,6,9,a,1,a,9,6,9,a,5,a,9,2,9,a,5,a,9,6,9,a,1,a,9,6,9,a,5,a,9,7,\\ & 9,a,5,a,9,6,9,a,1,a,9,6,9,a,5,a,9,2,9,a,5,a,9,6,9,a,1,a,9,6,9,a,\\ & 5,a,9,3,9,a,4,a,9,5,9,a,4,a,9,6,9,a,4,a,9,3,9,a,6,a,9,5,9,a,1,a,\\ & 9,5,9,a,6,a,9,2,9,a,3,a,9,6,9,a,3,a,9,5,9,a,3,a,9,2,9,a,4,a,9,2,\\ & 9,a,1,a,9,4,9,a,3,a,9,5,9,a,3,a,9,4,9,a,1,a,9,2,9,a,4,a,9,6,9,a,\\ & 4,a,9,2,9,a,1,a,9,4,9,a,3,a,9,5,9,a,3,a,9,4,9,a,1,a,9,2,9,a,4,a,\\ & 9,2,9,a,3,a,9,2,9,a,1,a,9,3,9,a,4,a,9,3,9,a,1,a,9,2,9,a,1,a,9,a.\end{aligned}$$

We can verify that the generated Gray code $G(10)$ of length 10 has transition count spectrum $(102, 102, 102, 102, 102, 102, 102, 102, 104, 104)$, and therefore, it is a balanced code. We emphasize that these balanced Gray codes of length 9 and 10 can not be constructed using Bakos' method nor Robinson-Cohn nor Bhat-Savage methods, since their constructions require that the pair of integers s_{i_1}, s_{i_2} and also the pair $s_{i_{l-1}}, s_{i_l}$ are consecutive in $\bar{S}(n)$.

3.2.4 Concluding remarks

Let $2^n = qn+r$, $0 \leq r < n$. If r is even, according to (3.7) a balanced Gray code with transition count spectrum $(\underbrace{q, ..., q}_{n-\frac{r}{2}}, \underbrace{q+2, ..., q+2}_{\frac{r}{2}})$, exists. In particular, if $r = 2$, a balanced Gray code with transition count spectrum $(\underbrace{q, ..., q}_{n-1}, q+2)$ exists. Let $G(n)$ be a balanced Gray code with this transition count spectrum and with transition sequence $\bar{S}(n)$. Let i_0 be the integer in $[n]$ which has transition count equal to $q+2$ in $\bar{S}(n)$. Assume that the transition $s_k = i_0$. Shift $\bar{S}(n)$ cyclicly over k positions to the left. Now, the new transition sequence $\bar{S}(n) := \bar{S}(n)^k$ has the integer i_0 as its closing transition. The *non-complete transition sequence* defined by the last transition sequence $\bar{S}(n)$ will generate a Gray code with transition counts satisfying the property that $|TC_n(i) - TC_n(j)| \leq 1$, for all $i, j \in [n]$. So, we proved the following theorem.

Theorem 3.2.8. *Let $2^n = qn + r$, $0 \leq r < n$. If $r = 0$ or $r = 2$, then a Gray code of length n exists with transition counts $TC_n(i)$, $i \in [n]$, satisfying the property that $|TC_n(i) - TC_n(j)| \leq 1$, for all $i, j \in [n]$.*

It is obvious that every integer n which is a two power satisfies the property that $2^n = qn + 0$, for some q. Below we shall show that every prime $p > 2$ satisfies the property that $2^p = pq + 2$, for some q. However, there are also integers n_0, for instance $341, 561, 645$, which are not prime, but which do have the property that $2^{n_0} = qn_0 + 2$, for some q. These three numbers satisfy the conditions required by the following lemma due to Dodunekov [14][2].

The smallest exponent $e > 0$ for which $b^e \equiv 1 (\text{mod } n)$, where b and n are given positive integers, is called the *multiplicative order* of b modulo n.

Lemma 3.2.9. *Let s be the multiplicative order of 2 modulo n. If $n \equiv 1 (mod\ s)$, then $2^n \equiv 2 (mod\ n)$.*

Proof. Let $2^s = x \cdot n + 1$ and $n = y \cdot s + 1$, for some positive integers x and y. We have that

$$2^n = 2^{y \cdot s+1} = 2 \cdot 2^{y \cdot s} = 2(x \cdot n + 1)^y = 2z \cdot n + 2$$

for some integer z. □

We now immediately have the following theorem.

Theorem 3.2.10. *Let s be the multiplicative order of $2 (mod\ n)$. If $n \equiv 1 (mod\ s)$, then there exists a Gray code of length n with transition counts $TC_n(i)$, $i \in [n]$, such that $|TC_n(i) - TC_n(j)| \leq 1$, for all $i, j \in [n]$.*

Below we shall show that the multiplicative order of 2 modulo p is equal to $p-1$, if p is an odd prime number. To this end, we need the following well-known result.

[2]The author is indebted to prof. dr. S.M. Dodunekov, Bulgarian Academy of Sciences, for drawing his attention to this result, which generalizes the original lemma where n was equal to an odd prime.

Lemma 3.2.11 (Fermat Theorem). *If p is a prime, then for every positive integer a which is not a multiple of p, we have that $a^{p-1} \equiv 1 \ (mod\ p)$.*

Now we are ready to proof the following lemma.

Lemma 3.2.12. *Let p be an odd prime number. Then the multiplicative order of 2 modulo p is equal to $p-1$.*

Proof. In particular, Fermat Theorem says that for every prime $p > 2$ we have that $2^{p-1} \equiv 1 (\text{mod } p)$. Assume that there exists an $l, 0 \leq l \leq p-1$ such that $2^l \equiv 1 (\text{mod } p)$. Let $k + l = p - 1$. Now we have

$$2^{p-1} = 2^{k+l} = 2^k(p \cdot q + 1) = 2^k \cdot q \cdot p + 2^k,$$

for some q. From the statement in the first line of this proof, it follows that $2^k = 1$. Hence, $k = 0$, $l = p - 1$ and the Lemma is proved. □

An immediate consequence of Theorems 3.2.8 and 3.2.10 and of Lemmas 3.2.9 and 3.2.12 is the following corollary.

Corollary 3.2.13. *If n is a two power or an odd prime, then a Gray code of length n exists, such that the transition counts $TC_n(i)$, $i \in [n]$, corresponding to its non-complete transition sequence satisfies the property that $|TC_n(i) - TC_n(j)| \leq 1$, for all $i, j \in [n]$.*

In [3, Section 4] Bhat and Savage posed the following question.

Is it possible, for all n, to construct a Gray code which has the property that for any bit positions i and j, $|TC_n(i) - TC_n(j)| \leq 1$?

Theorem 3.2.8 provides us with a partial answer to this problem.

Because Theorem 3.2.3 can be used to determine which Gray codes w.r.t. their transition count spectra certainly exist by applying Theorem 3.2.2 to a given Gray code, Theorem 3.2.3 provides us with a partial answer to Conjecture 3.1.4.

The conditions to apply Theorem 3.2.2 are less strict than those for the method of Robinson and Cohn in [60]. The latter method requires that the first two elements as well as the last two elements of the sequence T which constitutes the basis of that method, must be consecutive in the original transition sequence of the given Gray code. More in particular, the sequence T obtained in step 3 of Construction 3.1 can be applied directly in Theorem 3.2.2, without additional stipulations as required by the construction of Robinson and Cohn in [60].

3.3 Exponentially balanced Gray codes

The transition counts of a Gray code can be called *exponentially close* if they are all the same power of two, or are all equal to two consecutive powers of two. We call a Gray code with this property an *exponentially balanced* Gray code (cf. [77]), as a

generalization of (totally) balanced Gray codes. Thus, for an exponentially balanced Gray code $G(n)$ one has that the transition count of every bit position i, $1 \leq i \leq n$, is equal to $2^{e(i)}$ for some positive integer $e(i)$, and $|e(i) - e(j)| \leq 1$, $1 \leq i, j \leq n$.

Example 3.3.1. The reflected Gray codes of length $1, 2$ and 3 are examples of exponentially balanced codes since these codes have transition count spectra $(2^1), (2^1, 2^1)$, and $(2^1, 2^1, 2^2)$, respectively. A totally balanced Gray code of length 4 has transition count spectrum $(2^2, 2^2, 2^2, 2^2)$, and hence the code is also exponentially balanced(cf. Example 3.2.2).

In [88], Wagner and West conjectured that exponentially balanced Gray codes exist for all length n. By extending the method of Robinson and Cohn for the construction of Gray codes in [60], van Zanten and Suparta in [77] proved the conjecture of Wagner and West in positive sense. In the following we present a proof based on Bakos' construction of Gray codes which was formulated in Theorem 3.2.2. This proof is much simpler than the one given in [77].

3.3.1 A simple proof for the existence of exponentially balanced Gray codes

As mentioned just in the beginning of this section, the existence of exponentially balanced Gray codes was a longstanding conjecture of Wagner and West in [88]. In [77] we introduced a technique how to construct exponentially balanced Gray codes by applying the Robinson-Cohn Gray code construction [60], thus proving the conjecture of Wagner and West,

Theorem 3.3.1. *For every $n \geq 1$, there exists an n-bit exponentially balanced Gray code, and if n is a power of two, there exists an n-bit totally balanced Gray code.*

Here, we shall present a proof using Theorem 3.2.2. The proof is constructive like the proof in [77], but much simpler.

Proof. We accomplish the proof using the principle of mathematical induction. It is obvious that Gray codes of length $1, 2$, and 3 are exponentially balanced. Assume that an exponentially balanced Gray code $G(n)$ of length $n \geq 3$ exists with transition count spectrum

$$(TC_n(1), TC_n(2), ..., TC_n(n)) := (\underbrace{2^v, ..., 2^v}_{k}, \underbrace{2^{v+1}, ..., 2^{v+1}}_{n-k}), \tag{3.12}$$

for some k with $0 \leq k < n$. We distinguish two cases: $n - k > 1$ and $n - k = 1$.

Case I. $n - k > 1$. This case implies that $TC_n(n-1) = TC_n(n) = 2^{v+1}$. Let $\bar{S}(n)$ be the transition sequence of $G(n)$, and let i be some integer in $\{0, 1, ..., 2^n - 1\}$ such that the closing transition of $\bar{S}(n)^i$ is the integer n. Take a sequence T from the cyclically

shifted transition sequence $\bar{S}(n) := \bar{S}(n)^i$ with length $l := 2^{v+2} - 2$, consisting of $2^{v+1} - 2$ occurrences of integer n and all 2^{v+1} occurrences of integer $n-1$. Here, $b(1) = \cdots = b(n-2) = 0$, $b(n-1) = 2^{v+1}$, and $b(n) = 2^{v+1} - 2$. Apply Theorem 3.2.2 using this sequence T. Then we obtain a Gray code of length $n+2$ with transition counts satisfying the following (cf. eq. (3.2))

$$TC_{n+2}(i) := \begin{cases} 2^{v+2}, & \text{for all } i \in \{1, ..., n+2\} \backslash \{k+1, ..., n-2\}, \\ \\ 2^{v+3}, & \text{for all } i \in \{k+1, ..., n-2\}. \end{cases}$$

Thus, the resulting Gray code of length $n+2$ is exponentially balanced with transition count spectrum $(\underbrace{2^{v+2}, \ldots, 2^{v+2}}_{k+4}, \underbrace{2^{v+3}, \cdots, 2^{v+3}}_{n-k-2})$.

Notice that if $n - k = 2$ or equivalently $n + 2 = 2^{n-v}$ (a power of two), the resulting $(n+2)$-bit Gray code is *totally* balanced with transition count spectrum $(2^{v+2}, 2^{v+2}, ..., 2^{v+2})$.

Case II. $n - k = 1$. The transition count spectrum (3.12) now becomes

$$(TC_n(1), TC_n(2), ..., TC_n(n)) := (\underbrace{2^v, \ldots, 2^v}_{n-1}, 2^{v+1}). \tag{3.13}$$

Here, the transition count of integer n is equal to 2^{v+1}. Again we assume that n is the closing transition of $\bar{S}(n)^i$, for some integer i, $0 \leq i \leq 2^n - 1$. Take a sequence T from $\bar{S}(n) := \bar{S}(n)^i$ consisting of only $2^{v+1} - 2$ occurrences of integer n, and then apply Theorem 3.2.2. The resulting Gray code of length $n+2$ will have transition counts

$$TC_{n+2}(i) := \begin{cases} 2^{v+1}, & \text{if } i = n+1, n+2, \\ \\ 2^{v+2}, & \text{if } i \in \{1, ..., n\}. \end{cases}$$

Again the resulting Gray code of length $n+2$ is exponentially balanced with transition count spectrum $(2^{v+1}, 2^{v+1}, \underbrace{2^{v+2}, \ldots, 2^{v+2}}_{n})$.

We see that in each case the produced Gray code is exponentially balanced. Since exponentially balanced Gray codes of length 1, 2, and 3 exist, the Theorem is proved now by the principle of mathematical induction. □

Next we shall show an example how we exploit the proof of Theorem 3.3.1 for constructing an exponentially balanced Gray code from another exponentially balanced Gray code of smaller length.

Example 3.3.2. Consider an exponentially balanced Gray code of length 5 with transition count spectrum $(2^2, 2^2, 2^3, 2^3, 2^3)$, and with transition sequence $\bar{S}(5)$

$$1, 4, 5, 3, 5, 4, 2, 4, 5, 3, 5, 4, 1, 4, 5, 3, 5, 4, 2, 4, 3, 5, 3, 4, 2, 3, 1, 3, 2, 3, 1, 5.$$

Assume that we want to construct an exponentially balanced Gray code of length 7 based on the exponentially balanced Gray code of length 5. According to the transition count spectrum and to the transition sequence of the Gray code, we can see that we are in Case II of the proof of Theorem 3.3.1, and that the transition sequence of $\bar{S}(5)$ already has the integer 5 as closing transition. Take $b(1) = b(2) = b(3) = 0$, $b(4) = 2^3 = 8$, and $b(5) = 2^3 - 2 = 6$. Thus, the sequence T has length $8 + 6 = 14$. Furthermore, we may take the first eight integers 4 and the first six integers 5 in $\bar{S}(5)$ as elements of the sequence T. Applying Theorem 3.2.2 yields a Gray code with transition sequence $\bar{S}(7)$

$$\begin{aligned}
&1, 4, 5, 3, 5, 4, 2, 4, 5, 3, 5, 4, 1, 4, 5, 3, 5, 4, 2, 4, 3, 5, 3, 4, 2, 3, 1, 3, 2, 3, 1, 6,\\
&1, 3, 2, 3, 1, 3, 2, 7, 2, 3, 1, 3, 2, 3, 1, 6, 1, 3, 2, 3, 1, 3, 2, 4, 3, 5, 3, 6, 3, 5, 3, 7,\\
&3, 5, 3, 4, 2, 7, 2, 6, 2, 4, 6, 7, 5, 3, 7, 3, 6, 3, 5, 6, 7, 4, 1, 7, 1, 6, 1, 4, 6, 7, 5, 3,\\
&7, 3, 6, 3, 5, 6, 7, 4, 2, 7, 2, 6, 2, 4, 6, 7, 5, 3, 7, 3, 6, 3, 5, 6, 7, 4, 1, 7, 1, 6, 1, 7.
\end{aligned}$$

It is easy to verify that the produced Gray code is exponentially balanced with transition count spectrum $(2^4, 2^4, 2^5, 2^4, 2^4, 2^4, 2^4)$.

4

More Binary Gray Codes with Special Properties

This chapter consists of two sections. In the first section we discuss a method for the construction of Gray codes with maximum crossover Hamming distance. For n even, this type of Gray codes is also called *complementary* Gray codes in [32]. Two codewords in a Gray code $G(n)$ of length n are said to *crossover* each other in a cyclic n-bit Gray code if their distance in the list is equal to 2^{n-1}. If any two crossover codewords in a cyclic n-bit Gray code have the same Hamming distance k and if this Hamming distance is maximal for fixed n, the code is called a Gray code with *maximum crossover Hamming distance (MCHD)* k. We introduce a simple technique for constructing a class of Gray codes with this property. We also derive conversion rules between codewords and their indices in the list(*index problem*), and we solve the separability problem for this type of Gray codes.

The second part of this chapter addresses a problem of Wilmer and Ernst in [89] about a construction of an n-bit Gray code which induces the *undirected* complete graph K_n. For every $n > 0$ we introduce a technique how to construct Gray codes $G(n)$ which induce complete graphs K_n. This technique is also developed by applying the Gray construction formulated as Theorem 3.2.2 in Chapter 3. At the end of this section we state a thusfar unsolved problem about the existence of Gray codes which induce *directed* complete graphs.

4.1 A class of Gray codes with MCHD

Because of certain applications [8, 41, 64] and also for the sake of mathematical interest [18, 20, 30, 87, 80, 79], Gray codes have been designed to satisfy some additional requirements. Some examples of constraints considered are: restricting where bits

can change [8], restricting to the case that any two complementary codewords are at list distance equal to the length of the Gray code [30], or that such codewords are at maximal list distance [41], or requiring the same number of changes for each bit [41, 64, 87].

In this section, the term Gray codes stands for cyclic Gray codes. As already defined in Chapter 1, the *cyclic list distance* of two codewords $\mathbf{x}_i$ and $\mathbf{x}_j$, denoted by $D(\mathbf{x}_i, \mathbf{x}_j)$, is equal to

$$D(\mathbf{x}_i, \mathbf{x}_j) := min\{|j-i|, 2^n - |j-i|\}.$$

The codewords $\mathbf{x}_i$ and $\mathbf{x}_j$ are said to *crossover* each other in the Gray code $G(n)$, if their list distance $D(\mathbf{x}_i, \mathbf{x}_j) = 2^{n-1}$. If crossover codewords in a Gray code all have the same Hamming distance k, then one says that the Gray code is with *crossover Hamming distance* k. If k is maximal with respect to this property (for fixed n), then one says that the Gray code is with *maximum crossover Hamming distance*(*MCHD*) (See Figure 4.1 for an example). In this section we consider this type of Gray codes.

The existence of Gray codes with MCHD was proven by Knuth in [32]. The proof is based on the existence of *monotone Gray codes*, i.e. Gray codes in which consecutive pairs of codewords of weights i, $i+1$ precede those of weights j, $j+1$ for all $i < j \leq n$, which was demonstrated by Savage and Winkler in [62]. Here, the weight of a codeword is equal to the number of non-zero bits in the codeword. It is evident from [32] that Gray codes with MCHD can be constructed from monotone Gray codes. However, the construction of a monotone Gray code itself in [62] is not straightforward. Ludman in [41] presents a technique for generating Gray codes with MCHD, but his technique is rather cumbersome when applying it for large values of the code length n. In this paper we propose a simple construction which can be applied easily for any Gray code length.

The following lemma is obvious.

Lemma 4.1.1. *Let* $\mathbf{g}$ *and* $\mathbf{h}$ *be two complementary codewords in a Gray code of length* n*. Then their list distance,* $D(\mathbf{g}, \mathbf{h})$*, is equal to* $2l-1$ *if* n *is odd, and equal to* $2l$ *if* n *is even, for some* $l, 1 \leq l \leq 2^{n-2}$*.*

Lemma 4.1.1 implies that the maximal list distance that can be attained by a pair of n-bit complementary codewords is equal to $2^{n-1}-1$ for n odd, and equal to 2^{n-1} for n even. Consequently, the maximal possible value of k for which a Gray code $G(n)$ with MCHD of k can exist is equal to n if n even, and equal to $n-1$ if n odd.

4.1.1 Construction rules

Like in Chapter 3, let T^n be the non-complete transition sequence of the reflected Gray code of length n. It is well known that the sequence T^n, n is the complete transition sequence of the n-bit reflected Gray code. Our technique for constructing an n-bit Gray code with MCHD is based on the transition sequence T^n, and is defined by the following rules.

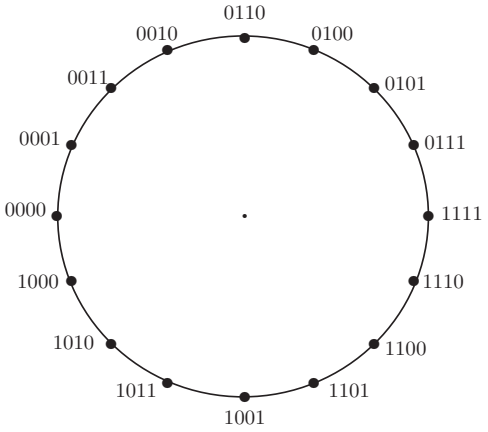


Figure 4.1: A Gray code of length 4 with MCHD

Construction 4.1

1. $TS(n) := T^n, n$, if $n = 1, 2, 3$.
2. For $n \geq 4$ define:

   (a)

$$\begin{aligned} U(n-1) := \quad & T^{n-2}, n-1, T^{n-4}, \mathbf{n-2}, T^{n-4}, \mathbf{n-3}, T^{n-4}, \mathbf{n-2}, \\ & T^{n-6}, \mathbf{n-4}, T^{n-6}, \mathbf{n-5}, T^{n-6}, \mathbf{n-4}, \ldots, \mathbf{2}, \mathbf{1}, \mathbf{2}, \end{aligned}$$

   for n even, and

$$\begin{aligned} U(u-1) := \quad & T^{n-2}, n-1, T^{n-4}, \mathbf{n-2}, T^{n-4}, \mathbf{n-3}, T^{n-4}, \mathbf{n-2}, \\ & T^{n-6}, \mathbf{n-4}, T^{n-6}, \mathbf{n-5}, T^{n-6}, \mathbf{n-4}, \ldots, 1, \mathbf{3}, 1, \mathbf{2}, 1, \mathbf{3}, 1, \end{aligned}$$

   for n odd.

   (b) $TS(n) := U(n-1), n, U(n-1), n$.

Notice that the sequence $U(n-1)$ is obtained from the non-complete transition sequence T^{n-1} of the reflected Gray code of length $n-1$ by interchanging the pairs of integers $n-2$ and $n-3$, of integers $n-4$ and $n-5$, etc. after the occurrence of the integer $n-1$.

Example 4.1.1.

$$\begin{aligned} TS(4) = \quad & T^2, 3, \mathbf{2}, \mathbf{1}, \mathbf{2}, 4, T^2, 3, \mathbf{2}, \mathbf{1}, \mathbf{2}, 4 \\ = \quad & 1, 2, 1, 3, \mathbf{2}, \mathbf{1}, \mathbf{2}, 4, 1, 2, 1, 3, \mathbf{2}, \mathbf{1}, \mathbf{2}, 4, \end{aligned}$$

and

$$TS(5) = \begin{array}{l} 1,2,1,3,1,2,1,4,1,\mathbf{3},1,\mathbf{2},1,\mathbf{3},1,5, \\ 1,2,1,3,1,2,1,4,1,\mathbf{3},1,\mathbf{2},1,\mathbf{3},1,5. \end{array}$$

The lists of codewords of the Gray codes $G(4)$ and $G(5)$ with transition sequences S_4 and S_5 are shown in Fig.4.2.a and in Fig.4.2.b, respectively.

0000	1111
0001	1110
0011	1100
0010	1101
0110	1001
0100	1011
0101	1010
0111	1000

a.

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

b.

Figure 4.2:

We observe that the list distance between any two complementary codewords is equal to $2^3 = 8$, if $n = 4$(Fig.4.2.a), and equal to $2^4 - 1 = 15$, if $n = 5$ (Fig.4.2.b). According to Lemma 4.1.1, we conclude that these two Gray codes are with MCHD 4. Below, we shall show that each Gray code which has a transition sequence $TS(n)$ obtained by using the Construction 4.1, is a Gray code with MCHD of n, if n is even, and of $n-1$, if n is odd.

Referring to Lemma 2.3.1 in Chapter 2, we can immediately prove the following lemma.

Lemma 4.1.2. *Every non-empty subsequence S of the sequence $U(n-1)$ in Construction 4.1 contains at least one integer which occurs an odd number of times.*

Proof. Let S be a non-empty subsequence of $U(n-1)$. Consider the sequence T^{n-k}, the non-complete transition sequence of the reflected Gray code of length $n-k$, for $k \in \{1, 2, \ldots, n-1\}$. Any non-empty subsequence of T^{n-k} will contain at least one integer which occurs an odd number of times. Hence, if S is a subsequence of T^{n-k}, then S contains at least one integer which occurs an odd number of times. Now assume that S is not a subsequence of T^{n-k} for any $k \in \{1, 2, \ldots, n-1\}$. Assume that the largest integer contained in S is equal to $n-k$, for some $k \in \{1, 2, \ldots, n-1\}$. It is clear that if $k = n-1$, S is a singleton and therefore, contains the integer 1 an odd number of times. If $k = 1$, then S will contain at least the integer $n-1$ an odd number of times. Now assume $k = 2l$ for some l. According to the pattern of $U(n-1)$, if $n-2l$ occurs an even number of times, then at least the integer $n-2l-1$ occurs an odd number of times. Finally, if the largest integer contained in S is equal to $n-2m-1$ for some m, then it is obvious, due to the pattern of $U(n-1)$, that S at least contains the integer $n-2m-1$ an odd number of times. □

Due to Lemmas 4.1.2 and 2.3.1, the following Lemma can immediately be proved.

Lemma 4.1.3. *For every n, the sequence $TS(n)$ obtained by applying Construction 4.1, is a transition sequence of a cyclic Gray code.*

Moreover, the Gray code having $TS(n)$ as its transition sequence is really with MCHD, as is formulated below.

Lemma 4.1.4. *The list distance of any two complementary codewords in a Gray code of length n which has $TS(n)$ as its transition sequence is equal to 2^{n-1} if n is even, and equal to $2^{n-1}-1$ if n is odd.*

Proof. Assume that S is a subsequence of $TS(n)$ which constitutes a transition sequence from a codeword to its complement. It implies that S contains n distinct integers which occur an odd number of times. Therefore, S contains all transition numbers, including $n-1$ and n. We also know that for all $k, 1 \leq k \leq n$, T^k only contains the integer k an odd number of times. Based on these observations, the sequence S will be equivalent to the sequence

$T^{n-2}, n-1, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-2, T^{n-6}, n-4, T^{n-6}$, $n-5, T^{n-6}$, $n-4$, ..., 2, 1, 2, n

for n even, and to

$T^{n-2}, n-1, T^{n-4}, n-2, T^{n-4}, n-3, T^{n-4}, n-2, T^{n-6}, n-4, T^{n-6}$, $n-5$, T^{n-6}, $n-4$, ..., 1, 3, 1, 2, 1, 3, 1, n,

for n odd,
when omitting the front integer 1 from T^{n-2}, and up to cyclic shifting.

The length of these subsequences equals 2^{n-1} and $2^{n-1}-1$ respectively. Because of Lemma 4.1.1, this is the maximal list distance which can be attained. □

Since we know that any pair of consecutive codewords in a Gray code differ in only one bit, one can easily derive the following theorem from Lemma 4.1.4.

Theorem 4.1.5. *The Gray code $G(n)$ which has transition sequence $TS(n)$ is with MCHD of value n if n is even, and of value $n-1$ if n is odd.*

4.1.2 Index system of a Gray code with MCHD

In this section, a Gray code with MCHD is referred to as a Gray code having $TS(n)$ as its transition sequence. Since a Gray code with MCHD is obtained by modifying the transition sequence of the reflected Gray code, we may expect that the *index system* of these Gray codes (i.e. the rules to convert a codeword $\mathbf{x}_i$ to its index i and vice versa) are also closely related.

Let x^c (resp. $\mathbf{x}^c$) be the complement of the bit x (resp. the vector $\mathbf{x}$). Let $\mathbf{g}$ be a codeword in the reflected Gray code of length n with n even, and let $\mathbf{h}$ be its counterpart in the Gray code with MCHD. We have the following conversion rules.

1. $\mathbf{h} = 00\mathbf{a}$, if and only if $\mathbf{g} = 00\mathbf{a}$;
2. $\mathbf{h} = 01\underbrace{11..11}_{2k}xy\mathbf{b}$, if and only if $\mathbf{g} = \begin{cases} 01\underbrace{00..00}_{2k}xy\mathbf{b} & \text{if } x \neq y \\ 01\underbrace{00..00}_{2k}x^cy^c\mathbf{b} & \text{if } x = y = 0 \end{cases}$, where k is the largest integer satisfying this condition;
3. $\mathbf{h} = 11\mathbf{a}$, if and only if $\mathbf{g} = 11\mathbf{a}^c$; and
4. $\mathbf{h} = 10\underbrace{00..00}_{2k}xy\mathbf{c}$, if and only if $\mathbf{g} = \begin{cases} 10\underbrace{00..00}_{2k}xy\mathbf{c}^c & \text{if } x = y = 1 \\ 10\underbrace{00..00}_{2k}x^cy^c\mathbf{c}^c & \text{if } x \neq y \end{cases}$, where k is the largest integer satisfying this condition.

For n odd, the conversion is accomplished by applying the same procedure for the first $(n-1)$ bits from the left and by keeping the last bit fixed.

The above convention rules can be understood by observing the transition sequence $TS(n)$ which we use to construct Gray code $G_{nop}(n)$. For instance, codewords of $G_{ref}(n)$ and their counterpart in $G_{nop}(n)$ will remain the same before passing through the first integer $n-1$ in $TS(n)$. In this case we have rule 1.

When we have passed the first integer $n-1$, the differences between codewords in $G_{nop}(n)$ and their counterpart in $G_{ref}(n)$ depend on whether we have passed or not the pairs of integers $n-2$ and $n-3$, $n-4$ and $n-5$, etc. In this case we have to apply the second rule.

If we passed the integer n but did not yet pass the second integer $n-1$, it implies that all pairs of integers $n-2$ and $n-3$, $n-4$ and $n-5$, etc. have interchanged. So, the parities of the occurrences of all integers in $TS(n)$ and in T^n, n are different. This is the case rule 3 applies to.

If both the first integer n and the second integer $n-1$ have been passed, the differences between codewords in $G_{nop}(n)$ and their counterpart in $G_{ref}(n)$ depend on whether we have passed or not the pairs of integers $n-2$ and $n-3$, $n-4$ and $n-5$, etc. Now rule 4 applies.

We remind the reader of the conversion rules between $\mathbf{g} = g_ng_{n-1}...g_1$ in the n-bit reflected Gray code and its index i, written as $\mathbf{i} = i_ni_{n-1}...i_1$, in binary representation(see Subsection 2.3.3):

$$g_n = i_n, \quad g_k = i_{k+1} \oplus i_k \quad \text{for } k = 1, 2, ..., n-1, \tag{4.1}$$

$$i_n = g_n, \quad i_k = g_k \oplus i_{k+1} \quad \text{for } k = 1, 2, ..., n-1. \tag{4.2}$$

Example 4.1.2. Let us determine the codeword with index 12 in the 4-bit Gray code with MCHD. The binary representation of 12 is 1100. Using (4.1), we find that the codeword $\mathbf{g} = 1010$ has index 12 in the 4-bit reflected Gray code. This codeword $\mathbf{g}$ is of type 4, hence its counterpart $\mathbf{h}$ is equal to 1001. So, the codeword with index 12 in the 4-bit Gray code with MCHD is equal to 1001.

Example 4.1.3. Assume that we want to obtain the index of $\mathbf{h} = 01110$ in the 5-bit Gray code with MCHD. This codeword is of type 2. So, $\mathbf{g} = 01000$. By using (4.2), we have that the binary representation of $\mathbf{g}$ is equal to 01111, which stands for the value of 15. Hence, the index of $\mathbf{h} = 01110$ is equal to 15.

4.1.3 Separability

As has been mentioned before, the list distance of any pair of complementary codewords in the n-bit Gray code with MCHD is equal to 2^{n-1} if n is even, and equal to $2^{n-1} - 1$ if n is odd. It is clear that the minimum distance of any pair of codewords with Hamming distance $m = 1$ or 2, is equal to 1 and 2 respectively. The following theorem gives the separability function for all other values of m, $2 < m < n$.

Theorem 4.1.6. *If the codewords* $\mathbf{g}$ *and* $\mathbf{h}$ *have Hamming distance* m, $2 < m < n$, *in the* n*-bit Gray code having* $TS(n)$ *as its transition sequence, then their list distance* $D(\mathbf{g}, \mathbf{h})$ *satisfies*

$$(i) \text{ if } n \text{ is odd, then } D(\mathbf{g}, \mathbf{h}) \geq \begin{cases} \lceil \frac{2^m}{3} \rceil & \text{if } m \text{ odd,} \\ 2^{m-2} + 2 & \text{if } m \text{ even,} \end{cases}$$

$$(ii) \text{ if } n \text{ is even, then } D(\mathbf{g}, \mathbf{h}) \geq \begin{cases} \lceil \frac{2^m}{3} \rceil & \text{if } m \text{ even,} \\ 2^{m-2} + 1 & \text{if } m \text{ odd.} \end{cases}$$

To prove the Theorem we need the following lemma.

Lemma 4.1.7. *Let* T^n *be the non-complete transition sequence of the standard Gray code, and let* p *and* m *be such that* $n \geq p \geq m > 1$.

(i) If $V(p, m-1) := p, 1, \ldots, m-1$ *is a subsequence of* T^n *such that* $m-1$ *is the first such integer to the right of* p, *then* $V(p, m-1)$ *contains precisely* m *distinct integers* $1, 2, \ldots, m-1, p$, *and* $m-2, m-1$ *and* p *are the only integers which occur an odd number of times.*

(ii) If $V(p, m-1, m-3, \ldots, m-2k-1 = 1 \text{ or } 2) := p, 1, \ldots, m-1, 1, \ldots, m-3, 1 \ldots, m-2k-1$, *is a subsequence of* T^n *such that* $m-i$ *is the first such integer to the right of* $m-i+2$ *for all relevant values of* i, *then* $V(p, m-1, m-3, \ldots, m-2k-1)$ *contains precisely* m *distinct integers* $1, 2, \ldots, m-1, p$, *and all these integers occur an odd number of times.*

(iii) $V(p, m-1, m-3, \ldots, m-2k-1)$ *is a shortest subsequence satisfying the property in* (*ii*)*, and its length is* $\lceil \frac{2^m}{3} \rceil$.

Proof. (i) It is obvious by the structure of T^n and by the definition of $V(p, m-1)$ that this subsequence contains $1, 2, \ldots, m-1, p$ and no other integers. It is also obvious that p and $m-1$ occur just once. Moreover, since between any two integers $m-2$ there occurs at least one integer greater than $m-2$, we also have that $m-2$ occurs just once. It follows that $V(p, m-1)$ has the structure $V(p, m-1) = p, T^{m-3}, m-2, T^{m-3}, m-1$. Since T^{m-3} occurs twice, all integers less than $m-2$ occur an even number of times.

(ii) This property follows by building up $V(p, m-1, m-3, \ldots, m-2k-1)$ as a concatenation of $V(p, m-1)$,$V(m-1, m-3)$,..., $V(m-2k+1, m-2k-1)$ and omitting double integers $m-1, m-3 \ldots, m-2k+1$.

(iii) From the structure of $V(p, m-1, m-3, \ldots, m-2k-1)$ as a concatenation of the various subsequences $V(a, b)$, we infer that its length is equal to

$$(2^{m-2} + 2^{m-4} + \cdots + 1) + 1 = \frac{2^m - 1}{3} = \frac{2^m + 2}{3}$$

for even values of m, while for odd values of m the length is

$$(2^{m-2} + 2^{m-4} + \cdots + 2) + 1 = 2\frac{2^{m-1} - 1}{3} = \frac{2^m + 1}{3}.$$

If there were a shortest subsequence with the same property, we should have a violation of the separability theorem for the standard Gray code.

□

Now we are ready to prove Theorem 4.1.6.

Proof. We shall only prove the Theorem for the case that n is odd, and omit case n even since it is similar. Let us consider the transition sequence $TS(n)$, $n > 3$, and an integer $m, 2 < m < n$. Comparing the constructions of $TS(n)$ and of T^n, n, we can see that the differences occur in elements which tend to the integer n after the occurrences of the integer $n-1$.

Let $V'(p, m-1) := m-1, 1, \ldots, 1, p$, $1 < m \leq p \leq n$, be the subsequence of $TS(n)$ such that $m-1$ is the first such integer to the left of p. We can see that if b is an odd integer, $1 < b < n-1$, then after the occurrences of the integer $n-1$ in $TS(n)$, we have a subsequence $b, T^{b-2}, b-1, T^{b-2}, b$. For the sake of clearness, we replace the first b by b_1 and the second b by b_2. Now we have that $V'(p, m-1) = V(p, m-1)^R$ if $p \neq b_2$, for all $p < n-2$, where R stands for reversed order.

Let $V'(p, m-1, m-3, \ldots, m-2k-1 = 1 \text{ or } 2) := m-2k-1, \ldots, 1, m-3, 1, \ldots, 1,$ $m-1,\ 1 \ldots, 1, p$, be a subsequence of $TS(n)$ such that $m-i$ is the first such integer to the left of $m-i+2$ for all relevant values of i. Then, we also have that $V'(p, m-1, m-3, \ldots, m-2k-1 = 1 \text{ or } 2) = V(p, m-1, m-3, \ldots, m-2k-1 = 1 \text{ or } 2)^R$, if $p \neq b_2$, for all $p < n-2$.

Now consider the subsequence $V(b_2, m-1, m-3, \ldots, m-2k-1)$. Assume that $b_2 = n-2$. If $m = n-2$ then $V(n-2, m-1, m-3, \ldots, n-2k-1)$ contains $m+1$ distinct integers $1, 2, ..., n-3$, $n-2$ and the integer n. But only 5 integers n, $n-2, n-3$, $n-4$ and 1 occur an odd number of times. The length of this subsequence is equal to $2 \cdot |T^{n-4}| + |V(n-3, n-5, n-7, ..., 2)| + 2 = 2 \cdot (2^{n-4} - 1) + \lceil \frac{2^{n-3}}{3} \rceil + 2$, which is greater than $\lceil \frac{2^5}{3} \rceil$ for all $n > 5$. Here, $|S|$ stands for the length of a sequence S.

The case $b_2 = n-2$ with $m < n-2$ can be dealt with similarly as the general case $b_2 < n-2$ with $m < b_2$, as follows.

Consider a subsequence $V(b_2, m-1, m-3, \ldots, n-2k-1)$. The sequence $V(p = b_2, m-1, m-3, \ldots, n-2k-1)$ might be different from $V(p, m-1, m-3, \ldots, n-2k-1)$ in the cases $m = b_2 - 1$ and $m = b_2 - 2$. First assume that $m = b_2 - 1$. Notice that the integer $b_2 - 2$ appears prior to the integer $b_2 - 3$. Here, the positions of the integers $b_2 - 2$ and $b_2 - 3$ are interchanged compared to their positions in T^n. It implies that the sequence $V(b_2, m-1, m-3, \ldots, n-2k-1)$ does not contain the integer $b_2 - 3$, and hence this sequence only contains $m - 1 = b_2 - 2$ integers $1, 2, \ldots, m-3, m-1$, and b_2. The length of this sequence is equal to $|T^{b_2-4}| + |T^{b_2-5}| + \cdots + |T^3| + |T^1| + |b_2, b_2-2, b_2-4, \ldots, 3| = 2^{b_2-4} - 1 + 2^{b_2-5} - 1 + \cdots + 2^2 - 1 + 1 + \frac{b_2-1}{2} = 2^{b_2-3} - (b_2 - 3) + \frac{b_2-1}{2}$, which is at least equal to $\lceil \frac{2^{b_2-2}}{3} \rceil$ (the minimum list distance in T^n for any two codewords with Hamming distance $m - 1 = b_2 - 2$).

Secondly, assume that $m = b_2 - 2$. The sequence $V(b_2, m-1, m-3, \ldots, m-2k-1)$ contains $m+1$ integers $1, 2, \ldots, m = b_2 - 2$, together with the integer b_2, but the integer $m - 2 = b_2 - 4$ occurs an even number of times. The length of this sequence is equal to $2 \cdot |T^{b_2-4}| + |T^{b_2-6}| + |T^{b_2-8}| + \cdots + |T^3| + |T^1| + |b_2, b_2-2, b_2-3, b_2-5, b_2-7, \ldots, 4, 2| = 2^{b_2-4} - 1 + 2^{b_2-4} - 1 + 2^{b_2-6} - 1 + 2^{b_2-8} - 1 + \cdots + 8 + 2 + \frac{b_2-3}{2} = 2^{b_2-4} - 1 + \frac{2}{3}(2^{b_2-3} - (b_2 - 4)) + 2 + \frac{b_2-3}{2} = 2^{b_2-4} + \frac{2^{b_2-2}}{3} + \frac{8-b_2}{3}$, which is greater than $\lceil \frac{2^{b_2-2}}{3} \rceil$ for all relevant values of b_2.

The case $p = n-1$ in the subsequences $V(p, m-1, m-3, \ldots, m-2k-1)$ and $V'(p, m-1, m-3, \ldots, m-2k-1)$ can be dealt with by a similar procedure as we used for case $p = b_2$, since the integer $n-1$ can be considered as b_2 w.r.t. the occurrences of integers following b_2.

We see that in any of the above cases, we have that the length of the resulting subsequences which contain exactly m integers an odd number of times is at least $\lceil \frac{2^m}{3} \rceil$. However, the following observation shows that for m *even* there exists a shorter subsequence containing exactly m distinct integers which occur an odd number of times. The subsequence is constructed as follows.

Consider the concatenated subsequence $1, V'(n, m-2)$ with $1 < m < n$ and m even. This sequence will contain m distinct integers and all these integers occur an odd number of times. The length of this sequence is equal to $2 \cdot |T^{m-3}| + 4 (= |1, n, m-2, m-1|) = 2 \cdot (2^{m-3} - 1) + 4 = 2^{m-2} + 2$. □

Remark The separability function $b(m)$ for the reflected binary Gray code, is equal to $\lceil \frac{2^m}{3} \rceil$, for *all* values of m (cf. [80, 79] and also Section 2.1.3 of this thesis). So, we see that the additional structure of MCHD is at the expense of the separability

power, since $\lceil \frac{2^m}{3} \rceil \geq 2^{m-2} + 2$, for $m > 3$.

4.2 A construction of Gray codes inducing complete graphs

The *graph* $\mathcal{G}_{G(n)}$ induced by a Gray code $G(n)$ has vertex set $\{1, 2, ..., n\}$ and edge set $\{\{s_i, s_{i+1}\} : 1 \leq i \leq 2^n - 2\}$, where s_i is the transition between codewords $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ of $G(n)$. The vertices of $\mathcal{G}_{G(n)}$ correspond to bit positions, and vertices i and j are adjacent when bit positions i and j flip consecutively when running through the list $G(n)$. We emphasize here that the graph $\mathcal{G}_{G(n)}$ is considered as an *undirected simple* graph, i.e. if a consecutive pair $\{u, v\}$ occurs more than once in the transition sequence $S(n)$ or in the complete transition sequence $\bar{S}(n)$, there is only one edge joining vertices u and v in $\mathcal{G}_{G(n)}$. If $G(n)$ is cyclic, the *cyclic* graph $\bar{\mathcal{G}}_{G(n)}$ induced by $G(n)$ is the graph $\mathcal{G}_{G(n)}$ completed with the edges $\{s_{2^n-1}, s_{2^n}\}$ and $\{s_{2^n}, s_1\}$ which may be already contained in $\mathcal{G}_{G(n)}$. The graphs $\mathcal{G}_{G(n)}$ and also $\bar{\mathcal{G}}_{G(n)}$ induced by the binary reflected Gray code of length n is the star S_n with central vertex 1 and edges $\{1, i\}$ for all $i \in \{2, \ldots, n\}$. In this section we shall mainly deal with cyclic graphs $\bar{\mathcal{G}}_{G(n)}$ which are identical to the complete graph K_n.

If a Gray code G induces a graph $\mathcal{G}$ with transition sequence S, then we also say that the transition sequence S induces the graph $\mathcal{G}$. A Gray code is called a *$\mathcal{G}$-code* (with a *$\mathcal{G}$-transition sequence*), if it induces a subgraph of $\mathcal{G}$ with the same number of vertices (cf. [8]).
A graph $\mathcal{G}$ is called *completely Gray* if there is a $\mathcal{G}$-transition sequence of a Gray code starting from any vertex of $\mathcal{G}$. It is obvious that a cyclic Gray code induces a completely Gray graph. A graph $\mathcal{G}$ is *Gray connected* if for every pair of vertices u and v there exists a $\mathcal{G}$-transition sequence of a Gray code starting at u and ending at v. A binary reflected Gray code is an example of a Gray code inducing a graph which is completely Gray as well as Gray connected.

The following two facts were proved in [8].

Theorem 4.2.1. *For any two leaves u and v of the star S_n, there exists a cyclic S_n-transition sequence of a Gray code of length n which starts at u and ends at v if and only if $u \neq v$.*

Since the complete graph K_n contains the star S_n as a spanning tree, the following fact is an immediate consequence of Theorem 4.2.1.

Corollary 4.2.2. *The complete graph K_n is Gray connected.*

We remark here that when a graph is Gray connected it does not imply that this graph is induced by some Gray code. Let us take the complete graph K_3. It is easy to verify that K_3 is Gray connected, but there is no Gray code which induces K_3(the "connecting Gray codes" induce a subgraph of K_3, but not K_3 itself).

A Gray code $G(n)$ inducing the complete graph K_n is characterized by the consecutiveness of integers i and j, for all $1 \leq i, j \leq n$, at least once in the transition sequence $S(n)$. For instance, a cyclic Gray code of length 4, having transition sequence $\bar{S}(4) = 1, 2, 1, 3, 4, 3, 1, 2, 3, 2, 4, 2, 1, 4, 3, 4$, will induce the complete graph K_4, and a cyclic Gray code of length 5, having transition sequence

$$\bar{S}(5) := 1, 2, 3, 4, 5, 1, 5, 2, 3, 5, 2, 4, 2, 3, 4, 1, 4, 3, 2, 3, 1, 5, 3, 4, 1, 5, 2, 5, 3, 4, 1, 3,$$

induces K_5, as one can immediately verify by observing that any pair $\{i, j\}$ occurs as a pair of consecutive integers i and j at least once, for $1 \leq i \neq j \leq 4$.

As remarked in [89], the largest Gray code known to induce a complete graph is a Gray code of length 8 appearing in [87]. With respect to complete graphs induced by Gray codes, Wilmer and Ernst in [89] posed the following problem.

Problem 4.2.1. *Construct an n-bit Gray code which induces the complete graph K_n, for all $n \geq 1$.*

In this section we shall mainly deal with Problem 4.2.1. We introduce a technique for the construction of Gray codes inducing complete graphs. This technique is based on extended versions of the Gray construction due to Bakos [1], and also to Robinson and Cohn [60] who independently developed a related technique. We also refer to [72, 77], where we used a similar technique for the construction of totally balanced and exponentially balanced Gray codes. An extended construction was formulated earlier as Theorem 3.2.2, and another one will be formulated as Theorem 4.2.3 in Subsection 4.2.1 below. In Subsection 4.2.2 we discuss a technique how to make use of this extended construction to produce Gray codes inducing complete graphs. At the end of Subsection 4.2.2, we construct a Gray code of length 9 which induces the complete graph K_9, thus demonstrating our technique for the first unknown case $n = 9$. Finally, in Subsection 4.2.3 we suggest a related problem for further investigation.

4.2.1 Another extension of Bakos' Gray construction

The following extended Gray construction of Bakos' method, formulated as Theorem 4.2.3, holds for *odd* length of l. Our technique described in Subsection 4.2.2 for building Gray codes inducing complete graphs, is based on Theorem 3.2.2 and on Theorem 4.2.3. This technique is recursive, i.e. for constructing an n-bit Gray code inducing K_n it starts from an $(n-2)$-bit Gray code inducing K_{n-2}.

Theorem 4.2.3. *Let $\bar{S}(n-2) := u_0, s_{j_1}, u_1, s_{j_2}, ..., u_{l-1}, s_{j_l}, u_l, s_{2^n}$ be the transition sequence of an $(n-2)$-bit Gray code, where each u_k is a possibly empty sequence of transitions, and l is odd. Then the sequence*

$$\begin{aligned} u_0, s_{j_0}, u_1, ..., s_{j_l}, u_l, n-1, \\ u_l^R, n, u_l, n-1, u_l^R, s_{j_l}, \\ u_{l-1}^R, n-1, u_{l-1}, n, u_{l-1}^R, s_{j_{l-1}}, \\ \vdots \\ u_1^R, n, u_1, n-1, u_1^R, s_{j_1}, \\ u_0^R, n-1, u_0, n, u_0^R, n-1, \end{aligned}$$

is the transition sequence of an n-bit Gray code.

Proof. Again the proof is immediate from Lemma 3.2.1 of Chapter 3. □

Notice that Gray constructions of Theorem 3.2.2 and of Theorem 4.2.3 has the following important property with respect to our needs in this section. If transitions s_{j_i} and $s_{j_{i+1}}$ in the sequence $T := s_{j_0}, s_{j_1}, \ldots, s_{j_l}$ sandwich a single transition i in the transition sequence $\bar{S}(n-2)$, then in the transition sequence $\bar{S}(n)$ of the resulting Gray code the integers i and $n-1$ as well as the integers i and n are consecutive. For example let us consider a Gray code of length 3 with transition sequence $\bar{S}(3) = 1, \mathbf{2}, 1, \mathbf{3}, \mathbf{1}, 2, 1, 3$. Assume that we set the sequence $T := s_{j_1}, s_{j_2}, s_{j_3} = 2, 3, 1$ (bold faces in $\bar{S}(3)$). On this choice of the sequence T, an integer 1 is sandwiched by $s_{j_1} = 2$ and $s_{j_2} = 3$. Applying Theorem 4.2.3, the resulted Gray code will have the transition sequence $\bar{S}(5) = 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 5, 2, 1, 4, 1, 2, 1, 4, 5, 3, \mathit{1}, 5, \mathit{1}, 4, \mathit{1}, 2, 1, 4, 1, 5, 1, 4$. We see that the integer 1 (italic faces in $\bar{S}(5)$) is consecutive with the integers 4 and 5.

4.2.2 Constructing Gray codes which induce complete graphs

The following lemma says that if we have a complete transition sequence $\bar{S}(n)$ inducing the complete graph K_n, then we can find some k such that in $\bar{S}(n) := \bar{S}(n)^k$, the new transition sequence $S(n)$ still has the property that for every pair $\{i, j\}$, $1 \leq i \neq j \leq n$, there is at least one location in $S(n)$ where i and j are consecutive elements. We shall speak of a consecutive pair $\{i, j\}$.

Lemma 4.2.4. *Let the complete transition sequence $\bar{S}(n)$, $n \geq 5$, induce the complete graph K_n. There exists some k, $1 \leq k \leq 2^n$, such that the (non-complete) transition sequence $S(n)$ defined by $\bar{S}(n) := \bar{S}(n)^k$, contains at least one consecutive pair $\{i, j\}$, for all $i, j \in [n]$ with $i \neq j$.*

Proof. Assume we have k_i integers $i \in [n]$ in $\bar{S}(n)$. Since $\bar{S}(n)$ generates K_n, every integer $i \in [n]$ must be consecutive at least once to all integers in $[n] \backslash \{i\}$ in $\bar{S}(n)$. This implies that $k_i \geq \lceil \frac{n-1}{2} \rceil$ for every $i \in [n]$. We index each integer $i \in [n]$ in $\bar{S}(n)$ from 1 to k_i. Let a_{i_l} and b_{i_l} be a pair of transitions in $\bar{S}(n)$ squeezing the integer i_l, $1 \leq l \leq k_i$. First, let us consider the integer 1. Take $n_1 \leq n-1$ subsequences $a_{1_l}, 1_l, b_{1_l}$ of length 3 such that $\bigcup_{j \in I_1} \{a_{1_j}, b_{1_j}\} = [n] \backslash [1]$, where I_1 is some subset of $[k_1]$ with cardinality n_1. Let S_1 be the concatenated sequence of these n_1 subsequences $a_{1_l}, 1_l, b_{1_l}$ where the order of its elements is the same as in $\bar{S}(n)$, i.e. x precedes y in

S_1 if and only if x precedes y in $\bar{S}(n)$. Notice that it is possible that b_{1_j} and $a_{1_{j+1}}$ are the same transitions in $\bar{S}(n)$, for some $j \in I_1$. So, the length of the sequence S_1 is less than or equal to $3n_1 \leq 3(n-1)$. Moreover, the sequence S_1 has the property that the integer 1 is consecutive to every integer in $[n]\backslash[1]$.
Next, consider the integer 2. Notice that the integer 2 is already known to be consecutive to the integer 1 in S_1. Take $n_2 \leq n-2$ subsequences $a_{2_l}, 2_l, b_{2_l}$ such that $\bigcup_{j\in I_2}\{a_{2_j}, b_{2_j}\} = [n]\backslash[2]$, where I_2 is some subset of $[k_2]$ with cardinality n_2. Let S_2 be the concatenated sequence of these n_2 subsequences $a_{2_l}, 2_l, b_{2_l}$ where its elements are again ordered as they are in $\bar{S}(n)$. Notice again that it is possible that the integers b_{2_j} and $a_{2_{j+1}}$ are the same transitions in $\bar{S}(n)$, for some $j \in I_2$. So, the length of the sequence S_2 is less than or equal to $3n_2 \leq 3(n-2)$. Let the sequence $S_{1,2}$ be the sequence which consisting of the elements of S_1 and S_2, where the order of these elements is the same as in $\bar{S}(n)$. By the construction of S_1 and S_2, it is clear that the integers 1 and 2 are consecutive to all other integers in $[n]$ in the sequence $S_{1,2}$. Since it is also possible that a_{i_r} and a_{j_s} or b_{i_x} and b_{j_y} are the same transitions in $\bar{S}(n)$ for some $r, x \in k_i$ and some $s, y \in k_j$, the length of the sequence $S_{1,2}$ is less than or equal to the sum of the lengths of S_1 and S_2. If we continue this process until the integer $n-1$, then we obtain a sequence $S_{1,2,\ldots,n-1}$ consisting of $S_1, S_2, \ldots, S_{n-1}$ with the property that the integers $1, 2, \ldots, n-2$ and $n-1$ are consecutive to all other integers from $[n]$. We emphasize that the elements in $S_{1,2,\ldots,n-1}$ are ordered in the same way as in $\bar{S}(n)$. The length of the sequence $S_{1,2,\ldots,n-1}$ is less than or equal to

$$\begin{aligned} 3n_1 + 3n_2 + \cdots + 3.2 + 3 &\leq 3(n-1) + 3(n-2) + \cdots + 3.2 + 3 \\ &= \frac{3}{2}n(n-1) \\ &< 2^n - 1 \text{ (=the length of } S(n)), \end{aligned}$$

for all $n \geq 5$. Hence, there exists at least one transition, say s_k, in $\bar{S}(n)$ which is not in $S_{1,2,\ldots,n-1}$. For our convenience we rename $\bar{S}(n)^k$ by $\bar{S}(n)$. Now the closing transition of $\bar{S}(n)$ is s_k, and hence the non-complete transition sequence $S(n)$ defined by this $\bar{S}(n)$ will contain $S_{1,2,\ldots,n-1}$, and therefore $S(n)$ has the property that for all $i \neq j \in [n]$, the pair $\{i, j\}$ occurs at least once in $S(n)$ as a pair of consecutive transitions. □

Now we are ready to introduce Construction 4.2. Since there is no Gray code $G(3)$ which induces the complete graph K_3, our technique can only be applied to codeword length $n \geq 4$.
As is stated right after Theorem 3.2.2, we use the notation T for the sequence consisting of all l transitions $s_{i_1}, s_{i_2}, \ldots, s_{i_l}$ in Theorem 4.2.3.

Let $G(n-2)$, $n \geq 6$, be a Gray code inducing K_{n-2}, which has transition sequence $\bar{S}(n-2)$.

Construction 4.2

1. Shift in cyclic sense, if necessary, the transitions sequence $\bar{S}(n-2)$ such that $S(n-2)$ satisfies the condition in Lemma 4.2.4.

2. Mark $n-2$ transitions $1, 2, ..., n-2$ in $S(n-2)\backslash\{s_1, s_{2^n-1}\}$.

3. Define T to be the sequence consisting of all transitions in $S(n-2)$ squeezing the marked transitions of Step 2. Put an additional element s_1 at the front of T, if the first transition is unequal to s_1.

4. Apply Theorem 3.2.2 or Theorem 4.2.3 using this sequence T.

Example 4.2.1. Consider again a Gray code of length 4 inducing the complete graph K_4 with transition sequence $\bar{S}(4) := 1, 2, 1, 3, 4, 3, 1, 2, 3, 2, 4, 2, 1, 4, 3, 4$. It is easy to see that the corresponding transition sequence S_4 satisfies already the condition described in Lemma 4.2.4. We now mark the transitions $s_2 = 2, s_4 = 3, s_7 = 1, s_{11} = 4$ in $S_4\backslash\{s_1, s_{2^4-1}\}$. The sequence T will consist of the transitions $s_1, s_3, s_5, s_6, s_8, s_{10}$, and s_{12} in S_4. Then, by applying Theorem 4.2.3, since $|T|$ is odd, with this sequence T, we obtain a 6-bit Gray code with transition sequence

$$\begin{aligned}\bar{S}(6) \;:=\; & 1,2,1,3,4,3,1,2,3,2,4,2,1,4,3,5,3,4,1,6,1,4,3,5,3,4,1,2,4,5,4,6,\\ & 4,2,3,6,3,5,3,2,1,5,1,6,1,3,6,5,4,3,5,3,6,3,1,2,6,2,5,2,1,5,6,5.\end{aligned}$$

One can easily verify that the transition sequence $\bar{S}(6)$ induces the complete graph K_6, by observing that each pair $\{i, j\}$ occurs at least once as a consecutive pair, for all $i \neq j$, $1 \leq i, j \leq 6$.

Next we consider a Gray code $G(5)$ with transition sequence.

$$\bar{S}(5) := 1,2,3,4,5,1,5,2,3,5,2,4,2,3,4,1,4,3,2,3,1,5,3,4,1,5,2,5,3,4,1,3.$$

One can verify that this transition sequence induces the complete graph K_5, and moreover that the transition sequence S_5 has the property described in Lemma 4.2.4. Now we mark 5 non-consecutive transitions $s_2 = 2, s_4 = 4, s_6 = 1, s_9 = 3$, and $s_{22} = 5$ in $\bar{S}(5)$. The sequence T will consist of transitions $s_1, s_3, s_5, s_7, s_8, s_{10}, s_{21}$, and s_{23} in $S(5)$. Using this sequence T in Theorem 3.2.2, Step 4 of Construction 4.2 now produces the following transition sequence.

$$\begin{aligned}\bar{S}(7) \;:=\; & 1,2,3,4,5,1,5,2,3,5,2,4,2,3,4,1,4,3,2,3,1,5,3,4,1,5,2,5,3,4,1,6,\\ & 1,4,3,5,2,5,1,4,7,4,1,5,2,5,3,4,1,6,1,4,3,5,2,5,1,4,3,5,6,5,7,5,\\ & 1,3,2,3,4,1,4,3,2,4,2,7,2,4,2,3,4,1,4,3,2,3,6,3,2,3,4,1,4,3,2,4,\\ & 2,5,3,6,3,7,3,2,7,6,5,1,6,1,7,1,5,4,7,4,6,4,3,2,6,2,7,2,1,7,6,7.\end{aligned}$$

Again, one can easily verify that $\bar{S}(7)$ induces the complete graph K_7.

Theorem 4.2.5. *A Gray code $G(n)$ resulting from Construction 4.1 induces the complete graph K_n.*

Proof. First we remark that Steps 1 and 2 of Construction 4.2 are valid, because of Lemma 4.2.4. The transition sequence $\bar{S}(n-2)$ induces, by assumption, the complete

graph K_{n-2}, i.e. every pair $\{i,j\}$ of bit positions i and j, $1 \leq i,j \leq n-2$, occurs at least once as a consecutive pair in $\bar{S}(n-2)$. Step 1 of Construction 4.2 says that this last property is preserved in the transition sequence $S(n-2)$. This implies that the property is also preserved in the transition sequence $\bar{S}(n)$(See Theorem 3.2.2 and Theorem 4.2.3), and hence, for every $1 \leq i,j \leq n-2$, the pair $\{i,j\}$ occurs at least once as consecutive pair in $\bar{S}(n)$.
Since, for every i in $[n-2]$, we can select a_i and b_i in T such that a_i, i, b_i is consecutive in $S(n-2)$, by Theorem 3.2.2 or Theorem 4.2.3, we have in the new transition sequence $\bar{S}(n)$, a subsequence $a_i, i, n-1, i, n, i, b_i$ or $a_i, i, n, i, n-1, i, b_i$ indicating that for every $i \in [n-2]$ there are consecutive pairs $\{i, n-1\}$ and $\{i, n\}$ in $\bar{S}(n)$. The choice of transition s_1 at the front of T, implies $u_0 = \emptyset$, hence there is also a consecutive pair of integers $n-1$ and n in $S(n)$(See again Theorem 3.2.2 and Theorem 4.2.3). So, $\bar{S}(n)$ has the property that for every $i \neq j \in [n]$, the pair $\{i,j\}$ occurs at least once as a consecutive pair in $\bar{S}(n)$, and therefore induces the complete graph K_n. □

As mentioned before, there is no Gray code of length 3 which induces the complete graph K_3. However, the Gray codes of length 1 and 2 do induce the complete graphs K_1 and K_2, respectively. Since Construction 4.2 works for $n \geq 6$, and since it is known from Example 4.2.1 that there are Gray codes of length 4 and 5 which induce the complete graphs K_4 and K_5, respectively, we have proved now the following theorem.

Theorem 4.2.6. *For every $n \geq 1, n \neq 3$, there exists a Gray code of length n which induces the complete graph K_n.*

From Example 4.2.1 we have a 7-bit Gray code inducing the complete graph K_7 with transition sequence

$$\begin{aligned}\bar{S}(7) \;:=\; & 1,2,3,4,5,1,5,2,3,5,2,4,2,3,4,1,4,3,2,3,1,5,3,4,1,5,2,5,3,4,1,6,\\ & 1,4,3,5,2,5,1,4,7,4,1,5,2,5,3,4,1,6,1,4,3,5,2,5,1,4,3,5,6,5,7,5,\\ & 1,3,2,3,4,1,4,3,2,4,2,7,2,4,2,3,4,1,4,3,2,3,6,3,2,3,4,1,4,3,2,4,\\ & 2,5,3,6,3,7,3,2,7,6,5,1,6,1,7,1,5,4,7,4,6,4,3,2,6,2,7,2,1,7,6,7.\end{aligned}$$

To apply Construction 4.2, we introduce a sequence T consisting of transitions

$$s_1, s_3, s_5, s_7, s_8, s_{10}, s_{21}, s_{23}, s_{31}, s_{33}, s_{40}, s_{42}.$$

The resulting transition sequence is

$$\begin{aligned}\bar{S}(9) \;:=\; & 1,2,3,4,5,1,5,2,3,5,2,4,2,3,4,1,4,3,2,3,1,5,3,4,1,5,2,5,3,4,1,6,\\ & 1,4,3,5,2,5,1,4,7,4,1,5,2,5,3,4,1,6,1,4,3,5,2,5,1,4,3,5,6,5,7,5,\\ & 1,3,2,3,4,1,4,3,2,4,2,7,2,4,2,3,4,1,4,3,2,3,6,3,2,3,4,1,4,3,2,4,\\ & 2,5,3,6,3,7,3,2,7,6,5,1,6,1,7,1,5,4,7,4,6,4,3,2,6,2,7,2,1,7,6,8,\\ & 6,7,1,2,7,2,6,2,3,4,6,4,7,4,5,1,7,1,6,1,5,6,7,2,3,7,3,6,3,5,2,4,\\ & 2,3,4,1,4,3,2,3,6,3,2,3,4,1,4,3,2,4,2,7,2,4,2,3,4,1,4,3,2,3,1,5,\\ & 7,5,6,5,3,4,1,5,2,5,3,4,1,6,1,4,3,5,2,5,1,9,1,5,2,5,3,4,1,6,1,4,\\ & 3,5,2,5,1,4,3,5,6,5,7,5,1,3,2,3,4,1,4,3,2,4,2,7,2,4,2,3,4,1,4,3,\\ & 2,3,6,3,2,3,4,1,4,3,2,4,2,5,3,6,3,7,3,2,7,6,5,1,6,1,7,1,5,4,7,4,\\ & 6,4,3,2,6,2,7,2,1,7,6,8,6,7,1,2,7,2,6,2,3,4,6,4,7,4,5,1,7,1,6,1,\\ & 5,6,7,2,3,7,3,6,3,5,2,4,2,3,4,1,4,3,2,3,6,3,2,3,4,1,4,3,2,4,2,7,\\ & 2,4,2,3,4,1,4,3,2,3,1,5,7,5,6,5,3,4,1,5,2,5,3,4,1,6,1,4,3,5,2,5,\\ & 1,4,7,8,7,9,7,4,1,5,2,5,3,4,9,4,3,5,2,5,1,8,1,5,2,5,3,4,1,6,8,6,\\ & 9,6,1,4,3,5,2,5,1,4,9,4,1,5,2,5,3,4,8,4,3,5,2,5,1,4,3,5,8,5,9,5,\\ & 1,3,2,3,4,1,4,3,2,4,2,9,2,4,2,3,4,1,4,3,2,3,8,3,2,3,4,1,4,3,2,4,\\ & 2,5,3,8,3,9,3,2,9,8,5,1,8,1,9,1,5,4,9,4,8,4,3,2,8,2,9,2,1,9,8,9.\end{aligned}$$

Because of Theorem 4.2.5 we may conclude that this transition sequence $\bar{S}(9)$ induces the complete graph K_9, as can also easily be verified.

4.2.3 Conclusions

In the previous section we introduced a recursive technique how to construct a Gray code inducing a complete graph. Furthermore, we showed that for every $n \neq 3$ an n-bit Gray code exists which induces the complete graph K_n, and hence we solved Problem 4.2.1 of Wilmer and Ernst in [89].
As for *digraphs*, Wilmer and Ernst in [89] also proved the existence of a Gray code of any length $n \geq 6$, the graph of which contains no bi-directional edges. This gives rise to the following problem.

Problem 4.1 Does there exist an n-bit Gray code, for every $n \geq 6$, which induces the complete graph K_n, and which has no bi-directional edges?

We remark that the extended Gray construction described in Subsection 4.2.1 can not be used to construct Gray codes specified in Problem 4.1, since the resulting sequence will contain subsequences of type $u, n-1, u^R, n, u$ and of type $u, n, u^R, n-1, u$ which take care that the induced graphs contain bi-directional edges $\{i, n-1\}$ and $\{i, n\}$ and also $\{j, n-1\}$ and $\{j, n\}$, where i and j are the first and last transition in u, respectively.

5

Balanced Maximum Counting Sequences and Uniform Counting Sequences

We discuss some specific counting sequences which are called maximum counting sequences and uniform counting sequences. We introduce a number of constructions for these types of counting sequences. Some of the constructions have the advantage of producing maximum counting sequences and uniform counting sequences which are balanced as well. These constructions enable us to settle a few conjectures of Robinson and Cohn [60] in positive sense.

5.1 Maximum counting sequences

Sometimes one needs a counting sequence such that the number of bit changes from a codeword to its successor is as *large* as possible, for example when testing a physical circuit for reliable behavior in worst-case conditions (see e.g. [32, Exercise 67, p. 35] or [60]). As is discussed in Chapter 1, a sequence which satisfies this criterium is called a maximum counting sequence, i.e. a counting sequence $\mathcal{O}(n)$ with average Hamming distance $d_A = n - \frac{1}{2}$(cf. Theorem 1.2.1). The counting sequence listed in Figure 5.1, is an example of a maximum counting sequence of length 4.
We can observe that the Hamming distance between two successive codewords is alternating between three and four. It implies that the average Hamming distance of this sequence is equal to $(3.8 + 4.8)/16 = 4 - \frac{1}{2} = 3\frac{1}{2}$ which proves the sequence to be a maximum counting sequence. In general, a maximum counting sequence of length $n > 1$ has the property that Hamming distance between two successive codewords is alternating between n and $n - 1$. Thus, it is easy to understand that any two

0000	0110
1111	1001
0001	0111
1110	1000
0011	0101
1100	1010
0010	0100
1101	1011

Figure 5.1: A maximum counting sequence of length 4.

codewords sandwiching another codeword will have Hamming distance exactly one. Hence we have the following theorem.

Theorem 5.1.1 (Robinson-Cohn [60]). *A maximum counting sequence consists of two interlaced cyclic Gray sequences such that every other pair of successive codewords are each other's complement.*

One immediate corollary of Theorem 5.1.1 is the following.

Corollary 5.1.2. *The two interlaced cyclic Gray sequences in a maximum counting sequence have the same transition sequence.*

Proof. Let $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{2^n-2}, \mathbf{x}_{2^n-1}$ be a maximum counting sequence. Then, according to Theorem 5.1.1, the sequences $\mathbf{x}_0, \mathbf{x}_2, \mathbf{x}_4, \ldots, \mathbf{x}_{2^n-2}$ and $\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5, \ldots, \mathbf{x}_{2^n-1}$ form cyclic Gray sequences. Without loss of generality, let $\mathbf{x}_i \oplus \mathbf{x}_{i+1} = \mathbf{1}$ for even values of i (hence $|\mathbf{x}_{i+1} \oplus \mathbf{x}_{i+2}| = n-1$ and $\mathbf{x}_{i+2} \oplus \mathbf{x}_{i+3} = \mathbf{1}$). Consider any four consecutive codewords $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}, \mathbf{x}_{i+3}$, in the maximum counting sequence, where indices are modulo 2^n and i is even. Assume that $\mathbf{x}_i$ and $\mathbf{x}_{i+2}$ differ in bit position j. So, $x_{i+1\,j} = x_{i+2\,j}$. Since, $\mathbf{x}_{i+2} \oplus \mathbf{x}_{i+3} = \mathbf{1}$, $x_{i+3\,j} = x_{i+2\,j} \oplus 1 \neq x_{i+1\,j}$. □

As was defined in Chapter 1, a counting sequence of codeword length n is called *balanced*, if it satisfies $|TC_{\mathcal{O}(n)}(i) - TC_{\mathcal{O}(n)}(j)| \leq 2$, and it is said to be *totally balanced* if $|TC_{\mathcal{O}(n)}(i) - TC_{\mathcal{O}(n)}(j)| = 0$, for every $1 \leq i, j \leq n$ (cf. [60]). The following sequence, shown in Figure 5.2, is an example of a balanced maximum counting sequence of length 5 with transition count spectrum (28, 30, 30, 28, 28).

Robinson and Cohn in [60] introduced a simple construction for producing maximum counting sequences. For constructing a maximum counting sequence of length n, they start with a cyclic Gray code of length $n-1$, and then add prefix zero to each codeword. The construction is continued by inserting the complement of each extended codeword right after the codeword itself. The resulting sequence is a maximum counting sequence of length n. However, it will be clear that the resulting maximum counting sequence is not balanced, except for $n = 2$ and 3. This is because the transition count of the n-th bit is always equal to 2^n, and therefore, there is at least one bit position which has transition count at most $2^n - 4$ due to the defining equation of d_A

$M5$

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

Figure 5.2: A 5-bit balanced maximum counting sequence.

$$\sum_{i}^{n} TC_{\mathcal{O}(n)}(i) = 2^n d_A,$$

which is satisfied by any counting sequence.

Furthermore, Robinson and Cohn claim that all maximum counting sequences of length 2, 3, and 4 have a form which corresponds to their construction. It implies that there is some bit position in any sequence of length n the transition count of which is equal to 2^n, for $2 \leq n \leq 4$. Due to the above equality, we may conclude that there is no balanced maximum counting sequence of length four. This can also be verified by inspection.

Conjecture 5.1.3 (Robinson-Cohn [60]). *For every $n > 1$, $n \neq 4$, a balanced maximum counting sequence exists.*

In the remaining part of this section we introduce two constructions for obtaining maximum counting sequences. One of these is a recursive method, while the other is inspired by Theorem 5.1.1, and will be explicitly formulated as Theorem 5.1.5 below. This last construction has the advantage that the constructed maximum counting sequences are balanced as well.

We start with our first construction. To this end, we introduce some special notation. If L is an ordered list, $[ab]L$ stands for a list obtained by putting prefix a to each codeword of L having an even index and prefix b otherwise.

For example, let $L := 00, 01, 11, 10$. Then we obtain

$$\begin{array}{lcl} [01]L & := & 000, 101, 011, 110; \\ [10]L & := & 100, 001, 111, 010; \\ [01]L^R & := & 010, 111, 001, 100. \end{array}$$

Furthermore, the list obtained from the list L by complementing each codeword of L will be denoted by L^C. In the above example we have

$$\begin{array}{rl} L^C & := 11, 10, 00, 01, \qquad \text{and} \\ ([10]L)^{CR} & := (([10]L)^C)^R = 101, 000, 110, 011. \end{array}$$

Construction 5.1

Let M_{n-1} be a maximum counting sequence of length $n-1$. Then the sequence

$$M_n := [01]M_{n-1}, [10]M_{n-1}^{RC},$$

is a maximum counting sequence of length n.

Example 5.1.1. Applying Construction 5.1 to the maximum sequence M_3 provides us with a maximum sequence M_4.

M_3	M_4	
000	0000	1010
111	1111	0101
001	0001	1011
110	1110	0100
011	0011	1001
100	1100	0110
010	0010	1000
101	1101	0111

It is easy to understand, that Construction 5.1 in general really produces a maximum counting sequence M_n if the sequence M_{n-1} is a maximum counting sequence. Let $c_n(i)$, $1 \leq i \leq n$, be the transition count of the integer i in a counting sequence n. An obvious property of M_n which is constructed using Construction 5.1, is formulated as the following theorem.

Theorem 5.1.4. *Let M_{n-1} be a maximum counting sequence with transition count spectrum*

$$(c_{n-1}(1), c_{n-1}(2) \ldots, c_{n-1}(i), \ldots, c_{n-1}(n-1)),$$

and let i be the unique position where the last and the first codeword in M_{n-1} have equal components. Then the resulting maximum counting sequence M_n has transition count spectrum

$$(2c_n(1), 2c_n(2), \ldots, 2c_{n-1}(i)+2, \ldots, 2c_n(n-1), c_n(n) = 2^n - 2).$$

From Example 5.1.1, we know that the transition count spectrum of M_3 is (8, 6, 6). The zero bit of the last codeword of M_3 is in position 2, and therefore we have that the transition count spectrum of M_4 is (14, 16 14, 12). Indeed, this sequence has the maximal averaged Hamming distance $3\frac{1}{2}$.

A Gray sequence of length n with period 2^{n-1} such that no two codewords are complements of each other will be called a *half Gray sequence* of length n. Because of Corollary 5.1.2, the following statement seems to be obvious. If we have a cyclic balanced Gray sequence of length n with period 2^{n-1} such that no two codewords are complements of each other, then by inserting complement right after each codeword of the given Gray sequence, we obtain a balanced maximum sequence of length n. In the next, the transition sequence of a cyclic Gray sequence of length n with period 2^{n-1} satisfying the requirement that no two codewords are complementary to each other, will be denoted by $S_h(n)$. The sequence M_5 of Figure 5.2 yields an example of this notion. The boldfaced codewords in M_5 constitute a cyclic balanced Gray sequence with transition sequence $S_h(5) = 1, 2, 1, 3, 5, 2, 5, 4, 1, 2, 1, 3, 5, 2, 5, 4$. In the next, a cyclic Gray sequence generated by a transition sequence $S_h(n)$ is occasionally called a *cyclic half Gray sequence* or a *half Gray cycle* of length n.

Theorem 5.1.5. *Let $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{2^{n-1}-1}$, be an n-bit balanced half Gray cycle of length n such that $\mathbf{a}_i \oplus \mathbf{a}_j \neq \mathbf{1}$ for every $0 \leq i, j < 2^{n-1}$. Then the sequence $M := \mathbf{a}_0, \mathbf{b}_0, \mathbf{a}_1, \mathbf{b}_1, \ldots, \mathbf{a}_{2^{n-1}-1}, \mathbf{b}_{2^{n-1}-1}$ with $\mathbf{a}_i \oplus \mathbf{b}_i = \mathbf{1}$, $0 \leq i < 2^{n-1}$, is a balanced maximum counting sequence of length n. Furthermore, if the list $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{2^{n-1}-1}$, has transition count spectrum $(c(1), c(2), ..., c(n))$, then the transition count spectrum of M is $(2^n - c(1), 2^n - c(2), ..., 2^n - c(n))$.*

Proof. It will be clear that the period of M is equal to 2^n. It is also obvious that M is a maximum counting sequence by construction. Since $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{2^{n-1}-1}$ is assumed to be balanced, $|c(i) - c(j)| \leq 2$ for all $1 \leq i, j \leq n$. Let $S_h(n)$ be the transition sequence of the n-bit Gray cycle. Consider codewords $\mathbf{a}_{i-1}$ and $\mathbf{a}_i$ which differ in position s_i. Then it is easy to see that $\mathbf{b}_{i-1}$ and $\mathbf{a}_i$ are equal in position s_i. Since the occurrence of each s_i in $S_h(n)$ decreases the transition count of M in the column s_i by one, the total transition count of bit position s_i will be equal to $tc_n(s_i) := 2^n - c(s_i)$. The assumption that $|c(i) - c(j)| \leq 2$, $1 \leq i, j \leq n$, implies that $|tc_n(i) - tc_n(j)| \leq 2$. □

The problem now is to determine half Gray cycles $G_h(n)$ of length n and of period 2^{n-1} as mentioned in Theorem 5.1.5 or to obtain sequences $S_h(n)$ for certain values of n. The following subsection will solve this problem.

5.1.1 A construction of Gray sequences

For the sake of clearness, a Gray sequence of length n and of period p will be denoted by $G(n|p)$. Let $S_{G(n|p)}$ be the complete transition sequence of a *Gray* sequence $G(n|p)$. Let u be a subsequence of $S_{\mathcal{O}(n|p)}$ which may be empty and let u^R be the sequence obtained from u by reversing its order.

The following theorem formulates a construction of Gray sequence from a Gray sequence of smaller length. This Gray sequence construction is a generalization of the Gray code construction formulated in Theorem 3.2.2. We also emphasize here that the construction can be adjusted easily such that it holds for l is odd.

Theorem 5.1.6. *Let $S_{G(n-2|p)} := u_0, s_{j_1}, u_1, s_{j_2}, u_2, , \ldots, s_{j_l}, u_l, s_p$ be the complete transition sequence of an $(n-2)$-bit cyclic Gray sequence, where each u_i is a possibly empty sequence of transitions, and where l is even. Then the sequence*

$$\begin{aligned} S := \; & u_0, s_{j_1}, u_1, s_{j_2}, u_2, \ldots, s_{j_l}, u_l, n-1, \\ & u_l^R, n, u_l, n-1, u_l^R, s_{j_l}, \\ & u_{l-1}^R, n-1, u_{l-1}, n, u_{l-1}^R, s_{j_{l-1}}, \\ & \vdots \\ & u_1^R, n-1, u_1, n, u_1^R, s_{j_1}, \\ & u_0^R, n, u_0, n-1, u_0^R, n, \end{aligned}$$

is the complete transition sequence of an n-bit cyclic Gray sequence of period $4p$.

Notice that if $p = 2^{n-2}$, the Theorem will be the same as Theorem 3.2.2. Therefore, we may infer that a proof of the Theorem can be completed using Lemma 3.2.1.

Example 5.1.2. $S_{G(3|6)} = 1, \underline{2}, \underline{3}, 1, 2, 3$ is the complete transition sequence of a 3-bit cyclic Gray sequence $000, 001, 011, 111, 110, 100$. Take $u_0 = 1$, $s_{j_1} = 2$, $u_1 = \emptyset$, $s_{j_2} = 3$, and $u_2 = 1, 2$. Then the resulting sequence S is equal to $1, 2,\ 3, 1,\ 2, 4,\ \ 2, 1,\ 5, 1,\ 2, 4,\ 2, 1,\ 3,\ \ 4,\ 5, 2,\ \ 1, 5,\ 1, 4,\ 1, 5$.

As we did w.r.t. Theorem 3.2.2, we shall denote the sequence $s_{j_1}, s_{j_2}, \ldots, s_{j_l}$, in Theorem 5.1.6 by T. Thus the length of sequence T is equal to l. Notice again that sequence T does not include the closing transition s_p.

If we assume that $G(n-2|p)$ and $G(n|4p)$ in Theorem 5.1.6 start from the zero codeword, then it is obvious that the list of the first p codewords of $G(n|4p)$ is equal to $00G(n-2|p)$, where $00G(n-2|p)$ stands for the sequence $G(n-2|p)$ the codewords of which have prefix 00. Now, let $G(n|4p)$ be equal to the concatenated sequence $00G(n-2|p), G(n|3p)$. The following corollary is obvious.

Corollary 5.1.7. *For every $\mathbf{x} \in G(n|3p)$ one has that $\mathbf{x} = ab\mathbf{y}$, for some $\mathbf{y} \in G(n-2|p)$, with $ab \in \{01, 11, 10\}$.*

A Gray sequence $G(n|2^{n-1})$ of length n and of period 2^{n-1} satisfying the property that no two codewords in $G(n|2^{n-1})$ are complements of each other will be called a *half Gray sequence*. Let $G(n)$ be a cyclic Gray code of length n. Then the sequence

$0G(n)$ is a cyclic half Gray sequence of length $n+1$. Hence, we can conclude that a cyclic half Gray sequence of length n exists for every $n \geq 1$. Theorem 5.1.8 shows that a cyclic half Gray sequence of length n can also be constructed from an existing half Gray sequence of length $n-2$ using Theorem 5.1.6.

For the simplicity, a cyclic half Gray sequence of length n will be denoted by $G_h(n)$ and the transition sequence which generates $G_h(n)$ will be denoted by $S_h(n)$. Furthermore, $\mathbf{1}_n$ will stand for the n-bit codeword all components of which are 1.

Theorem 5.1.8. *Let $G_h(n-2)$ be a cyclic half Gray sequence with transition sequence $S_h(n-2)$. Then the list L of codeword length n generated when using Theorem 5.1.6 with $S_h(n-2)$ as basis of the construction, is a cyclic half Gray sequence.*

Proof. The sequence L is a Gray sequence as is guaranteed by Theorem 5.1.6. We just need to show that every pair of codewords in L are not complements of each other. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be two different codewords in L. According to Corollary 5.1.7, we have that $\mathbf{x}_1 = ab\mathbf{y}_1$ and $\mathbf{x}_2 = cd\mathbf{y}_2$, for some $\mathbf{y}_1$,$\mathbf{y}_2 \in G_h(n-2)$, with $ab, cd \in \{00, 01, 11, 10\}$. Since $\mathbf{x}_1 \oplus \mathbf{x}_2 \neq \mathbf{1}_{n-2}$, we conclude that $\mathbf{y}_1 \oplus \mathbf{y}_2 \neq \mathbf{1}_n$. □

We close this section by showing some examples where Theorem 5.1.6 is applied to construct cyclic half Gray sequences of length n from cyclic Gray sequences of length $n-2$.

Example 5.1.3. Let us consider $S_h(5) = 1, 2, 1, 3, 5, 2, 5, 4, 1, 2, 1, 3, 5, 2, 5, 4$ which corresponds to the list shown in Figure 2.a. The transition count spectrum of the cyclic half Gray sequence generated by $S_h(5)$ is $(4, 4, 2, 2, 4)$. Shift $S_h(5)$ cyclicly to the form

$$4, \underline{1}, \underline{2}, \underline{1}, 3, \underline{5}, \underline{2}, \underline{5}, 4, \underline{1}, \underline{2}, 1, 3, 5, 2, 5.$$

Choose the sequence T consisting of the underlined transitions. Apply Theorem 5.1.6 to obtain the following complete transition sequence of a cyclic half Gray sequence of length 7 with period 2^6

$$\underline{4}, \underline{1}, \underline{2}, \underline{1}, \underline{3}, \underline{5}, \underline{2}, \underline{5}, \underline{4}, \underline{1}, \underline{2}, \underline{1}, \underline{3}, 5, \underline{2}, \underline{6}, \underline{2}, 5, 3, \underline{1}, \underline{7}, \underline{1}, 3, 5, \underline{2}, \underline{6}, 2, 5, 3, 1, 2, \underline{6},$$
$$\underline{7}, 1, 4, \underline{7}, 4, \underline{6}, 4, 5, \underline{6}, \underline{7}, 2, 7, \underline{6}, 5, 3, 6, 3, 7, 3, 1, 7, 6, 2, 6, 7, 1, 4, 7, 4, 6, 4, 7.$$

One can verify by inspection that this sequence is really an $S_h(7)$. Also by inspection, we find that the cyclic half Gray sequence generated by $S_h(7)$ has transition count spectrum $(10, 10, 8, 8, 8, 10, 10)$.
If we want to construct an $S_h(9)$ from this $S_h(7)$ we can do so by defining a sequence T of length 28 as basis of the construction of Theorem 5.1.6 which consists of six integers $1, 2$ and 6 each, two integers $3, 4$, and 5 each, and four integers 7. For instance, we can choose the underlined integers in $S_h(7)$ for the sequence T. The resulting $S_h(9)$ will generate a cyclic half Gray sequence of length 9 which has transition count spectrum $(28, 28, 28, 28, 28, 28, 28, 30, 30)$.

Example 5.1.4. The cyclic half Gray sequence which is shown in Figure 2.b. has transition sequence

$$S_h(6) = 1,6,2,3,5,3,4,6,5,4,1,3,1,4,1,3,5,6,4,3,5,4,2,1,2,3,2,1,2,4,2,6,$$

and transition count spectrum $(6,6,6,6,4,4)$.
Shift the sequence $S_h(6)$ cyclicly, such that an integer having transition count 6 is the closing transition. For example, we write $S_h(6)$ as the sequence

$$6,2,3,5,3,4,6,5,4,1,3,1,4,1,3,5,6,4,3,5,4,2,1,2,3,2,1,2,4,2,6,1,$$

with 1 as its closing transition. Select two integers 1, and four integers 2, 3, and 4 each, to be elements of the sequence T. Hence, the length of T is 14. By applying Theorem 5.1.6, we have that the resulting sequence is an $S_h(8)$ with transition count spectrum (16,16,16,16,16,16,16,16).

5.1.2 Constructing balanced cyclic half Gray sequences

In this section we shall show that a *balanced* cyclic half Gray sequence of length n exists for every $n > 1, n \neq 4$.
It is clear that the sequences 0, $0G(1)$, and $0G(2)$ are examples of balanced cyclic half Gray sequences of codeword length $1, 2$ and 3, respectively. Later on we shall show that balanced cyclic half Gray sequences of length 4 do not exist. Furthermore, Figures 5.3.a. and 5.3.b. show that balanced cyclic half Gray sequences of length 5 and of length 6 exist.

First let us turn to Theorem 5.1.6. Further observation shows that in the transition sequence S constructed in Theorem 5.1.6 we have

$$TC_{G(n|4p)}(i) := \begin{cases} l+2, & \text{if } i = n-1, n, \\ 4TC_{G(n-2|p)}(i) - 2b(i), & \text{if } 1 \leq i \leq n-2 \text{ and } i \neq s_p, \\ 4(TC_{G(n-2|p)}(s_p) - 1) - 2b(s_p), & \text{if } i = s_p, \end{cases} \quad (5.1)$$

where $b(i)$ is the cardinality of the set $\{k|s_{j_k} = i\}$. Notice that for each i, $b(i)$ is equal to the number of times that the integer i occurs in the sequence T. This implies that the sum of $b(i)$ over all i, $1 \leq i \leq n-2$, is equal to l, the cardinality of T.

If we know the number of integers i, $1 \leq i \leq n$, which must occur in a complete transition sequence $S_h(n)$ of a cyclic half Gray sequence $G_h(n)$ such that $G_h(n)$ is balanced, then because of (5.1), the choice of the integers $b(i)$ can be planned. This can rather easily be accomplished by observing *n-partitions* of the integer 2^{n-1}, $n > 1$, i.e. partitions $(p_1, p_2, \ldots, p_n)$, with $p_i > 0$ for $1 \leq i \leq n$, and $p_1 \leq p_2 \leq \ldots \leq p_n$, such that $\sum_{j=1}^{n} p_j = 2^{n-1}$. The partition $(p_1, p_2, \ldots, p_n)$ is called a *balanced even* n-partition of the integer 2^{n-1}, if p_i is even for every i, $1 \leq i \leq n$, and $|p_i - p_j| \leq 2$,

00000	000000	001010
00001	000001	011010
00011	100001	111010
00010	100011	110010
00110	100111	110110
10110	110111	100110
10100	110011	101110
00100	111011	101100
01100	011011	101101
01101	001011	101111
01111	000011	101011
01110	000010	101001
01010	000110	101000
11010	000111	101010
11000	001111	100010
01000	001110	100000
a.	*b.*	

Figure 5.3: a. A 5-bit *balanced* cyclic half Gray sequence, b. a 6-bit *balanced* cyclic half Gray sequence.

for all $1 \leq i, j \leq n$. It is clear that balanced even n-partitions of the integer 2^{n-1} do not exist for $n = 1, 2, 3$. However, for every $n > 3$ there exists a unique balanced even n-partition of 2^{n-1}. This is a consequence of the following lemma dealing with *balanced partitions*, i.e. the various parts of the partitions differ at most 1. The proof is straightforward.

Lemma 5.1.9. *Let $m, n \in \mathbb{Z}^+$ with $m \geq n$, and let $a \in \{0, 1, \ldots, n-1\}$ be the unique integer determined by $\frac{m+a}{n} \in \mathbb{Z}^+$. Then*

$$(\lfloor \frac{m+0}{n} \rfloor, \lfloor \frac{m+1}{n} \rfloor, \ldots, \lfloor \frac{m+n-1}{n} \rfloor)$$

is a unique balanced n-partition of m, consisting of a integers $\frac{m+a}{n} - 1$ and $n - a$ integers $\frac{m+a}{n}$.

It follows immediately that

$$\left.\begin{array}{l} (c_n(1), c_n(2), \ldots, c_n(n)), \\ \\ c_n(i) = 2\lfloor \frac{2^{n-2}+i-1}{n} \rfloor, \quad 1 \leq i \leq n, \end{array}\right\} \tag{5.2}$$

is a unique balanced *even* n-partition of 2^{n-1} for $n > 3$.

We are now ready to prove that there exists a balanced cyclic half Gray sequence $G_h(n)$ for $n > 4$. Assume that there is such a sequence for some fixed value $n > 4$. Because of the uniqueness of the partition (5.2), the transition count spectrum of this $G_h(n)$ must be equivalent to (5.2), up to a permutation of bit positions. Now, by applying the construction of Theorem 3.2.2, we want to construct a balanced cyclic half Gray sequence $G_h(n+2)$ with a transition count spectrum which is, up to a permutation of the bit positions, equivalent to

$$\left.\begin{array}{l} (c_{n+2}(1), c_{n+2}(2), \ldots, c_{n+2}(n+2)), \\ c_{n+2}(i) = 2\lfloor \frac{2^n+i-1}{n+2} \rfloor, \quad 1 \leq i \leq n+2. \end{array}\right\} \tag{5.3}$$

Since the construction described in Theorem 3.2.2 always yields transition count spectra such that the last two numbers are equal, we distinguish between case (i) $c_{n+2}(n+2) = c_{n+2}(n+1)$ and case (ii) $c_{n+2}(n+2) > c_{n+2}(n+1)$. In case (ii) we interchange $c_{n+2}(n+2)$ and $c_{n+2}(n)$, and consider the distribution ($c_{n+2}(1)$, $c_{n+2}(2)$, $\ldots, c_{n+2}(n+2)$, $c_{n+2}(n+1)$, $c_{n+2}(n)$). In this distribution the last two numbers are equal. Furthermore, if necessary, we shift in both cases the complete transition sequence $S_h(n)$ cyclicly such that the integer n becomes its closing transition. In order to construct $S_h(n+2)$ with a transition count spectrum equivalent to ($c_{n+2}(1)$, $c_{n+2}(2)$, $\ldots, c_{n+2}(n+2)$) by using Theorem 3.2.2 with $S_h(n)$ as basis of the construction, we have to chose $b(i)$ integers i in $S_h(n)$, where $b(i)$ is defined as

$$b(i) = \begin{cases} \frac{4c_n(i)-c_{n+2}(i)}{2}, & 1 \leq i < n, \\ \frac{4(c_n(i)-1)-c_{n+2}(i)}{2}, & i = n, \end{cases} \tag{5.4}$$

where $n = s_{2^{n-1}}$ is the closing transition of $S_h(n)$ (cf. eq. (5.1)). The condition to make such a choice possible is that $0 \leq b(i) \leq c_n(i)$ for all $1 \leq i \leq n-1$, and $0 \leq b(n) \leq c_n(n) - 1$.
Using the expressions for $c_n(i)$ and $c_{n+2}(i)$ in (5.2) and (5.3), we can easily see that in cases (i) and (ii) for $1 \leq i \leq n-1$ and $n \geq 5$, one has

$$b(i) = 4\lfloor \frac{2^{n-2}+i-1}{n} \rfloor - \lfloor \frac{2^n+i-1}{n+2} \rfloor \geq 0, \tag{5.5}$$

$$\begin{array}{rl} b(n) & \geq 4\lfloor \frac{2^{n-2}+n-1}{n} \rfloor - 2 - \lfloor \frac{2^n+n+2-1}{n+2} \rfloor \\ & = 4\lfloor \frac{2^{n-2}-1}{n} \rfloor - \lfloor \frac{2^n-1}{n+2} \rfloor + 1 \geq 0, \end{array} \tag{5.6}$$

where the $>$ sign in the first inequality of (5.6) holds in case (i) and the $=$ sign in case (ii). On the other hand, we also have for $1 \leq i \leq n-1$ and for all $n \geq 5$ in both cases

$$b(i) = 4\lfloor\frac{2^{n-2}+i-1}{n}\rfloor - \lfloor\frac{2^n+i-1}{n+2}\rfloor \leq 2\lfloor\frac{2^{n-2}+i-1}{n}\rfloor = c_n(i), \tag{5.7}$$

$$\begin{aligned} b(n) &\leq 4\lfloor\tfrac{2^{n-2}+n-1}{n}\rfloor - 2 - \lfloor\tfrac{2^n+n-1}{n+2}\rfloor \\ &\leq 2\lfloor\tfrac{2^{n-2}+n-1}{n}\rfloor - 1 = c_n(n) - 1, \end{aligned} \tag{5.8}$$

where the $<$ sign in the first inequality of (5.8) only holds in case (ii).

From Eqs. (5.5)-(5.8) and from the fact that the construction of Theorem 3.2.2 does not produce complementary codewords when starting from an $S_h(n)$ (see subsection 5.1.1), it now follows that we can chose $b(i)$ integers i from the sequence $S_h(n)$, $1 \leq i \leq n$, and hence there exists a sequence $S_h(n+2)$ as soon as we have a sequence $S_h(n)$. Since we know already that there exist sequences $S_h(5)$ and $S_h(6)$ (Fig. 5.3), we may conclude by applying mathematical induction that the following theorem holds.

Theorem 5.1.10. *A balanced cyclic half Gray sequence $G_h(n)$ exists for every $n \geq 5$.*

The maximum sequences $M_2 = 00, 11, 01, 10$ and M_3 in Example 5.1.1 are both balanced. So, due to the above observation, we have the following theorem which proves the Robinson-Cohn conjecture in [60], formulated as Conjecture 5.1.3 in this section.

Theorem 5.1.11. *An n-bit balanced maximum counting sequence exists for every $n > 1$, $n \neq 4$.*

5.2 Uniform counting sequences

Let $\mathcal{O}(n)$ be a counting sequence of length n(cf. Section 1.2) with transition sequence $\bar{S}_{\mathcal{O}(n)}(n) = s_1, s_2, \ldots, s_{2^n}$. If for every $i \in \{1, 2, \ldots, 2^n\}$ we have that $|s_i| = t$, for some $t \in \{1, 2, \ldots, n-1\}$, we call the sequence $\mathcal{O}(n)$ a *uniform counting sequence*. A uniform counting sequence of length n with $|s_i| = t$ is frequently called (n, t)-*counting sequence* or, shortly an (n, t)-*sequence*. Uniform counting sequences have applications in many areas such as testing and fault diagnosis in combinational logic circuits (cf. [25, 60]). The notion of uniform counting sequence generalizes the notion of cyclic Gray code where any two successive codewords differ in precisely one bit position. It is easy to understand that a uniform sequence exists only for odd values of t. This is because the parity of the weight of successive codewords remains the same for even t(see e.g. [32, p. 35] and [60]).

The counting sequence shown in Figure 5.2 is a balanced uniform counting sequence of length 4 with transition sequence equal to

$$\begin{aligned} \bar{S}(4,3) = \; & \{1,2,3\},\{2,3,4\},\{1,2,3,\},\{1,3,4\},\{1,2,4\},\{1,3,4\},\{1,2,3\},\{2,3,4\}, \\ & \{1,3,4\},\{2,3,4\},\{1,2,4\},\{2,3,4\},\{1,2,3\},\{1,2,4\},\{1,3,4\},\{1,2,4\}, \end{aligned}$$

0000 1100
0111 0001
1001 1111
1110 0100
0011 1010
1000 1101
0101 0110
0010 1011.

Figure 5.4: A $(4, 3)$-sequence

and with transition count spectrum $TC_{(4,3)} = (12, 12, 12, 12)$.

With respect to this structure Robinson and Cohn in [60] defined a *realizable pair* as a pair of integers (n, t) with $1 \leq t < n$ and t odd, and they made the following conjecture.

Conjecture 5.2.1 (Robinson-Cohn [60]). *For every realizable pair of n and t, $2 \leq t < n$, a balanced (n, t)-sequence exists.*

Of course, $(n, 1)$-sequences exist for all $n \geq 1$, since these are identical to cyclic Gray codes.

Robinson and Cohn in [60] gave a method for the construction of uniform counting sequences based on linear codes. They arrange the codewords of a linear code $[n, k, t]$-code such that each pair of successive codewords have minimum distance t, by making use of a minimum weight basis and by a normal Gray code $G(k)$ for the enumeration of all linear combinations of the basis codewords. To obtain all codewords of length n, this arrangement is followed by its cosets which are also built up in such a way that the uniformity with respect to the Hamming distance between any two successive codewords is maintained (cf. also [81, 84, 82], where the same principle is applied). This construction however, does not guarantee that the resulting sequences are balanced sequences.

Knuth in [32, p. 88] also introduced a non-recursive technique for obtaining uniform counting sequences of length n. Using Knuth's method, a uniform counting sequence of length n is obtained the words of which are the images of a special mapping defined on the ordinary binary number system of length n. But again, the resulting uniform counting sequences are not balanced.

In this section we introduce two constructions (Construction 5.2 and Construction 5.3 on the next pages) for obtaining uniform sequences. One of these can be applied for any realizable pair of codeword length n and Hamming distance t, whereas the second one only works for some realizable pairs of n and t. The resulting uniform sequences produced by the first construction are far from being balanced, but the simplicity of the construction is of considerable interest. The second construction can produce balanced uniform sequences whenever n is a prime or a power of two. Before coming to the formulation of the constructions, we first discuss a special uniform counting sequence which could be called an *anti-Gray code*, i.e. a $(2m, 2m - 1)$-

sequence. The following construction was introduced by Robinson and Cohn in [60].

Construction 5.2(Robinson and Cohn [60])

1. Start with a cyclic Gray code $G(2m)$ of length $2m$,
2. Construct the sequence $U(2m)$ by complementing every other codeword.

The resulting sequence $U(2m)$ is a $(2m, 2m-1)$-sequence.

Example 5.2.1. Let us consider the standard Gray code $G(4)$ of length 4. Construction 5.2 will produce $U(4)$ as listed below. The boldface words in $U(4)$ are obtained by complementing their counterparts in $G(4)$. Vice versa, $G(n)$ can be obtained from $U(n)$ in precisely the same way by complementing every other codeword.

$G(4)$		$U(4)$	
0000	1100	0000	1100
0001	1101	**1110**	**0010**
0011	1111	0011	1111
0010	1110 $\leftrightarrow$	**1101**	**0001**
0110	1010	0110	1010
0111	1011	**1000**	**0100**
0101	1001	0101	1001
0100	1000	**1011**	**0111**

We formulate a construction equivalent to Construction 5.2 using a different terminology. We define the so-called *anti-transition sequence* $AS(2m)$ of a $(2m, 2m-1)$-sequence to be equal to the transition sequence $\bar{S}(2m)$ of the corresponding cyclic Gray code of length $2m$, the elements of which indicate the bit positions which do not change. The formulation is described as the following Construction 5.2′.

Construction 5.2′

1. Take a transition sequence $\bar{S}(2m)$ of a Gray code of length $2m$, $m \geq 1$,
2. Start with the zero codeword of length $2m$, and apply $\bar{S}(2m)$ as an anti-transition sequence $AS(2m)$ to produce a complete sequence of codewords $U(2m)$.

As an example let us consider Example 5.2.1. The transition sequence of $G(4)$ in Example 5.2.1 is

$$\bar{S}(4) = 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 4.$$

If we construct a sequence starting from the zero codeword 0000 and apply $AS(4) := \bar{S}(4)$, then again we obtain the same sequence $U(4)$ as in Example 5.2.1.
We now study the transition counts of the resulting sequence $U(2m)$ using Construction 5.2′. It is clear that the cyclic sequence 0 1 0 1 0 1 ... 0 1 of period 2^n has 2^n bit changes. It is also obvious that in a cyclic sequence there is an even number of

bit changes. This is equivalent to the fact that each integer in a transition sequence $\bar{S}(n)$ occurs an even number of times. In a uniform counting sequence of codeword length n with anti-transition sequence $AS(n)$, the occurrence of a transition i means that when going to the next codeword in the list, the only bit which is maintained is the bit in position i. Let $Ab(i)$ be the number of integers i which occur in the anti-transition sequence $AS(n)$. So, the transition count of bit position i in the anti Gray code, i.e. the number of times that the bit in position i changes, will be equal to $2^n - Ab(i)$. Notice that $Ab(i)$ is equal to $TC_n(i)$ in $\bar{S}(n)$. From this observation, it will be clear now that if we have a balanced Gray code of length $n = 2m$ with transition sequence $\bar{S}(n)$, then by taking $AS(n) := \bar{S}(n)$ we obtain an $(n, n-1)$-sequence with $|c_n(i) - c_n(j)| \leq 2$, where $c_n(i) = 2^n - Ab(i)$ is the transition count of the integer i in $AS(n)$. Since for every $n \geq 1$, a balanced Gray code of length n exists, we have the following theorem.

Theorem 5.2.2. *For every $m \geq 1$, there exists a balanced uniform $(2m, 2m-1)$-sequence, and if m is a power of two, there exists a totally balanced uniform $(2m, 2m-1)$-sequence.*

Instead of Conjecture 5.2.1 the remaining (weaker) conjecture can be formulated as follows

Conjecture 5.2.3. *For every realizable pair (n, t), $3 \leq t < n-1$, a balanced (n, t)-sequence exists.*

The following is an obvious property of a $(2m, 2m-1)$-sequence, $m \geq 1$.

Theorem 5.2.4. *Every two codewords sandwiching another codeword in a $(2m, 2m-1)$-sequence have Hamming distance 2.*

Proof. Let $\mathbf{x}$ and $\mathbf{y}$ be two codewords in the $(2m, 2m-1)$-sequence sandwiching codeword $\mathbf{w}$. Let i be the bit position where the only components of $\mathbf{x}$ and $\mathbf{w}$ are equal and let j be the bit position where the only component of $\mathbf{w}$ and $\mathbf{y}$ are equal. Since $\mathbf{x}$ and $\mathbf{y}$ are different codewords, $i \neq j$. It is clear that the bits of $\mathbf{x}$ and $\mathbf{y}$ are the same at the positions $[2m] \backslash \{i, j\}$ and differ otherwise. So, the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$ is 2. □

Now we describe a construction for an (n, t)-sequence for every pair of relevant values of n and t.

Construction 5.3

1. Let $U(n, 2m-1) := \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{2^n-1}$ be an $(n, 2m-1)$-sequence, $n \geq 2m$.
2. Take a codeword $\mathbf{z}$ such that $d(\mathbf{x}_{2^n-1}, \mathbf{z}) = 2m-2$.
3. Let the sequence $V(n, 2m-1) := \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{2^n-1}$, where $\mathbf{y}_i = \mathbf{x}_i \oplus \mathbf{z} \oplus \mathbf{x}_{2^n-1}\}$, for all $i \in \{0, 1, \ldots, 2^n - 1\}$.

4. $U := 0\mathbf{x}_0, 1\mathbf{y}_0, 1\mathbf{y}_1, 0\mathbf{x}_1, 0\mathbf{x}_2, 1\mathbf{y}_2, 1\mathbf{y}_3, 0\mathbf{x}_3, \ldots, 0\mathbf{x}_{2^n-2}, 1\mathbf{y}_{2^n-2}, 1\mathbf{y}_{2^n-1}, 0\mathbf{x}_{2^n-1}$.

It is obvious that U is an $(n+1, 2m-1)$-sequence.

Example 5.2.2. In this example we produce $U(5,3)$ starting from $U(4,3)$.

$U(4,3)$		$V(4,3)$			$U(5,3)$	
0000		0011			00000	01100
1110		1101			10011	11111
0011		0000			11101	10001
1101		1110			01110	00010
0110		0101			00011	01111
1000		1011			10000	11100
0101		0110			11110	10010
1011		1000		$\longrightarrow$	01101	00001
1100		1111			00110	01010
0010		0001			10101	11001
1111		1100			11011	10111
0001		0010			01000	00100
1010		1001			00101	01001
0100		0111			10110	11010
1001		1010			11000	10100
0111	$\leftarrow \mathbf{x}_{2^n-1}$	0100	$\leftarrow \mathbf{z}$		01011	00111

Theorem 5.2.5. *For any positive integer m and for $n > 2m$, Construction 5.3 can be used to construct an $(n, 2m-1)$-sequence.*

Proof. The proof of the Theorem follows from the existence of a $(2m, 2m-1)$-sequence for every positive integer m. □

In fact, if we start with an $n-2$-bit uniform counting sequence, Theorem 5.1.6 and Theorem 4.2.3 can be used to construct uniform counting sequences by replacing integers $n-1$ and n by the sets $\{n-1\} \cup I$ and $\{n\} \cup I$ respectively, where I is a subset of $[n-2]$ which contains $t-1$ elements.

5.2.1 An alternative construction for (n,t)-sequences

We noticed already that none of the mentioned constructions for constructing uniform counting sequences guarantees the production of *balanced* uniform sequences. In this subsection we shall introduce an alternative construction for obtaining uniform counting sequences which has the advantage that in some cases it enables us to produce balanced uniform counting sequences. To this end, we first need to discuss the concept of – what we call – *regular distribution*, and which will be the subject of the following subsection.

5.2.1.1 Regular distribution

In this subsection we consider ordered multi-sets of two integers a and b. We call such sets *distributions* of a and b. A special distribution is

$$\mathcal{D}_0 = (\underbrace{a, ..., a}_{k_a}, \underbrace{b, ..., b}_{k_b}) \tag{5.9}$$

which contains k_a integers a followed by k_b integers b. For possible applications we consider a distribution $\mathcal{D}$ as a *cyclic list*, i.e. the first integer in $\mathcal{D}$ is the successor of the last integer, and we require the integers a and b to be arranged as "regularly" as possible. To this end we introduce the notion of t-block. A t-block Γ in a distribution $\mathcal{D}$ consists of t consecutive elements of $\mathcal{D}$, with $t \leq |\mathcal{D}|$.

Example 5.2.3. Let $\mathcal{D}$ be the distribution (a, b, b, b, a, a, b, a). With respect to $\mathcal{D}$, we have that $\Gamma_1 = a, b, b$ and $\Gamma_2 = b, b, a$ are 3-blocks, while $\Gamma_3 = b, b, a, a$ *and* $\Gamma_4 = a, b, a, a$ *are* 4*-blocks.*

Definition 5.2.1. *If $\mathcal{D}$ is a distribution of the integers a and b, then $\mathcal{D}$ is called regular if for any two t-blocks the numbers of integers b in these blocks (and hence also the numbers of integers a) differ at most by 1, for any fixed value of t, $1 \leq t \leq |\mathcal{D}|$.*

Example 5.2.4. The distribution $\mathcal{D} = (a, b, a, b, a, b, b, a, b, a, b, b)$ is clearly regular. The distribution $\mathcal{D} = (a, b, b, a, b, b, a, b, a, b, a, b)$ is not regular, since it contains 5-blocks b, b, a, b, b and a, b, a, b, a which contain 4 and 2 integers, respectively.

It appears that any distribution $\mathcal{D}$ of integers a and b can be rearranged such that it is regular. Such an arrangement can be obtained by carrying out the following recursive procedure. We start with a distribution (5.9) with k_a integers a and k_b integers b.

Construction 5.4

1. Define $m_1 = k_a$ and $\bar{m}_1 = k_b$, and introduce m_1 blocks $\mathcal{B}_1 = a$ and $\bar{m}_1$ blocks $\bar{\mathcal{B}}_1 = b$.
2. Introduce a series of new blocks $\mathcal{B}_{i+1}$ and $\bar{\mathcal{B}}_{i+1}$, $i > 0$, in the following way. If $m_i \geq \bar{m}_i$, then define $m_{i+1} = m_i - \bar{m}_i$ and $\bar{m}_{i+1} = \bar{m}_i$. Introduce m_{i+1} blocks $\mathcal{B}_{i+1} = \mathcal{B}_i$ and $\bar{m}_{i+1}$ blocks $\bar{\mathcal{B}}_{i+1} = \mathcal{B}_i\bar{\mathcal{B}}_i$.
   If $m_i < \bar{m}_i$, then define $m_{i+1} = m_i$ and $\bar{m}_{i+1} = \bar{m}_i - m_i$. Introduce m_{i+1} blocks $\mathcal{B}_{i+1} = \bar{\mathcal{B}}_i\mathcal{B}_i$ and $\bar{m}_{i+1}$ blocks $\bar{\mathcal{B}}_{i+1} = \bar{\mathcal{B}}_i$.
   Repeat this process until for some k one has either $\bar{m}_k = 0$ or $m_k = 0$.
3. Define the resulting distribution as

$$\mathcal{D} = (\underbrace{\mathcal{B}_k, \ldots, \mathcal{B}_k}_{m_k})$$

   or

$$\mathcal{D} = (\underbrace{\bar{\mathcal{B}}_k, \ldots, \bar{\mathcal{B}}_k}_{\bar{m}_k}).$$

Example 5.2.5. Start with distribution (5.9) with $k_a = 10$ and $k_b = 7$. The various steps carried out when applying Construction 5.4 deliver the following results.

1. $m_1 = 10$, $\bar{m}_1 = 7$,
   $\mathcal{B}_1 = a, \quad \bar{\mathcal{B}}_1 = b;$
2. $m_2 = 3$, $\bar{m}_2 = 7$,
   $\mathcal{B}_2 = a, \quad \bar{\mathcal{B}}_2 = a, b;$
3. $m_3 = 3$, $\bar{m}_3 = 4$,
   $\mathcal{B}_3 = a, b, a, \quad \bar{\mathcal{B}}_3 = a, b;$
4. $m_4 = 3$, $\bar{m}_4 = 1$,
   $\mathcal{B}_4 = a, b, a, b, a, \quad \bar{\mathcal{B}}_4 = a, b;$
5. $m_5 = 2$, $\bar{m}_5 = 1$,
   $\mathcal{B}_5 = a, b, a, b, a, \quad \bar{\mathcal{B}}_5 = a, b, a, b, a, a, b;$
6. $m_6 = 1$, $\bar{m}_6 = 1$,
   $\mathcal{B}_6 = a, b, a, b, a, \quad \bar{\mathcal{B}}_6 = a, b, a, b, a, a, b, a, b, a, a, b;$
7. $m_7 = 0$, $\bar{m}_7 = 1$,
   $\bar{\mathcal{B}}_7 = a, b, a, b, a, a, b, a, b, a, a, b, a, b, a, b, a;$
8. $\mathcal{D} = \bar{\mathcal{B}}_7$

We shall now prove that Construction 5.4 always leads to a regular distribution.

Lemma 5.2.6. *For the intermediate states of Construction 5.4 the following relations hold for all i with $1 < i \leq k$:*

(i) (a) $\bar{\mathcal{B}}_i = \underbrace{\mathcal{B}_i, \ldots, \mathcal{B}_i}_{\geq 1}, \mathcal{B}_i^l$, *if* $|\bar{\mathcal{B}}_i| > |\mathcal{B}_i|$;

(b) $\mathcal{B}_i = \underbrace{\bar{\mathcal{B}}_i, \ldots, \bar{\mathcal{B}}_i}_{\geq 1}, \bar{\mathcal{B}}_i^l$, *if* $|\mathcal{B}_i| > |\bar{\mathcal{B}}_i|$,

where $\mathcal{B}_i^l (\neq \emptyset, \mathcal{B}_i)$ is a left subblock of $\mathcal{B}_i := \mathcal{B}_i^l, \mathcal{B}_i^r$, and likewise $\bar{\mathcal{B}}_i^l (\neq \emptyset, \bar{\mathcal{B}}_i)$ is a left subblock of $\bar{\mathcal{B}}_i := \bar{\mathcal{B}}_i^l, \bar{\mathcal{B}}_i^r$;

(ii) (a) $\mathcal{B}_i^l = \underbrace{\mathcal{B}_i^r, \ldots, \mathcal{B}_i^r}_{\geq 1}, \mathcal{B}_i^{rl}$, *if* $|\mathcal{B}_i^l| > |\mathcal{B}_i^r| > 1$;

(b) $\mathcal{B}_i^r = \underbrace{\mathcal{B}_i^l, \ldots, \mathcal{B}_i^l}_{\geq 1}, \mathcal{B}_i^{ll}$, *if* $|\mathcal{B}_i^r| > |\mathcal{B}_i^l| > 1$,

and similar relations for $\bar{\mathcal{B}}_i^l$ and $\bar{\mathcal{B}}_i^r$.

Proof. (i) For $i = 2$ this certainty is true. Assume that it is true for some $i, 1 < i < k$. If $m_i \geq \bar{m}_i$, the construction yields $\mathcal{B}_{i+1} = \mathcal{B}_i$ and $\bar{\mathcal{B}}_{i+1} = \mathcal{B}_i, \bar{\mathcal{B}}_i$. Hence,

$$\bar{\mathcal{B}}_{i+1} = \mathcal{B}_i, \mathcal{B}_i, \ldots, \mathcal{B}_i, \mathcal{B}_i^l = \mathcal{B}_{i+1}, \mathcal{B}_{i+1}, \ldots, \mathcal{B}_{i+1}, \mathcal{B}_{i+1}^l,$$

in case (a), and

$$\bar{\mathcal{B}}_{i+1} = \mathcal{B}_i, \bar{\mathcal{B}}_i = \mathcal{B}_{i+1}, \bar{\mathcal{B}}_i = \mathcal{B}_{i+1}, \mathcal{B}_i^l = \mathcal{B}_{i+1}, \mathcal{B}_{i+1}^l,$$

in case (b). If $m_i < \bar{m}_i$, we can argue similarly. So, relations (a) and (b) hold for all $i, 1 < i \leq k$.

(ii) Let $|\mathcal{B}_i^l| > |\mathcal{B}_i^r|$. From the proof of (i) we know that $\mathcal{B}_i^l = \mathcal{B}_j$ and $\mathcal{B}_i^r = \bar{\mathcal{B}}_j$ or $\mathcal{B}_i^l = \bar{\mathcal{B}}_j$ and $\mathcal{B}_i^r = \mathcal{B}_j$, for some $j < i$. We consider, without restriction of the generality, the first possibility. Then we can write by applying the results in (i)

$$\mathcal{B}_i^l = \mathcal{B}_j = \bar{\mathcal{B}}_j, \ldots, \bar{\mathcal{B}}_j, \bar{\mathcal{B}}_j^l = \mathcal{B}_i^r, \ldots, \mathcal{B}_i^r, (\mathcal{B}_i^r)^l.$$

All other cases can be proved similarly. □

Lemma 5.2.7. *Any distribution $\mathcal{D}$ of integers a and b can be rearranged such that it is regular.*

Proof. We shall prove the Lemma by mathematical induction, showing that after having carried out step 2 of Construction 5.4 for the i-th time, we have a regular distribution

$$\mathcal{D}_i = (\mathcal{B}_i, \mathcal{B}_i, \ldots, \mathcal{B}_i)$$

of size $m_i|\mathcal{B}_i|$, as well as a regular distribution

$$\bar{\mathcal{D}}_i = (\bar{\mathcal{B}}_i, \bar{\mathcal{B}}_i, \ldots, \bar{\mathcal{B}}_i)$$

of size $\bar{m}_i|\bar{\mathcal{B}}_i|$, for all values i with $1 \leq i \leq k$.

For $i = 1$ and $i = 2$ this property is trivially true. Assume that the property is true for all values less than or equal to some i, $1 < i < k$. This induction assumption implies that after iteration i we have the following regular distributions, written schematically and omitting indices i and $i - 1$.

$$\cdots \underbrace{\bar{\mathcal{B}} \; \bar{\mathcal{B}} \; \bar{\mathcal{B}} \; \bar{\mathcal{B}}} \cdots \underbrace{\bar{\mathcal{B}} \; \bar{\mathcal{B}}} \cdots \qquad (i)$$

$$\cdots \underbrace{\mathcal{B} \; \mathcal{B} \; \mathcal{B} \; \mathcal{B}} \cdots \underbrace{\mathcal{B} \; \mathcal{B}} \cdots \qquad (ii)$$

Here, $\mathcal{B}$ stands for $\mathcal{B}_i$ and $\bar{\mathcal{B}}$ for $\bar{\mathcal{B}}_i$, and we consider the sequences in (i) and (ii) as being cyclic sequences.
In order to prove the regularity of $\mathcal{D}_{i+1}$, and $\bar{\mathcal{D}}_{i+1}$, we only have to show that the following cyclic sequence represents a regular distribution

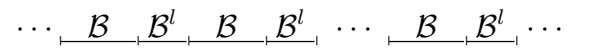
$$\cdots \underbrace{\mathcal{B} \; \bar{\mathcal{B}} \; \mathcal{B} \; \bar{\mathcal{B}}} \cdots \underbrace{\mathcal{B} \; \bar{\mathcal{B}}} \cdots \qquad (iii)$$

Because of the periodicity of (iii), we can restrict ourselves to the comparison of two t-blocks for t-values satisfying $0 < t < |\mathcal{B}| + |\bar{\mathcal{B}}|$.

Let $|\mathcal{B}| < |\bar{\mathcal{B}}|$. Applying Lemma 5.2.6 $(i)(a)$, we can write for (iii)

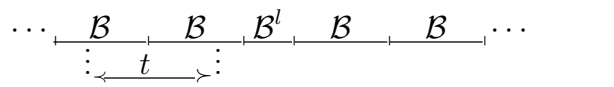
$$\cdots \underbrace{\mathcal{B}} \cdots \underbrace{\mathcal{B} \; \mathcal{B}^l \; \mathcal{B} \; \mathcal{B}} \cdots \underbrace{\mathcal{B} \; \mathcal{B} \; \mathcal{B}^l} \cdots \qquad (iv)$$

The induction assumption also implies that the following picture represents a regular distribution

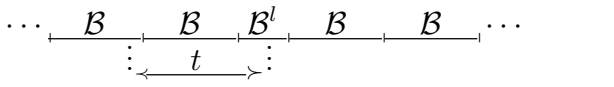


(v)

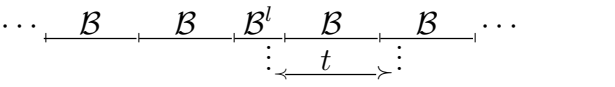
Here, we applied the regularity for some $j < i$. So, for $t \leq |\mathcal{B}|$ the two t-blocks satisfy the regularity condition.
For $|\mathcal{B}| < t < |\mathcal{B}| + |\bar{\mathcal{B}}| = p \cdot |\mathcal{B}| + |\mathcal{B}^l|$, we have the following typical situations for the t-blocks(for reasons of simplicity we take $p = 2$)



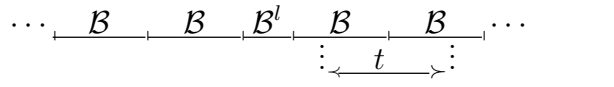
(vi)



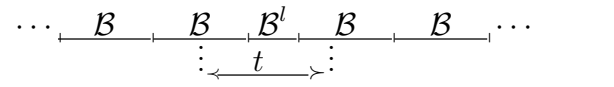
(vii)



$(viii)$

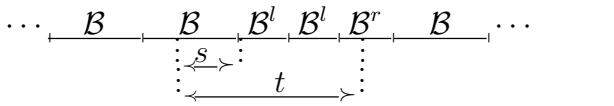


(ix)



(x)

If the two t-blocks are both of one of the types (vi)-(ix), we can remove a complete $|\mathcal{B}|$-block from both blocks either from the left end or from the right end, reducing the t-blocks to $(t - |\mathcal{B}|)$-blocks, which satisfy the regularity condition because of (v). If both t-blocks are of type (x), we remove from both the complete $\mathcal{B}^l$-block, reducing the situation to the regularity of (ii). If one of the t-blocks is of type (x) and the other one of type (vii) or $(viii)$, we remove from both blocks their intersection, yielding reduced blocks which satisfy the regularity condition because of (v).
The only remaining case is that one t-block is of type (x) and the other one of type (vi) or type (ix). We rewrite (x) as



(xi)

It follows from Lemma 5.2.6 (ii) that in all cases ($|\mathcal{B}^l| > |\mathcal{B}^r|$ and $|\mathcal{B}^r| > |\mathcal{B}^l|$), the subsequence right after the first $\mathcal{B}^l$-block starts with $\mathcal{B}^r$ possibly apart from the last element of $\mathcal{B}^r$. Since $s \geq 1$, this implies that the t-block starts at the left hand

side with a block of size $|\mathcal{B}|$ the contents of which is precisely equal to the contents of $\mathcal{B}$ itself. Removing this block and removing a complete $\mathcal{B}$-block from (vi)(or from (ix)) reduces the two t-blocks to $(t - |\mathcal{B}|)$-blocks which are part of (ii) and which satisfy the regularity condition by our induction assumption. Hence, the original two t-blocks satisfy this condition as well. So, we proved in all cases that the regularity condition also holds right after the $(i + 1)$-st iteration. Therefore, the Lemma holds for $1 < i \leq k$. □

By carefully observing the iteration steps when carrying out Construction 5.4, one will notice that the process of determining the successive values m_i and $\bar{m}_i$, $1 \leq i \leq k$, is rather close to the computation of the remainders in the successive iterations of Euclid's algorithm, when determining the greatest common divisor of k_a and k_b. Based on this observation we present the following algorithm, which is equivalent to Construction 5.4, but which takes less iteration steps. We assume w.r.t. that $k_a \geq k_b$.

Construction 5.4′

1. Define $m_1 = k_a$, $\bar{m}_1 = k_b$, and determine the pair of integers (q_1, r_1) by the relations $m_1 = q_1\bar{m}_1 + r_1$, $0 \leq r_1 < \bar{m}_1$. Introduce m_1 blocks $\mathcal{B}_1 = a$ and $\bar{m}_1$ blocks $\bar{\mathcal{B}}_1 = b$.
2. Repeat the following process for $i = 1, \ldots, l$, where l is such that $r_l = 0$.
   Define $m_{i+1} = \bar{m}_i$, $\bar{m}_{i+1} = r_i$, and determine the pair of integers (q_{i+1}, r_{i+1}) by the relations $m_{i+1} = q_{i+1} \cdot \bar{m}_{i+1} + r_{i+1}$, $0 \leq r_{i+1} < \bar{m}_{i+1}$. Introduce m_{i+1} blocks $\mathcal{B}_{i+1} = \mathcal{B}_i, \bar{\mathcal{B}}_i, \underbrace{\mathcal{B}_i, \ldots, \mathcal{B}_i}_{q_i - 1}$ and $\bar{m}_{i+1}$ blocks $\bar{\mathcal{B}}_{i+1} = \mathcal{B}_i$.
3. Define $m_{l+1} = \bar{m}_l$ and introduce $\mathcal{B}_{l+1} = \mathcal{B}_l, \bar{\mathcal{B}}_l, \underbrace{\mathcal{B}_l, \ldots, \mathcal{B}_l}_{q_l - 1}$.
4. $\mathcal{D} = (\underbrace{\mathcal{B}_{l+1}, \ldots, \mathcal{B}_{l+1}}_{m_{l+1}})$.

The reader should be aware of the fact that the blocks in Construction 5.4′ are different from the blocks in Construction 5.4.

Example 5.2.6. Start with the distribution (5.9) with $k_a = 51$ and $k_b = 15$. The various steps carried out when applying Construction 5.4′ yield the following results:

1. $m_1 = 51$, $\bar{m}_1 = 15$, $q_1 = 3, r_1 = 6$,
$\mathcal{B}_1 = a, \quad \bar{\mathcal{B}}_1 = b$;
2. $m_2 = 15$, $\bar{m}_2 = 6$, $q_2 = 2, r_2 = 3$,
$\mathcal{B}_2 = a, b, a, a, \quad \bar{\mathcal{B}}_2 = a$;
3. $m_3 = 6$, $\bar{m}_3 = 3$, $q_3 = 2, r_3 = 0$,
$\mathcal{B}_3 = a, b, a, a, a, a, b, a, a, \quad \bar{\mathcal{B}}_3 = a, b, a, a$;
4. $m_4 = 3$,
$\mathcal{B}_4 = a, b, a, a, a, a, b, a, a, a, b, a, a, a, b, a, a, a, a, b, a, a$;
5. $\mathcal{D} = (\mathcal{B}_4, \mathcal{B}_4, \mathcal{B}_4)$.

It will be obvious that in general the resulting regular distribution is periodic with periodicity m_{l+1} being the greatest common divisor of k_a and k_b. A proof that

Constructions 5.4 and 5.4′ are equivalent is rather easy, and is omitted.
We can verify that if we define $\mathcal{B}_{i+1} = \underbrace{\mathcal{B}_i, \ldots, \mathcal{B}_i}_{q_i}, \bar{\mathcal{B}}_i$ instead of $\mathcal{B}_{i+1} = \mathcal{B}_i, \bar{\mathcal{B}}_i, \underbrace{\mathcal{B}_i, \ldots, \mathcal{B}_i}_{q_i - 1}$ in Construction 5.4′, then the resulting distribution is also regular.

An immediate consequence of Lemma 5.2.7 is the following.

Corollary 5.2.8. *If $|b-a| = 2$ in Lemma 5.2.7, then the absolute value of the difference between the sums of the components of any two t-blocks in a regular distribution is at most 2.*

5.2.1.2 A construction of uniform counting sequences based on transition sequences of Gray codes

Let us consider Lemma 2.3.1 again in Subsection 2.3.1 which is due to Gilbert [20]. Below we shall formulate another characterization of such a sequence S to be a transition sequence of a cyclic Gray code. The relevance of this characterization will appear in Subsection 5.2.1.3.

Consider a sequence $S = s_1, s_2, ..., s_{2^n}$, $s_i \in [n]$ for every $i \in [2^n]$. We define

$$S_i := s_1, ..., s_i, \quad 1 \leq i \leq 2^n, \text{ and } S_0 = \emptyset; \tag{5.10}$$

$$O(S_i) := \{x \in [n] | x \text{ occurs an odd number of times in } S_i\}. \tag{5.11}$$

Let A and B be two sets and let $\dot{-}$ stand for the symmetric difference of two sets, i.e. $x \in A \dot{-} B$ if and only if either $x \in A, x \notin B$ or $x \notin A, x \in B$. The next lemma is obvious.

Lemma 5.2.9. *Let u, v, w be sequences consisting of integers from $[n]$ such that v is equal to the concatenated sequence w, u. Then we have that $O(u) = O(v) \dot{-} O(w)$.*

For practical cases, the following theorem can be of advantage to determine whether a sequence is a transition sequence of a cyclic Gray code or not.

Theorem 5.2.10. *The sequence $S = s_1, s_2, ..., s_{2^n}$ is a transition sequence of a cyclic Gray code of length n if and only if $O(S_i) \neq O(S_j)$ for all $1 \leq i \neq j \leq 2^n$, and $O(S_{2^n}) = \emptyset$.*

Proof. First let S be a transition sequence of a cyclic Gray code of length n. So, we immediately have that $O(S_{2^n}) = \emptyset$. Now, take two arbitrary distinct integers i and j, $0 \leq i < j \leq 2^n$, and consider the sequences S_i and S_j. Let $u = S_j - S_i = s_{i+1}, ..., s_j$. Here, the sequence S_j is equal to the concatenated sequence S_i, u. It is clear that u is not the empty sequence. If we have that $O(S_i) = O(S_j)$, based on Lemma 5.2.9 we obtain that $O(u) = \emptyset$, which contradicts the fact that S is a transition sequence of a Gray code. So, it must be the contrary, i.e. that $O(S_j) \neq O(S_i)$.
Now, let $O(S_i) \neq O(S_j)$, for every $0 \leq i \neq j \leq 2^n$, and suppose that S is not a

transition sequence of a cyclic Gray code. Due to Lemma 2.3.1, there exists at least one non-empty proper subsequence u of S such that $O(u) = \emptyset$. Let $u = s_{i+1}, ..., s_j$, where $0 \leq i < j \leq 2^n$. We have that $u = S_j - S_i$. Again according to Lemma 5.2.9, since $O(u) = \emptyset$, we obtain that $O(S_j) = O(S_i)$ which contradicts the assumption that $O(S_j) \neq O(S_i)$ for every $1 \leq i \neq j \leq 2^n$. □

It should be remarked here that the condition in Theorem 5.2.10 is equivalent to saying that $O(u) \neq \emptyset$ for all proper subsequences of $\bar{S}(n)$.

For a fixed positive integer n, let σ be the cycle $(1\ 2\ \cdots\ n)$. We notice here that for every element a of this cycle σ, and for every integer $i \equiv j(mod\ n)$ we have

$$a + i := \begin{cases} a + j, & 1 \leq a + j \leq n, \\ a + j - n, & \text{otherwise.} \end{cases}$$

Next we define for every element a of σ the integer $a^{(i)} = a + i$, for every integer i. For instance, with respect to the cycle $\sigma = (1\ 2\ 3\ 4\ 5)$, one has $3^{(1)} = 3 + 1 = 4$, $4^{(3)} = 4 + 3 - 5 = 2$, $3^{(-1)} = 3 + 4 - 5 = 2$, and $2^{(-3)} = 2 + 2 = 4$. For each m, t, with $0 < m \leq t < n$, we define a mapping $\Phi_{m|t}$ from the set $[n]$ into the power set $\mathbf{2}^n$ of n, which maps every integer $a \in [n]$ to $\Phi_{m|t}(a) := \{a, a^{(m)}, a^{(m+1)}, ..., a^{(m+t-2)}\}$.
Let us consider the above cycle σ. We have for example $\Phi_{1|3}(2) = \{2, 3, 4\}$, $\Phi_{3|3}(2) = \{2, 5, 1\}$, etc.
Below we introduce a *heuristic* construction for uniform sequences which is based on transition sequences of Gray codes.

Construction 5.5

Let (n, t) be a fixed realizable pair of integers which have a greatest common divisor $gcd(n, t)$ equal to m. Let $\bar{S}(n) = s_1, s_2, ..., s_{2^n}$ be the transition sequence of a Gray code $G(n)$. Start with an n-bit codeword (usually the zero codeword). Apply the sequence $\bar{S}(n, t) = \Phi_{m|t}(s_1), \Phi_{m|t}(s_2), ..., \Phi_{m|t}(s_{2^n})$ to generate the next $2^n - 1$ codewords.

Example 5.2.7. Consider the transition sequence of a balanced Gray code of length $n = 5$,

$$\bar{S}(5) = 2, 1, 3, 1, 2, 1, 3, 4, 3, 5, 3, 4, 3, 1, 4, 5, 2, 1, 5, 1, 4, 1, 3, 4, 5, 1, 2, 5, 2, 4, 2, 5$$

and $t = 3$. Here, $gcd(n, t) = 1$. Then we have that $\bar{S}(5, 3)=$

$\{2, 3, 4\}, \{1, 2, 3\}, \{3, 4, 5\}, \{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 3\}, \{3, 4, 5\}, \{4, 5, 1\}, \{3, 4, 5\}, \{5, 1, 2\}, \{3, 4, 5\}, \{4, 5, 1\}, \{3, 4, 5\}, \{1, 2, 3\}, \{4, 5, 1\}, \{5, 1, 2\}, \{2, 3, 4\}, \{1, 2, 3\}, \{5, 1, 2\}, \{1, 2, 3\}, \{4, 5, 1\}, \{1, 2, 3\}, \{3, 4, 5\}, \{4, 5, 1\}, \{5, 1, 2\}, \{1, 2, 3\}, \{2, 3, 4\}, \{5, 1, 2\}, \{2, 3, 4\}, \{4, 5, 1\}, \{2, 3, 4\}, \{5, 1, 2\}.$

The sequence of codewords of length 5 constructed by using this transition sequence $\bar{S}(5, 3)$ is a balanced $(5, 3)$-sequence with transition count spectrum $(18, 18,\ 20, 20,\ 20)$ and is presented in Fig. 5.5 We shall prove this in the next section.

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

Figure 5.5: The list of a balanced $(5,3)-$sequence.

For the sake of convenience, in the next we shall characterize the set $O(S_i)$ of (5.11) as a cycle of length n, shortly n-cycle, as follows.

Let $O(S_i)$=$\{i_1, i_2, \ldots, i_l\}$, with $1 \leq l \leq n$, and $1 \leq i_1 \leq i_2 \leq \cdots \leq i_l \leq n$. We express $O(S_i)$ as the following cycle

$$(1_e \cdots (i_1 - 1)_e \; i_{1_o} \; (i_1 + 1)_e \cdots (i_l - 1)_e \; i_{l_o} \; (i_l + 1)_e \cdots n_e).$$

The notation j_o(resp. j_e) in the expression means that the integer j occurs an odd(resp. even) number of times in S_i. For example, let us consider the complete transition sequence $\bar{S}(4)$=1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 4 of the binary reflected Gray code of length 4. We take $S_6 = 1, 2, 1, 3, 1, 2$, and hence, $O(S_6) = \{1, 3\}$. In terms of cycles we write $O(S_6) = (1_o \; 2_e \; 3_o \; 4_e)$.

5.2.1.3 Construction of balanced (n,t)-sequences with $gcd(n,t) = 1$

In this subsection we shall discuss the special case of Construction 5.5 when $m = 1$ or equivalently, when n and t are co-prime. If $gcd(n,t) = 1$, $\Phi_{1|t}(a)$:= $\{a, a^{(1)}, a^{(2)}, ..., a^{(t-1)}\}$. Example 5.2.7 is an example of this case. The following observation shows that if $m = 1$, Construction 5.5 will produce an (n,t)-sequence.

For the sake of simplicity we shall write S_i for the sequence $\bar{S}(n)_i$, and write ΦS_i for the sequence $\bar{S}(n,t)_i$.

Since $gcd(n,t) = 1$, one can find integers a and b such that $bn - at = 1$. Suppose that there is some i, $1 \leq i \leq 2^n$, such that $O(\Phi S_i) = \emptyset$. We claim that any t-block in $O(S_i)$ has an even number of integers with index o. Suppose that there is some t-block which has an *odd* number of integers with index o. Say this t-block is equal to(leaving out the indices e and o)

$$k, k+1, \ldots, k+t-1.$$

Consider the parity of the number of integers $k+t-1$ in ΦS_i. Since each integer in the above t-block contributes to the number of integers $k+t-1$ in ΦS_i, and since there are no other integers in S_i doing so, the odd number of integers which have index o yields that the parity of $k+t-1$ is odd. This contradicts the assumption that $O(\Phi S_i) = \emptyset$. It now follows immediately, that when starting from a fixed position in $O(S_i)$, all successive t-blocks have the same pattern with respect to the indices o and e. We shall show that in fact there are no indices o at all in $O(S_i)$.
Consider a sequence of b identical n-cycles $O(S_i)$. Then, starting from the rightmost integer n, mark off to the left, a t-blocks. We may conclude now that there is exactly one integer, that is the leftmost integer 1 of the b n-cycles, which is not covered by these a t-blocks. Since each t-block must have the same pattern w.r.t the indices o and e, the index of the integer 1 must be the same as the index of the integer n, since $O(S_i)$, which is of length n, is considered to be a cyclic sequence. Then, shift $O(S_i)$ cyclicly over one position to the left. So, apart from indices e and o, $O(S_i) = (2, 3, ..., n, 1)$. As mentioned before, the indices of the integers n and 1 are equal. Again, we sequence b identical n-cycles of $O(S_i) = (2, 3, ..., n, 1)$ and mark off a t-blocks to the left, starting from the rightmost integer 1. Because of the same argument, we conclude that the index of the integer 2 is the same as the index of the integer 1. Now we have that the indices of the integers $n, 1$, and 2 are the same. We repeat this process $n-1$ times, and we conclude that all integers in $O(S_i)$ have the same index. Since we proved already that each t-block contains an even number of integers with index o, and since t is an odd integer, all these indices must be e. This can happen only if $i = 2^n$, since S_i is a subsequence of the transition sequence of a Gray code. Now, if S_i is a *real* subsequence of $\bar{S}(n)$, then it contains at least one integer occurring an odd number of times in S_i, and so ΦS_i also contains at least one integer occurring an odd number of times. The above arguments also hold for any proper subsequence u of $\bar{S}(n)$, when applying the remark right after the proof of Theorem 5.2.10. Hence, we may conclude that $O(\Phi u) \neq \emptyset$ for all proper subsequences u of $\bar{S}(n)$. This observation proves the following theorem.

Theorem 5.2.11. *If $gcd(n,t) = 1$, Construction 5.5 will produce an (n,t)-sequence.*

Consider a balanced Gray code G of length n with transition count spectrum $TC_n = (TC_n(1), TC_n(2), ..., TC_n(n))$. Furthermore, consider a uniform counting sequence produced by applying Construction 5.5 for $m = 1$, with transition count spectrum $TC_{(n,t)} = (TC_{(n,t)}(1), TC_{(n,t)}(2), ..., TC_{(n,t)}(n))$. Because of the mapping $\Phi_{1|t}$, we have that

$$TC_{(n,t)}(i) = TC_n(i) + TC_n(i^{(-1)}) + TC_n(i^{(-2)}) + \cdots + TC_n(i^{(-(t-1))}). \qquad (5.12)$$

Since TC_n is the transition count spectrum of a balanced Gray code G, it only contains integers a and $b := a+2$; with $a = \lfloor \frac{2^n}{n} \rfloor$. Due to Lemma 5.2.7, we can permute these integers such that they constitute a regular distribution. This distribution can be considered as the transition count spectrum of a Gray code G' which is equivalent to G. Actually, G' can be obtained from G by a similar permutation of the columns of G. It now follows from (5.12) and Corollary 5.2.8 that the integers $TC_{(n,t)}(i)$, $1 \leq i \leq n$, differ by at most 2, and so we have the following main theorem of this section.

Theorem 5.2.12. *For every integer n and odd t, $1 \leq t \leq n-1$, with $gcd(n,t) = 1$, there exists a balanced (n,t)-sequence.*

5.2.2 Computer results

By computer calculations we checked that for certain n-values, Construction 5.5 produces (n,t)-sequences for all t, $t \leq n-1$, with $gcd(n,t) > 1$. For instance, we were able to construct $(6,3)$, $(9,3)$, $(12,3)$, $(15,3)$, $(20,3)$, $(10,5)$, $(15,5)$, $(20,5)$, $(14,7)$ and $(21,7)$-sequences. Moreover, by adjusting the distribution of the transition count spectra, we can construct *balanced* $(9,3)$, $(15,3)$, $(15,5)$, and $(21,7)$-sequences.
We firmly believe that Construction 5.5 will produce an (n,t)-sequence for every pair of n and t, $1 \leq t \leq n-1$.

Lemma 5.2.13. *For every positive integer k, one has*

$$2^{3^k} = 3^k \cdot q + (3^k - 1),$$

for some integer q.

Proof. We prove this Lemma by induction to k. It is obvious that the Lemma is true for $k = 1, 2, 3$. Assume now that the Lemma is true for $k \geq 3$. So, we have that $2^{3^k} = 3^k \cdot s + (3^k - 1)$, for some integer s. Then we have

$$\begin{aligned} 2^{3^{k+1}} &= 2^{3^k \cdot 3} \\ &= (3^k \cdot s + (3^k - 1))^3 \\ &= (3^k \cdot s)^3 + 3(3^k \cdot s)^2(3^k - 1) + 3(3^k \cdot s)(3^k - 1)^2 + (3^k - 1)^3 \\ &= (3^k \cdot s)^3 + 3(3^k \cdot s)^2(3^k - 1) + 3(3^k \cdot s)(3^k - 1)^2 + 3^{3k} - 3(3^{2k}) + \\ &\qquad (3 \cdot 3^k - 1). \end{aligned}$$

We see that the first four terms on the right hand side are divisible by 3^{k+1}, for all $k \geq 1$. Hence, we have $2^{3^{k+1}} = 3^{k+1}.q + (3^{k+1} - 1)$, for some q. Because of the principle of mathematical induction we have proved the Lemma now. □

Since a balanced even n-partition of 2^n exists for every n (see in particular eq. (3.7)), Lemma 5.2.13 implies that a transition count spectrum of a balanced Gray code of length 3^k, for some non-negative integer k, can be arranged as $(a, b, a, b, ..., a, b, a)$, with $b = a + 2$. Hence, if Construction 5.5 produces a $(3^k, t)$-sequence, then we have from Corollary 5.2.8, that the resulting sequence will be balanced when starting from a balanced Gray code with transition count spectrum ordered as $(a, b, a, b, ..., a, b, a)$.

Remark By using Construction 5.5 for the construction of uniform counting sequences, we can determine the transition count spectrum of the resulting counting sequence by using the transition count spectrum of the Gray code which we use as basis of the construction. Since for any distribution $(p_1, p_2, ..., p_n)$, with $\sum_{i=1}^{n} p_i = 2^n$, p_i is even for every $i \in [n]$, and $|p_i - p_j| \leq 2$, $i, j \in [n]$, a balanced Gray code exists with transition count spectrum $(p_1, p_2, ..., p_n)$, we expect to be able to produce "approximately balanced" or balanced uniform counting sequences for every n and odd $t < n$.

PART TWO

LINEAR q-ARY LEXICODES

6

On the Construction of Linear q-ary Lexicodes

Let V be a list of all vectors of $GF(q)^n$, lexicographically ordered with respect to some basis. Algorithms which search list V from top to bottom, any time selecting a codeword which satisfies some criterion, are called greedy algorithms and the resulting set of codewords is called a lexicode or greedy code. In this thesis we stick to the term lexicode. If $q = 2$, then such a lexicode turns out to be linear, for many selection criteria. In this chapter we present a greedy algorithm for the construction of a large class of linear q-ary lexicodes which generalizes the algorithms of several other papers and puts these into a wider framework. By applying this method, one can produce linear lexicodes which cannot be constructed by previous algorithms, because the characteristics or the underlying field of the codes do not meet the conditions of those algorithms.

6.1 Introduction

Let $\mathbf{e}_i = e_n e_{n-1} \dots e_1$, $1 \leq i \leq n$, be the i-th unity standard vector in a vector space $GF(q)^n$, where q is a power of a prime number, i.e. e_i is equal to one and e_j is zero for all $j \neq i$. Originally [12, 13, 37], binary lexicodes were defined in the following way. Let $\mathbf{B} = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n)$ be the ordered standard basis of the vector space $V := GF(2)^n$. With respect to this ordered basis one defines lexicographically ordered lists $V_i = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{2^i}$ recursively by

$$V_0 = \mathbf{0}, \qquad V_i = V_{i-1}, \mathbf{e}_i + V_{i-1}, \;\; 1 \leq i \leq n.$$

According to this definition the list V_i consists of the list V_{i-1}, followed by the list obtained by adding $\mathbf{e}_i$ to all vectors of V_{i-1} in the same order in which these vectors

occur in V_{i-1}.
In the resulting *lexicographically* ordered list $V(= V_n)$ all binary n-tuples occur in normal lexicographic order, i.e. arranged according to increasing values of these tuples when interpreted as binary numbers. With respect to this order the inequality $\mathbf{x} < \mathbf{y}$ means that $\mathbf{x}$ occurs before $\mathbf{y}$ in the list V_n. Conway in [12], Conway and Sloane in [13] and Levenstein in [37] developed a greedy algorithm to produce binary codes with minimum distance d, which can be formulated in the following recursive way:

Choose the next vector $\mathbf{x}$ *of the list* V *whose distance vector* $\mathbf{x}+\mathbf{y}$ *with respect to each previously chosen vector* $\mathbf{y}$ *satisfies the property that its Hamming weight* $||\mathbf{x}+\mathbf{y}||$ *is at least equal to* d.

The resulting *lexicode* has the surprising property of being *linear*, and hence can be characterized as an $[n, k, d]$-code if the dimension of the lexicode is equal to $k \geq 1$. This result has been generalized in various ways. In [13] Conway and Sloane extended their own theory to codes over fields $GF(2^{2^l})$, $l \in \mathbf{N}$, (sometimes called Fermat fields [83]), with a proof for linearity embedded in the context of the theory of impartial games. Moreover, they also proved linearity when a more general type of selection criterion is used in the greedy algorithm, i.e. a criterion imposed by so-called *turning sets*. In [7] Brualdi and Pless discussed another generalization of binary lexicodes. Their starting point is again a list of all binary vectors of length n, but now ordered lexicographically with respect to an arbitrary ordered basis (called **B**-*ordering*) instead of the standard basis. The resulting codes, with minimum distance *at least* d, also turn out to be linear. The proof is accomplished by showing that there exists a homomorphism defined on V and mapping into the set of non-negative integers, such that its kernel is the constructed lexicode. So, this homomorphism defines the *parity check matrix* of the code as well.
Of particular interest in the paper of Brualdi and Pless is the application of *triangular bases*. Using a computer program for carrying out their greedy algorithm, they found, at least in the cases they considered, that lexicodes based on this type of basis have a dimension either equal to or one less than the dimension of the best codes known. In particular, a *Gray basis* of the vector space V gave almost always rise to optimal codes (with respect to the dimension) for the cases the algorithm was applied to in [7].
Fon-Der-Flaass [19] proved, applying the method of Brualdi and Pless, that binary lexicodes are linear for any family of turning sets as selection criterion, and for any ordered basis, thus generalizing both the results in [7] and [13]. Monroe in [49] applied and generalized the parity check approach of [7] to construct quaternary codes, which in several cases turned out to be larger than the codes known thus far for the same parameter sets.
Van Zanten, in [83, 85], further generalized all previous results for a wide class of selection properties P. More precisely, $P : V \longmapsto \{true, false\}$ is a Boolean-valued function depending on one variable. The only requirement such a criterion $P[.]$ has to satisfy is that it is fixed and is not changed dynamically during the course of the

algorithm, and that all vectors of V can be tested as to whether they satisfy P or not. A vector $\mathbf{x} \in V$ is selected if and only if $P[\mathbf{x}+\mathbf{y}]$ is true for all previously chosen vectors $\mathbf{y}$. It was proven in [83, 85] that any binary lexicode is linear for any such selection criterion and for any ordered basis. It was also proven that if P is a so-called *multiplicative property* on V, i.e. if "$P[\mathbf{x}]$ is true" implies "$P[\alpha\mathbf{x}]$ is true for all $\alpha \neq 0$ of the underlying field", then the lexicode constructed by the greedy algorithm is linear for any such selection criterion P, and for any binary basis of $GF(2^{2^l})^n$. All proofs in [83, 85] are of a purely (linear) algebraic nature. Essential in these proofs is that the elements of the field $GF(2^{2^l})^n$ are ordered in a special (*canonical*) way (cf. [83]).

Trachtenberg in [75] generalized the original notion of lexicodes having a certain minimal distance in two different ways. In the first place he introduced an algorithm which is initialized with a linear $[n, k, d]$-code C (*seed code*) which replaces the trivial code $\mathbf{0}$ in the usual algorithms. Next, the algorithm adds in greedy way the lexicographically earliest vector whose distance is at least d with respect to the span of C and the previously added vectors. This type of construction could be incorporated in the theory of $\mathbf{B}$-ordering of $GF(2)^n$ in [4, 7, 83], by taking an ordered basis $\mathbf{B}$ of $GF(2)^n$ the first k vectors of which span C. A second generalization in [75] is the replacement of the heuristic "lexicographically earliest" by some other generating function, giving rise to other families of codes.

Bonn in [4] also generalizes the algorithms as presented in [7] and [13]. He searches a list of all vectors over $GF(q)$ of length n which even need not to be ordered in some specific way. As soon as a vector $\mathbf{a}$ is found satisfying $d(\mathbf{a}, \mathbf{y}) \geq d$ for all previously found vectors $\mathbf{y}$, this $\mathbf{a}$ is added to the lexicode as well as all its multiples and distance vectors $\mathbf{a} - \mathbf{y}$ with respect to all previously found $\mathbf{y}$. Here, $d(\mathbf{a}, \mathbf{y})$ stands for the Hamming distance in $GF(q)^n$. Obviously, the resulting lexicode is forced to be linear for all finite fields $GF(q)$ with a basis constituted by the selected vectors $\mathbf{a}$, and it can be proven to have a minimum distance at least d.
In this chapter we introduce the following generalization of Bonn's algorithm. After having started by selecting the zero vector, a lexicographically ordered list of all vectors over $GF(q)$ of length n is searched from top to bottom. As soon as a vector $\mathbf{a}$ is found such that $\mathbf{a} + \mathbf{y}$ satisfies some criterion P for all previously found vectors $\mathbf{y}$, this $\mathbf{a}$ is added to the lexicode as a new basis vector, together with all vectors of the list which are generated by this new basis vector and the vectors already in the code. Only then the process of searching the list is continued. Like in [4] it is obvious again that the resulting lexicode C is linear. Furthermore, it can be proven easily (cf. Section 6.2), that if the selection criterion P is *multiplicative* (cf. the definition given before in this Introduction), P not only holds for the vectors $\mathbf{a} + \mathbf{y}$, but for *all* vectors $\mathbf{c} \in C$. So, we can say that the lexicode C possesses property P. The class of multiplicative selection criteria is rather large, as will be demonstrated by a number of examples in Sections 6.3, 6.4, and 6.5. One special example is the criterion $P[\mathbf{x}]$ is true if and only if the Hamming weight of $\mathbf{x}$, denoted by $||\mathbf{x}||$ in this chapter, is at least d. It will be clear that the resulting q-ary lexicodes are precisely the codes dealt

with in [4]. So, in this sense the algorithm of this chapter is a generalization of Bonn's algorithm. Furthermore, it will be shown in Section 6.2 that lexicodes constructed by applying our algorithm in binary cases are identical to the codes produced by the algorithm developed in [83], when using the same parameter values. As a consequence, we have as our main theorem, that lexicodes arising from the new algorithm are linear for any ordered basis, for any finite field and for any multiplicative property P, thus generalizing all previous results.

In Section 6.3, a number of special cases are briefly discussed. Some of these special cases were also mentioned in [83, 85], but only for the binary field. In Section 6.4, a number of examples are presented for the ternary field, with various choices for the selection criterion P. One of these examples shows that our algorithm, when applied to $GF(3)^8$ with selection criterion "$P[\mathbf{x}]$ is true if and only if $||\mathbf{x}|| \geq 3$," produces a linear ternary $[8, 5, 3]_3$-code, contrary to the algorithm in [13], which delivers a non-linear code in this case consisting of 198 codewords (cf. [13, p. 341]). Another special case, which was not dealt with in previous publications, is obtained by defining "$P[\mathbf{x}]$ is true if and only if $\mathbf{x}.\mathbf{x} = 0$". Here, $\mathbf{x}.\mathbf{x}$ stands for the scalar product of vector $\mathbf{x}$ with itself. The algorithm then produces *self-orthogonal*, and occasionally self-dual codes, over fields of characteristic *unequal* to 2. This case is discussed in Section 6.5.

Finally, we emphasize once more that unlike in the original search algorithms [7, 13, 12, 19, 37, 49, 83, 85], the property of being linear does not come as a surprise anymore, due to construction rule 3 of our new Algorithm 6.1 (cf. Section 6.2), just as in [4]. The surprising part of this linearity now seems to be that the rule of taking all linear combinations as soon as a new basis vector is found, can be left out from our algorithm for fields of type $GF(2^{2^l})$, $l \geq 0$, at the expense of testing now, for *all* $\mathbf{x} \in V$, the condition $P[\mathbf{x} + \mathbf{y}]$ for all previously found $\mathbf{y}$. In this sense the original algorithms are modified versions of Algorithm 6.1. At the end of Section 6.3, right after Corollary 6.3.2, it will be made clear that these modified algorithms produce the same lexicodes -in the relevant cases -as Algorithm 6.1 itself, be it that the number of tests these algorithms have to carry out is much larger.
A second remark, already made by Bonn in [4], is that a lexicographic ordering of the vectors of $V(= GF(q)^n)$ is not really necessary in order to construct linear codes. Any complete list of the q^n vectors of V will give rise to linear codes when submitted to the algorithm of Section 6.2, contrary to the algorithms in [7, 13, 12, 19, 37, 83, 85] where the lexicographic order is essential (cf. [83]). Nevertheless, our algorithm is based on a lexicographically ordered list V, since it simplifies the execution of rule 3 (cf. Section 6.2). Due to this order one can skip a major part of the sublist V_i, every time a new basis vector $\mathbf{a}_i$ is found, in the process of searching for a next basis vector of the code to be constructed.
Summarizing this Introduction, one could say that the new algorithm, which combines the features of the algorithms in [4] and [83], makes it possible to produce new types of linear lexicodes, and is of help to put several greedy algorithms into a general context. As for our notation, we shall not make any distinction between vectors and words. Therefore, there are no commas between successive components of a vector,

and neither do we put brackets at the front and at the back.

6.2 Construction of q-ary linear lexicodes

Let q be a power of a prime number p and let $GF(q)$ be the finite field with q elements. We shall identify, in some way or another, the elements of $GF(q)$ with those of the set $\{0, 1, \ldots, q-1\}$, and assume that $0 < 1 < \ldots < q-1$. For $q = p$ we shall apply the usual addition and multiplication rules mod p. For $q = p^r$, $r > 1$, we do not specify these rules at the moment. Now, let $V := GF(q)^n$, and consider V as a vector space over the field $GF(q)$, spanned by some basis $\mathbf{B}$. If we order the n basis vectors in some way we speak of the *ordered basis* $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$. With respect to this ordered basis we construct lexicographically ordered lists $V_i = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{q^i}$ in the following recursive way

$$\begin{aligned} V_0 &:= \mathbf{0}, \\ V_i &:= V_{i-1}, \mathbf{b}_i + V_{i-1},\ 2\mathbf{b}_i + V_{i-1}, \ldots, (q-1)\mathbf{b}_i + V_{i-1},\ 1 \le i \le n, \end{aligned} \tag{6.1}$$

(cf. the definition of the lexicographically ordered list of $GF(2)^n$ in Section 6.1). Apart from the order of the vectors, V is identical with V_n. From now on, we shall not distinguish between the n-dimensional vector space V and the *lexicographically ordered* list V_n of its vectors.

Assume that P denotes some property or criterion which is used to test whether or not a vector in V is selected. With respect to the chosen basis $\mathbf{B}$, and also with respect to this property P, we shall formulate a recursive greedy algorithm for the construction of a linear code. If some vector $\mathbf{x}$ satisfies property P, we shall express this by writing $P[\mathbf{x}]$ is true, or briefly by $P[\mathbf{x}]$. In this chapter we assume that P is a so-called *multiplicative property*, i.e. $P[\mathbf{x}]$ implies $P[\alpha\mathbf{x}]$ for all non-zero elements $\alpha \in GF(q)$ (cf. [83]). Many relevant properties like $||\mathbf{x}|| \ge d$, or $||\mathbf{x}||$ is even in the binary case, or more generally $||\mathbf{x}||$ belongs to some prescribed weight spectrum, or $\mathbf{x}$ satisfies some set of turning rules, belong to this class (cf. Section 6.3).
We now formulate our greedy algorithm to produce ordered lists of codewords C_i, $0 \le i \le n$.

Algorithm 6.1

1. $C_0 := \mathbf{0}$; $i := 1$;
2. select the first vector $\mathbf{a}_i$ in $V_i \backslash V_{i-1}$ such that $P[\mathbf{a}_i + \mathbf{c}]$ for all $\mathbf{c}$ in C_{i-1};
3. if such an $\mathbf{a}_i$ exists, then $C_i := C_{i-1}, \mathbf{a}_i + C_{i-1}, 2\mathbf{a}_i + C_{i-1}, \ldots, (q-1)\mathbf{a}_i + C_{i-1}$, otherwise $C_i := C_{i-1}$;
4. $i := i + 1$; return to 2.

It is obvious that the code C_i, $0 < i \leq n$, is identical with the set of all linear combinations of $l(\leq i)$ selected vectors $\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \ldots, \mathbf{a}_{i_l}$. So, the linearity of the lexicode $C := C_n$ is trivially satisfied. More precisely, $\mathcal{B} = (\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \ldots, \mathbf{a}_{i_l})$ constitutes a basis for C_i. This code C_i is not necessarily different from C_{i-1}, since not every sublist V_i will yield a vector $\mathbf{a}_i$ satisfying the property as formulated in rule 2. A question which might come up is the following: Could it be that there exists a vector $\mathbf{x} \in V_i \backslash V_{i-1}$, with $P[\mathbf{x} + \mathbf{c}]$ for all $\mathbf{c} \in C_i$, and $\mathbf{x} \notin C_i$? Or, in other words, would it be possible that the algorithm "misses" a codeword $\mathbf{x}$ when skipping the remaining part of V_i after execution of rule 3? The following theorem guarantees that such a vector $\mathbf{x}$ does not exist.

Theorem 6.2.1. *Let $\mathbf{a}_i \in V_i$ be such that $P[\mathbf{a}_i + \mathbf{c}]$ for all $\mathbf{c} \in C_{i-1}$, for $i \geq 1$. Then, every $\mathbf{x} \in V_i \backslash V_{i-1}$ satisfying $P[\mathbf{x} + \mathbf{c}]$ for all $\mathbf{c} \in C_i$, is in C_i.*

Proof. We proceed by induction to i. We only consider i-values for which a basis vector $\mathbf{a}_i$ exists.
(i) Let $j > 0$ be the first index such that $P[\mathbf{a}_j]$, i.e. $\mathbf{a}_j (= \mathbf{a}_{i_1})$ is the first vector selected by Algorithm 6.1, and hence $C_0 = C_1 = \cdots = C_{j-1} = \mathbf{0}, C_j = \mathbf{0}, \mathbf{a}_j, 2\mathbf{a}_j, \ldots, (q-1)\mathbf{a}_j$. Let $\mathbf{x} \in V_j \backslash V_{j-1}$ be a vector such that $P[\mathbf{x} + \alpha \mathbf{a}_j]$ for all $\alpha \in GF(q)$. Since $\mathbf{x} \in V_j \backslash V_{j-1}$, we can write $\mathbf{x} = \beta \mathbf{a}_j + \mathbf{v}$, for some $\beta \neq 0$ and for some $\mathbf{v} \in V_{j-1}$. If $\mathbf{v} = \mathbf{0}$, it follows immediately that $\mathbf{x} \in C_j$. If $\mathbf{v} \neq \mathbf{0}$, we take $\alpha = -\beta$ which yields $P[\mathbf{v}]$. However, this contradicts the assumption about j.
(ii) Let $\mathbf{a}_i \in V_i$, $i > j$, be a selected vector such that $P[\mathbf{a}_i + \mathbf{c}]$ for all $\mathbf{c} \in C_{i-1}$. Assume that the Theorem holds for all relevant index values less than i. Let $\mathbf{x} \in V_i \backslash V_{i-1}$ satisfy $P[\mathbf{x} + \mathbf{c}]$ for all $\mathbf{c} \in C_i$. Since $\mathbf{x} \in V_i \backslash V_{i-1}$, we can write $\mathbf{x} = \beta \mathbf{a}_i + \mathbf{v}$, for some $\beta \neq 0$ and some $\mathbf{v} \in V_{i-1}$. If we take $\mathbf{c} = -\beta \mathbf{a}_i + \mathbf{c}'$, it follows that $P[\mathbf{v} + \mathbf{c}']$, for all $\mathbf{c}' \in C_{i-1}$. Because of the induction assumption we may conclude that $\mathbf{v} \in C_{i-1}$, and hence $\mathbf{x} \in C_i$. □

Theorem 6.2.1 justifies that as soon as we have found a vector $\mathbf{a}_i \in V_i$ satisfying rule 2 of Algorithm 6.1, and after having extended the list of codewords by rule 3, we can continue the selection procedure by searching the sublist $V_{i+1} \backslash V_i$. So, by applying the algorithm as formulated in the beginning of this Section, we produce a nested sequence of ordered codes

$$\mathbf{0} = C_0 \subseteq C_1 \subseteq \cdots \subseteq C_n = C. \tag{6.2}$$

For each i, $0 < i \leq n$, we have that either $dim(C_i) = dim(C_{i-1})$ or $dim(C_i) = dim(C_{i-1}) + 1$ (cf. also [83]). Since the lexicode C depends on the basis $\mathbf{B}$ and on the property P, we denote this code occasionally by $C(\mathbf{B}, P)$.
Let k be the dimension of C. In the remaining part of this chapter we shall denote the k basis vectors of C by $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ instead of $\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \ldots, \mathbf{a}_{i_k}$ for reasons of convenience. The indices of the corresponding lists of codewords will also be redefined, and so from now on C_i is the lexicographically ordered list of codewords defined by

$(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_i)$ for $1 \leq i \leq k$.
From the construction of C we have for all vectors of the form $\mathbf{a}_i + \mathbf{c}$, $1 \leq i \leq k$, $\mathbf{c} \in C_{i-1}$, that $P[\mathbf{a}_i + \mathbf{c}]$. If we take a multiplicative property P as selection criterion, it can be proved very easily that P holds for *all* non-zero vectors of the code. Thus, we arrive at our main theorem.

Theorem 6.2.2. *For any basis* $\mathbf{B}$ *of* $GF(q)^n$ *and for any multiplicative selection criterion* P*, the lexicode* $C(\mathbf{B}, P)$ *is linear, and* $P[\mathbf{x}]$ *holds for each codeword* $\mathbf{x} \neq \mathbf{0}$*.*

Proof. The linearity of the code follows trivially from its construction, as was already remarked in Section 6.1. Since $P[\mathbf{a}_i + \mathbf{c}]$, for all $\mathbf{c} \in C_{i-1}$, we also have $P[\alpha\mathbf{a}_i + \alpha\mathbf{c}]$ for all $\alpha \in GF(q)$, $\alpha \neq 0$, and for all $\mathbf{c} \in C_{i-1}$, due to the assumption that P is a multiplicative property. Equivalently, $P[\alpha\mathbf{a}_i + \mathbf{c}]$ for all $\mathbf{c} \in C_{i-1}$, since C_{i-1} is linear. Applying this result for $i = 1, 2, \ldots, k$, respectively, yields that $P[\mathbf{x}]$ is true for any codeword $\mathbf{x} \neq 0$, since the vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ constitute a basis for $C(\mathbf{B}, P)$. □

6.3 Special cases

In this section we deal with some special cases of Theorem 6.2.2, and the resulting conclusions will be presented as corollaries to that theorem. All of these constitute generalizations of the results presented in [83] for the binary case. As usual, we define the Hamming weight $||\mathbf{x}||$, or just the weight, of a vector $\mathbf{x} \in GF(q)^n$ as the number of components x_i which are unequal to 0. We start by introducing a set of Hamming weights

$$S = \{d_0, d_1, d_2, \ldots\}, \tag{6.3}$$

with $d_0 < d_1 < d_2 < \ldots \leq n$, which we call a *designed weight spectrum* (cf. [83]). Here, the selection criterion we use is that the weight of the vector $\mathbf{a}_i + \mathbf{c}$ is in S for all $\mathbf{c} \in C_{i-1}$. The resulting lexicode is denoted by $C(\mathbf{B}; S)$. More generally, let us consider *partitioned codewords*. Let

$$n = n_1 + n_2 + \cdots + n_l, \tag{6.4}$$

with $n_j \in \mathbf{Z}^+$, $1 \leq j \leq l$, be a certain partition of n[1]. We define the partitioned codeword $\mathbf{x} \in V$ with respect to the partition n as

$$\mathbf{x}^{(n)} \equiv \mathbf{x}^{(n_1, n_2, \ldots, n_l)} := \mathbf{x}^1 \mathbf{x}^2 \ldots \mathbf{x}^l, \tag{6.5}$$

where each part $\mathbf{x}^j$ consists of n_j consecutive components of $\mathbf{x}$. Furthermore, we introduce designed weight spectra

$$S^j = \{d_0^j, d_1^j, d_2^j, \ldots\}, \tag{6.6}$$

[1]For conventional reasons we use the word "partition", though "composition" would be a better term.

with $(d_0 =)d_0^j < d_1^j < d_2^j \ldots \leq n_j$, for $1 \leq j \leq l$. Applying our greedy algorithm, the vector $\mathbf{a}_i$ will be chosen if and only if the distance vector $\mathbf{a}_i^j + \mathbf{c}^j \in S^j$, for $1 \leq j \leq l$, and for all vectors $\mathbf{c}$ in C_{i-1}. We denote the resulting lexicode by $C(\mathbf{B}; (n), (S))$, where (n) stands for the partition (6.4) and (S) for the ordered sequence of weight spectra (6.6). We shall say that $||\mathbf{a}_i + \mathbf{c}||$ is in (S). The following corollary generalizes Corollary 6.3.1 in [83], where it is stated for the binary case.

Corollary 6.3.1. *The lexicode $C(\mathbf{B}; (n), (S))$ is linear with respect to an arbitrary basis $\mathbf{B}$, for any partition (n), and for any choice of the designed weight spectra $(S) = (S^1, S^2, \ldots, S^j)$.*

The proof is immediate by observing that the selection criterion P is multiplicative in this case.

If we take the trivial partition, i.e. $n_1 = n$, and if we take the weight spectrum $S = \{d | d \geq d_0\}$, we obtain Bonn's result in [4].

Another consequence of Theorem 6.2.2 which is related to Corollary 6.3.1, is obtained by considering the notion of *turning sets* as defined in [13]. A turning set is a set of indices of the vectors $\mathbf{x} \in GF(q)^n$. Let Γ be a family of turning sets. As in [13], the selection criterion P is that the set of indices of the non-zero components of the vector $\mathbf{a} + \mathbf{c}$ is not a member of Γ. Since this criterion is also multiplicative we have immediately the following result which generalizes similar results for the binary case dealt with in [13, 19].

Corollary 6.3.2. *The lexicode $C(\mathbf{B}; \Gamma)$ is linear with respect to any basis $\mathbf{B}$ and for any family Γ of turning sets, and each codeword $\mathbf{x}$ meets the rules prescribed by Γ.*

We remark that for those cases when the greedy algorithms of [7, 13, 19, 83] produce linear lexicodes, i.e. for binary vector spaces or, more generally, for vector spaces over $GF(2^{2^l})$, it can easily be proved that these lexicodes are identical to the lexicodes produced by Algorithm 6.1. The proof is almost immediate, if one realizes that both types of lexicodes are spanned by the same basis vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$. For details we refer to [78].

Example 6.3.1. A special case of Corollary 6.3.1 is obtained by taking a weight spectrum $S = \{d | d \leq m\}$, i.e. by requiring an upper bound for the distance between codewords. In the binary case, codes with codeword length n, maximum distance m and dimension k, are called (linear) *anticodes*, and are used for the construction of "normal" linear codes having a minimum distance (cf. [17, 43]). More precisely, if we denote such an anticode by $[n, k, m]_a$, the construction described in [17, 43] gives rise to a linear $[2^k - 1 - n, k, 2^{k-1} - m]$-code.
We can apply Algorithm 6.1 to produce binary linear anticodes. We take the following

basis for $GF(2)^{12}$.

$$\begin{aligned} b_1 &= 011100110000 \\ b_2 &= 000000000100 \\ b_3 &= 111111000001 \\ b_4 &= 111110101010 \\ b_5 &= 110100111100 \\ b_6 &= 001001101010 \\ b_7 &= 101111111101 \\ b_8 &= 111110101000 \\ b_9 &= 111010010110 \\ b_{10} &= 011101010111 \\ b_{11} &= 111110110111 \\ b_{12} &= 101111101011. \end{aligned}$$

By a computer program we find the basis

$$\begin{aligned} a_1 &= 011100110000 \\ a_2 &= 000000000100 \\ a_3 &= 111111000001 \\ a_4 &= 110100111100 \\ a_5 &= 100110010111 \\ a_6 &= 010100111101 \\ a_7 &= 100110011110. \end{aligned}$$

spanning a linear anticode of dimension 7. By the construction mentioned above, this anticode gives rise to a $[115, 7, 56]$-code. According to the table of Brouwer in [53, p. 321], this is an optimal code in the sense that it has a maximal d-value with respect to $[115, 7, d]$-codes.

6.4 Examples of linear ternary lexicodes

In this section we present a number of examples of linear ternary lexicodes produced by Algorithm 6.1, which demonstrate that Theorem 6.2.2 and its corollaries give rise to results that cannot be obtained by previous algorithms as referred to in the Introduction.

Example 6.4.1. Conway and Sloane in [13, p. 341] present an example of a ternary lexicode of codeword length 8 and minimum distance 3, obtained by applying their algorithm to $GF(3)^8$ listed in lexicographic order with respect to the standard basis. The resulting lexicode appears to consist of 198 codewords, and hence is clearly not linear. Applying Algorithm 6.1 yields a lexicode with 243 codewords (cf. [78]) which is linear by Corollary 6.3.1. The basis vectors of this code, as produced by the

algorithm, are

$$\begin{aligned} a_1 &= 00000111 \\ a_2 &= 00001012 \\ a_3 &= 00110001 \\ a_4 &= 01010002 \\ a_5 &= 10010010. \end{aligned}$$

Example 6.4.2. Now we list the words of $GF(3)^8$ lexicographically with respect to the ordered triangular basis

$$\begin{aligned} b_1 &= 00000002 \\ b_2 &= 00000020 \\ b_3 &= 00000211 \\ b_4 &= 00001210 \\ b_5 &= 00011011 \\ b_6 &= 00212221 \\ b_7 &= 02102211 \\ b_8 &= 21011112. \end{aligned}$$

As selection criterion we take "$P[\mathbf{x}]$ is true if and only if $||\mathbf{x}||$ is in the spectrum $S = \{3, 6\}$". Applying Algorithm 6.1 yields a linear ternary code (cf. Corollary 6.3.1) of dimension 4, with basis vectors

$$\begin{aligned} a_1 &= 00000211 \\ a_2 &= 00001202 \\ a_3 &= 02110211 \\ a_4 &= 21011110. \end{aligned}$$

Example 6.4.3. Again we apply Algorithm 6.1 of Section 6.2 to $GF(3)^8$, now listed in standard lexicographic order (cf. Example 6.4.1). Let $\mathbf{x} = x_8x_7x_6x_5x_4x_3x_2x_1$ be a vector in $GF(3)^8$. This time we take as selection criterion "$P[\mathbf{x}]$ is true if and only if $||\mathbf{x}|| \geq 3$ and $wt(\mathbf{x}) = 0$", where $wt(\mathbf{x}) = \sum_{i=1}^{8} x_i \bmod 3$. It will be clear that this property P is multiplicative on $GF(3)^8$, and hence the resulting lexicode will be linear, and all its codewords will satisfy P according to Theorem 6.2.2. In fact, we obtain a code of dimension 5 generated by the basis vectors of the following triangular basis

$$\begin{aligned} a_1 &= 00000111 \\ a_2 &= 00011001 \\ a_3 &= 00101010 \\ a_4 &= 01001022 \\ a_5 &= 10002012. \end{aligned}$$

If we use the same criterion $P[\mathbf{x}]$, but now applied to $GF(3)^8$ lexicographically

ordered with respect to the basis

$$\begin{aligned} b_1 &= 01010112 \\ b_2 &= 00200022 \\ b_3 &= 00200020 \\ b_4 &= 00002211 \\ b_5 &= 00020022 \\ b_6 &= 22202202 \\ b_7 &= 01010102 \\ b_8 &= 00000212, \end{aligned}$$

we obtain a code of dimension 4, spanned by the basis vectors

$$\begin{aligned} a_1 &= 01010112 \\ a_2 &= 00200202 \\ a_3 &= 01212000 \\ a_4 &= 22102101. \end{aligned}$$

These two examples demonstrate that the dimension may increase when taking a triangular basis, like in the binary case [7]. We remark that the selection criterion in this example, i.e. the requirement $wt(\mathbf{x}) = 0$, *cannot* be expressed in terms of turning sets (cf. [13] and the end of Section 6.3).

Example 6.4.4. In this example we order the words of $GF(3)^8$ lexicographically with respect to the Gray basis (cf. [7])

$$\begin{aligned} b_1 &= 00000001 \\ b_2 &= 00000011 \\ b_3 &= 00000110 \\ b_4 &= 00001100 \\ b_5 &= 00011000 \\ b_6 &= 00110000 \\ b_7 &= 01100000 \\ b_8 &= 11000000. \end{aligned}$$

The words of $GF(3)^8$ are partitioned according to $\mathbf{x} = \mathbf{x}^1\mathbf{x}^2$, where both $\mathbf{x}^1$ and $\mathbf{x}^2$ have length 4. As selection criterion P we take "$P[\mathbf{x}]$ is true if and only if $||\mathbf{x}^1|| \geq 1$ and $||\mathbf{x}^2|| \geq 2$". Algorithm 6.1 now produces a linear lexicode of dimension 3 spanned by the basis vectors

$$\begin{aligned} a_1 &= 00011001 \\ a_2 &= 00110011 \\ a_3 &= 01100110. \end{aligned}$$

Since P is multiplicative, all non-zero codewords satisfy the required property by Corollary 6.3.1.

Example 6.4.5. Here, we order $GF(3)^8$ lexicographically with respect to the basis

$$\begin{aligned} b_1 &= 00000001 \\ b_2 &= 00000011 \\ b_3 &= 00000111 \\ b_4 &= 00001110 \\ b_5 &= 00011100 \\ b_6 &= 00111000 \\ b_7 &= 01110000 \\ b_8 &= 11100000. \end{aligned}$$

Furthermore, we partition the words of $GF(3)^8$ as described in Example 6.4.4. Applying Algorithm 6.1 with selection criterion "$P[\mathbf{x}]$ is true if and only if $||\mathbf{x}^1|| \geq 2$ and $wt(\mathbf{x}^2) = 0$" provides us with a linear lexicode of dimension 3 generated by the basis vectors

$$\begin{aligned} a_1 &= 00111002 \\ a_2 &= 01121101 \\ a_3 &= 11100000. \end{aligned}$$

Since P is multiplicative, it follows from Theorem 6.2.2 that all non-zero codewords of this code satisfy the above condition.

6.5 Self-orthogonal codes

Finally, we consider an application of Theorem 6.2.2, the binary version of which is discussed in [47], but which has no counterpart in [83, 85]. Let $\mathbf{x}.\mathbf{y} := \sum_{i=1}^{n} x_i y_i$ denote the dot product of vectors $\mathbf{x}, \mathbf{y} \in GF(q)^n$. Let C be a linear code over $GF(q)$ and let

$$C^{\perp} = \{\mathbf{x} \in GF(q)^n | \mathbf{x}.\mathbf{c} = 0 \text{ for all } \mathbf{c} \in C\}. \tag{6.7}$$

The linear code $C^{\perp}$ is called the *dual* or the *orthogonal* code of C. A code C is *self-orthogonal* if $C \subseteq C^{\perp}$, and *self-dual* if $C = C^{\perp}$. We shall discuss how to construct self-orthogonal lexicodes by applying Algorithm 6.1. The selection criterion P in this case is defined as "$P[\mathbf{x}]$ is true if and only if $\mathbf{x}.\mathbf{x} = 0$". This property P is clearly multiplicative. So, according to Theorem 6.2.2 we have for all vectors $\mathbf{z}$ in the lexicode C that $\mathbf{z}.\mathbf{z} = 0$. Hence, for all $\mathbf{x}, \mathbf{y} \in C$ we have $(\mathbf{x} + \mathbf{y}).(\mathbf{x} + \mathbf{y}) = 0$, $\mathbf{x}.\mathbf{x} = 0$ and $\mathbf{y}.\mathbf{y} = 0$. For vector spaces over a field of characteristic unequal to 2 these equalities imply $\mathbf{x}.\mathbf{y} = 0$. This gives rise to the following result, as another special case of Theorem 6.2.2.

Corollary 6.5.1. *Let V be a vector space over a finite field of characteristic unequal to 2, and let P be the criterion such that $P[\mathbf{x}]$ is true if and only if $\mathbf{x}.\mathbf{x} = 0$. Then the lexicode $C(\mathbf{B}, P)$ is a maximal self-orthogonal code for any basis $\mathbf{B}$.*

Proof. It was already pointed out that the resulting lexicode is self-orthogonal. As a consequence of Theorem 6.2.1 it is even maximal self-orthogonal, i.e. the code is not contained in a larger self-orthogonal code (cf. [44]). □

Remark 6.5.1.

(i) From [55, Theorem 1] it follows that the dimension of the lexicode $C(\mathbf{B}, P)$ of Corollary 6.5.1 is equal to $\frac{n}{2}$, and hence this code is self-dual, whenever n is even and $(-1)^{\frac{n}{2}}$ is a square in $GF(q)$. In all other cases the dimension of $C(\mathbf{B}, P)$ is equal to $\frac{n}{2} - 1$ for n even and to $\frac{(n-1)}{2}$ for n odd.

(ii) When applying rule 2 of Algorithm 6.1 in the case of Corollary 6.5.1, it suffices to test $P[\mathbf{x} + \mathbf{0}]$ and $P[\mathbf{x} + \mathbf{a}_j]$, $1 \leq j < i$, since these conditions imply $P[\mathbf{x} + \mathbf{c}]$ for *all* $\mathbf{c} \in C_{i-1}$.

(iii) Results similar to Corollary 6.5.1 would follow by applying a criterion P defined as $P[\mathbf{x}]$ is true if and only if $\mathbf{x}.\mathbf{x}^\star = 0$, where $\mathbf{x}^\star$ stands for some fixed, but otherwise arbitrary conjugate of $\mathbf{x}$ with respect to $GF(q)$.

Example 6.5.1. A maximal self-orthogonal lexicode is produced by Algorithm 6.1 with the selection criterion P as described in Corollary 6.5.1, with parameters $q = 3$, $n = 4$, and with respect to the ordered Gray basis in $GF(3)^4$, i.e. the basis consisting of the vectors 0001, 0011, 0110, 1100. The list of codewords is as follows.

0000	1102	2201
0111	1210	2012
0222	1021	2120.

Since n is a multiple of 4 (cf. Remark 6.5.1 (i)) this code is self-dual and is denoted by ξ_4 [44]. If we consider the case $q = 3$, $n = 7$, with respect to the ordered Gray basis with vectors 0000001, 0000011, 0000110, 0001100, 0011000, 0110000, 1100000, our algorithm produces a maximal self-orthogonal code with list

0000000	1111021	2222012
0000111	1111102	2222120
0000222	1111210	2222201
0001102	1112120	2220111
0001210	1112201	2220222
0001021	1112012	2220000
0002201	1110222	2221210
0002012	1110000	2221021
0002120	1110111	2221102

This code is not self-dual, e.g. because the word 1200000, which is orthogonal to the code, is not in the code. This also follows from the fact that the word length is not a multiple of 4, which is a necessary condition for a ternary code to be self-dual (cf. [44]) and from Remark 6.5.1 (i)). A criterion like "$P[\mathbf{x}]$ is true if and only if $\mathbf{x}.\mathbf{x} = 0$ and $||\mathbf{x}|| \geq d$" is also multiplicative. The resulting code however, although self-orthogonal, is not necessarily *maximal* self-orthogonal. Applying this P to the

case $q = 3$, $n = 8$ and $d = 6$, with respect to the standard basis, yields the following ternary self-orthogonal code C with minimum distance 6.

$$\begin{array}{c} 00000000 \\ 00111111 \\ 00222222 \\ 11001122 \\ 11112200 \\ 11220011 \\ 22002211 \\ 22110022 \\ 22221100 \end{array}$$

Obviously this code is not self-dual, since its dimension is less than $\frac{n}{2}(= 4)$. One could say that the code is a maximal self-orthogonal ternary code of length 8 and minimum distance 6 in the sense that it is not contained in a larger self-orthogonal code with the same parameter values according to Theorem 6.2.1. However, C is not maximal self-orthogonal in absolute sense, since $C' := span(C, 01200001)$ is a self-orthogonal code, with minimum distance 3, containing C.

We emphasize here that Corollary 6.5.1 is no longer valid for finite fields of characteristic equal to 2. This is shown by the following counterexamples.

Example 6.5.2. We list the words of $GF(2)^3$ lexicographically with respect to the standard basis, and we apply the selection property $P[\mathbf{x}]$ is true if and only if $\mathbf{x}.\mathbf{x} = 0$. The resulting lexicode is spanned by the basis vectors $\mathbf{a}_1 = 011$ and $\mathbf{a}_2 = 101$. It is clear that these two basis vectors are not orthogonal to each other, hence the resulting lexicode is not a self-orthogonal code.

Example 6.5.3. Let $GF(4)$ be the field with elements $0, 1, \omega$ and $\omega + 1$, where ω is defined by $\omega^2 + \omega + 1 = 0$. The canonical order (cf. [83]) is obtained by identifying $2 \equiv \omega$ and $3 \equiv \omega + 1$. We list the words of $GF(4)^3$ lexicographically with respect to the ordered triangular basis

$$\begin{array}{c} b_1 = 001 \\ b_2 = 032 \\ b_3 = 201 \end{array}$$

By applying the selection criterion P of Example 5.4, we obtain a lexicode with basis vectors $\mathbf{a}_1 = 033$ and $\mathbf{a}_2 = 202$. Again we have that $\mathbf{a}_1.\mathbf{a}_2 = 1 \neq 0$, hence the resulting lexicode is not a self-orthogonal code.

7

Self-Orthogonal Ternary Lexicodes

Some interesting properties of self-orthogonal ternary lexicodes are derived. We formulate a characterization for self-orthogonal ternary lexicodes which *cannot* be a subcode of a corresponding lexicode. We also derive self-orthogonal $[n, 3, d]$-codes, with $d = 3(3k), 3(3k+1)$ or $3(3k+2)$, $k \geq 1$, which can be used as seed codes when applying a self-orthogonal greedy algorithm.

7.1 Introduction

Let $\mathbf{x} = x_n x_{n-1} \cdots x_1$ be a vector of length n and let $\mathbf{x} \cdot \mathbf{x} = \sum_{i=1}^{n} x_i \cdot x_i$. The property $P[\mathbf{x}]$ if and only if $\mathbf{x} \cdot \mathbf{x} = 0$, is obviously multiplicative in V. Therefore the resulting lexicodes are linear by Theorem 6.2.2. As mentioned in Chapter 6, if the underlying field has characteristic *unequal* to 2, we obtain $\mathbf{x} \cdot \mathbf{y} = 0$ for all $\mathbf{x}, \mathbf{y} \in C$, when applying Algorithm 6.1. This means that C is a self-orthogonal code. In the sequel, if Algorithm 6.1 operates with the property $P[\mathbf{x}]$ if and only if $\mathbf{x} \cdot \mathbf{x} = 0$, and if the characteristic of the underlying field is unequal to 2, then the algorithm will be called an *orthogonal greedy algorithm*. The term self-orthogonal (self-dual) lexicode will mean that this lexicode is produced by using the orthogonal greedy algorithm.

Definition 7.1.1. *Basis* $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ *is called a triangular basis, if the basis vector* $\mathbf{b}_i$*, has a non-zero component in the* i*-th coordinate and zero components in all coordinates left from* i*, for* $1 \leq i \leq n$*. If in a triangular basis the vectors have only* 0 *and* 1 *as components, the basis will be called a* (0-1)*-triangular basis. The standard basis is an example of a* (0-1)*-triangular basis.*

From now on, we shall restrict ourselves to the ternary case, i.e. $q = 3$. We start the discussion with the following lemma.

Lemma 7.1.1. *For any triangular basis, the lexicode of length* 4 *produced by the orthogonal greedy algorithm is a self-dual code.*

Proof. The first vector selected by the orthogonal greedy algorithm is a vector which has the form $0x_3x_2x_1$, where x_3, x_2, x_1 are all non-zero. So, the lexicode C_3 is represented by the list

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & x_3 & x_2 & x_1 \\ 0 & 2x_3 & 2x_2 & 2x_1 \end{array}$$

The first selected vector will now be $y_4y_3y_2y_1$, $y_4 \neq 0$, which is orthogonal to C_3. Since this vector is also orthogonal to itself, due to the selection criterion P, the string $y_3y_2y_1$ must have weight 2. Now, according to the second part of Theorem 6.2.2, each of the next six vectors which are produced by the orthogonal greedy algorithm is also orthogonal to itself, and hence, all these vectors have weight three. More precisely, these six vectors all have a non-zero component on the fourth position from the right, and there is only one zero component among the first three components. This implies that there is no vector in $V \backslash C_4$ of weight $1, 2$, or 3, which is orthogonal to C_4.
Next, it can be proved that no vector of weight 4 is orthogonal to C_4 (cf. [78]). Thus we can conclude that $C_4 = C_4^{\perp}$, or equivalently, that C_4 is self-dual. □

The following theorem is a slight generalization of [78, Theorem 6.2.2].

Theorem 7.1.2. *For $n = 4k$, $k \in \mathbb{Z}^+$, and for any triangular basis, the lexicode produced by the orthogonal greedy algorithm is a self-dual $[n, \frac{n}{2}]$ code.*

Proof. The proof is based on Lemma 7.1.1 and is similar to the proof of Theorem 6.2.2 presented in [78]. □

Definition 7.1.2. *Let M be an nxn-matrix. Let $\mathbf{B}$ be a triangular ordered basis where the length of the basis vectors is a multiple of n. Consider $\mathbf{B}$ as a matrix the rows of which are these basis vectors in the same order. If the matrix $\mathbf{B}$ is equal to*

$$\begin{pmatrix} 0 & \cdots & 0 & 0 & M \\ 0 & \cdots & 0 & M & \star \\ 0 & \cdots & M & \star & \star \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M & \cdots & \star & \star & \star \end{pmatrix},$$

where "$\star$" is any nxn-matrix, we call the basis $\mathbf{B}$ (or equivalently, the matrix $\mathbf{B}$) M-self-similar with period n.

Example 7.1.1. Consider the ordered triangular basis $\mathbf{B}$ with basis vectors 0000001, 000011, 000101, 001001, 011101, 101101. The rows of the matrix

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

are equal to these basis vectors in the same order. It is clear that this matrix has two sub-matrices $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$ along its minor diagonal. So, the triangular basis $\mathbf{B}$ is $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$-self-similar with period 3.

Definition 7.1.3. *Let C_i, $1 \leq i \leq l$, be a linear code of dimension k_i, respectively. Let G_i be the generator matrix of C_i, $1 \leq i \leq l$, and let G be the matrix*

$$G = \begin{pmatrix} 0 & \cdots & 0 & 0 & G_1 \\ 0 & \cdots & 0 & G_2 & G_1' \\ 0 & \cdots & G_3 & G_2' & G_1' \\ \vdots & & \vdots & \vdots & \vdots \\ G_l & \cdots & G_3' & G_2' & G_1' \end{pmatrix},$$

where G_i' is a matrix the rows of which are codewords of C_i, $1 \leq i \leq l$. The code C which has generator matrix G is called the direct product $C_1 \otimes C_2 \otimes \cdots \otimes C_l$, and has dimension $k_1 + k_2 + \cdots + k_l$.

Now, we have the following corollary which was proved for the binary case in [47].

Corollary 7.1.3. *If the orthogonal greedy algorithm produces a self-dual $[n, \frac{n}{2}]$ lexicode C with respect to some triangular basis $\mathbf{B}$, then the lexicode of length mn produced by the orthogonal greedy algorithm with respect to some $\mathbf{B}$-self-similar triangular basis is a self-dual $[mn, \frac{mn}{2}]$ code which is equal to the direct product of m copies of C.*

The proof is immediate due to Theorem 7.1.2.

Example 7.1.2. The orthogonal greedy algorithm will produce a self-dual $[4, 2]$ code C, when it is applied to the ordered basis $\mathbf{B} = (0001, 0012, 0110, 1201)$. This code C is spanned by basis vectors 0111 and 1201. The ordered basis (00000001,00000012, 00000110, 00001201, 00011001, 00120111, 01101102, 12010021) is $\mathbf{B}$-self-similar. According to Corollary 7.1.3, the self-dual lexicode of length 8 produced by the orthogonal greedy algorithm with respect to this basis is a self-dual $[8, 4]$ code. The code is a direct product of 2 copies of the code C. Actually, this code is spanned by basis vectors 00000111, 00001201, 01112102 and 12010000.

7.2 Self-orthogonal ternary lexicodes with prescribed minimum distance

It will be clear that all weights in a ternary self-dual code are in the set of $\{3, 6, 9, \ldots\}$. In particular, self-orthogonal lexicodes have minimum distance $3l$, $l = 1, 2, \ldots$. Next, we shall discuss self-orthogonal lexicodes with prescribed minimum distance d. In this case we apply the property P defined as "$P[\mathbf{x}]$ is true if and only if $\mathbf{x} \cdot \mathbf{x} = 0$ and $||\mathbf{x}|| \geq d$". Here, the orthogonal greedy algorithm is extended with a minimum distance requirement. It is obvious that this property P is again multiplicative on V. For reasons of convenience we shall call Algorithm 6.1 equipped with this criterion, Algorithm 7.1.

Algorithm 7.1

1. Let $d \geq 3$ be fixed and let $C_0 := \mathbf{0}$; $i := 1$;
2. select the first vector $\mathbf{a}_i$ in $V_i \backslash V_{i-1}$ such that $(\mathbf{a}_i + \mathbf{c}) \cdot (\mathbf{a}_i + \mathbf{c}) = 0$ and that $||\mathbf{a}_i + \mathbf{c}|| \geq d$, for all $\mathbf{c}$ in C_{i-1};
3. if such an $\mathbf{a}_i$ exists, then $C_i := C_{i-1}, \mathbf{a}_i + C_{i-1}, 2\mathbf{a}_i + C_{i-1}, \ldots, (q-1)\mathbf{a}_i + C_{i-1}$, otherwise $C_i := C_{i-1}$;
4. $i := i + 1$; return to 2.

We have the following theorem.

Theorem 7.2.1. *Let $d = 3k - \varepsilon$, with $k \geq 1$ and $\varepsilon = 1$ or 2. Then a ternary self-orthogonal lexicode of minimum distance d is equal to the one of minimum distance $3k$.*

Proof. The proof follows from the fact that each codeword in a ternary self-orthogonal code has a weight equal to a multiple of 3. □

Example 7.2.1. If the orthogonal greedy algorithm is applied for parameter values $n = 12$, $d = 6$, and if $\mathbf{B}$ is equal to the standard basis, the resulting lexicode is generated by the generator matrix

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 2 & 1 & 2 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 2 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 2 & 1 & 1 & 0 \end{pmatrix}.$$

The lexicode generated by this matrix is a ternary self-dual extremal code (see [53, p. 270]), and is known to be unique (cf. [54, p. 168]). Now, according to Theorem 7.2.1, the same lexicode will be produced whenever the algorithm is applied for $d = 4$ or $d = 5$.

For some small relevant values of n, computer calculations show that if Algorithm 7.1 produces a self-dual code with respect to some triangular basis, then for an arbitrary triangular basis Algorithm 7.1 also produces a self-dual code of the same length and minimum distance. These observations lead us to the following conjecture.

Conjecture 7.2.2. *If for some relevant values of n and d, Algorithm 7.1 produces a self-dual code for some triangular basis, then it produces for any arbitrary triangular basis a self-dual code with the same parameter values for n and d.*

Theorem 7.2.3. *If Algorithm 7.1 produces a self-dual $[n, \frac{n}{2}, d]$ code for a triangular basis $\mathbf{B}$, then the lexicode of length mn produced by Algorithm 7.1 with respect to a $\mathbf{B}$-self-similar triangular basis is a self-dual $[mn, \frac{mn}{2}, d]$ code.*

Proof. Let us denote the original self-dual $[n, \frac{n}{2}, d]$ lexicode by C. It is necessary that n is a multiple of 4. We proceed by induction to m. For $m = 1$ the theorem is trivially true. Assume that the theorem is true for each $k, 1 \leq k \leq m-1$, for some fixed $m \geq 1$. So, we have a self-dual $[(m-1)n, \frac{(m-1)n}{2}, d]$ lexicode. Let us denote this code by $(m-1)C$. Since $(m-1)C$ is a self-dual code, any additional vector added to $(m-1)C$ by the algorithm is identical to some element of $(m-1)C$ in the first $(m-1)n$ positions from the right. Because of Corollary 7.1.3, the next $q^{\frac{n}{2}}$ codewords, when restricted to the leftmost n positions, will be self-dual vectors and will have minimum distance d. □

It follows that if $d = 6$ and if $\mathbf{B}$ is the standard basis, Algorithm 7.1 will produce self-dual lexicodes when the codeword length n is a multiple of 12. This result is due to Example 7.2.1 and Theorem 7.2.3.

We conclude that the following extremal self-dual lexicodes are produced by Algorithm 7.1 with prescribed minimum distance:

$$[4, 2, 3], [8, 4, 3], [12, 6, 6].$$

We assume that Algorithm 7.1 will produce extremal self-dual lexicodes for any relevant value of the minimum distance d. More precisely, we have the following conjecture.

Conjecture 7.2.4. *For every relevant value of d, and for any triangular basis, Algorithm 7.1 will produce extremal self-dual lexicodes with minimum distance d.*

In the next, Algorithm 6.1 equipped with the selection criterion $P[\mathbf{x}]$ if and only if $||\mathbf{x}|| \geq d$, for some $d \geq 1$, will shortly be called Algorithm 6.1. The lexicodes produced by Algorithm 6.1 are simply called lexicodes. Self-orthogonal lexicodes are the codes produced by the self-orthogonal greedy algorithm. Notice that Algorithm 7.1 is the self-orthogonal greedy algorithm extended with the requirement $||\mathbf{x}|| \geq d$. So, if it is not mentioned specifically, the term self-orthogonal(self-dual) lexicode with minimum distance d implies that it is produced by Algorithm 7.1. We verified by computer calculations that Algorithm 6.1 also produces the same lexicode of Example 7.2.1 for

parameter values $n = 12$ and $d = 6$. So, the self-orthogonal lexicode produced by Algorithm 7.1 in Example 7.2.1, is a subcode of the lexicode with these parameter values. Since the lexicode is indeed a self-dual code, we can easily observe that for any relevant value of n with minimum distance 6, self-orthogonal lexicodes are subcodes of lexicodes. More generally, we pose the following conjecture.

Conjecture 7.2.5. *In the ternary case, for a* (0-1)*-triangular basis and for minimum distance* $d \equiv 0$ *or* $6(\ mod\ 9)$*, the self-orthogonal lexicode produced by Algorithm 7.1 is a subcode of the generated lexicode for the same parameter values.*

Notice that for a minimum distance not equal to 0 or 6 mod 9, Conjecture 7.2.5 is not always true, as is illustrated by the next three examples.

Example 7.2.2. When the lexicographic list is ordered with respect to the standard basis, the ternary lexicode C of length 7 and minimum distance 3 appears to have the basis vectors 0000111, 0001012, 0110001, and 10100002, whereas the ternary self-orthogonal lexicode C^o with the same parameter values is generated by basis vectors 0000111, 0001012, and 1110000. The basis vector 0110001 of C is at Hamming distance 2 from the basis vector 1110000 of C^o. This implies that the vector 1110000 of C^o is rejected as member of C. So, the self-orthogonal lexicode C^o is not a subcode of the lexicode C.

Example 7.2.3. When the lexicographic list is ordered with respect to the ordered basis $(0000001, 0000011, 0000111, 0001111, 0011111, 0111110, 1111100)$, the lexicode C of length 7 and minimum distance 3, obtained by applying Algorithm 6.1, has basis vectors 0000111, 0001120, 0111110, and 1122211, whereas the self-orthogonal lexicode C^o produced by Algorithm 7.1, for the same parameter values, has basis vectors 0000111, 0001120, and 1111120. It is clear that basis vector 1111120 of C^o and basis vector 0111110 of C have Hamming distance 2. Hence, the vector 1111120 does not belong to the code C. Thus, C^o is not a subcode of C.

In both examples we see that the third basis vector of C^o is rejected by Algorithm 6.1, because of its counterpart in the lexicode C. However, the following example shows that the above phenomenon does not always occur. It shows that a length 7 self-orthogonal lexicode with minimum distance 3 can be a subcode of the related lexicode, for some appropriate basis.

Example 7.2.4. Let $GF(3)^7$ be lexicographically ordered with respect to the following basis vectors 0000001, 00000011, 0000101, 0001010, 0010101, 0110110, and 1010110. The resulting lexicode C produced by Algorithm 6.1 with $P[\mathbf{x}]$ if and only if $d \geq 3$, is generated by basis vectors 0000112, 0001011, 0110110, and 1010111. On the other hand, the related self-orthogonal lexicode C^o, produced by Algorithm 7.1, is generated by the basis vectors 0000112, 0001011, and 1120221. Here, the basis vector 1120221 is equal to the linear combination of the basis vectors 0110110 and 1010111 in the lexicode. Since the subcode $C_2 \subset C^o$ spanned by the first two basis vectors is a subcode of the lexicode C, it is obvious that $span(1120221, C_2)$ is again in C.

From the above examples, it follows that we cannot say whether or not a self-orthogonal lexicode of length 7 with minimum distance 3 is a subcode of the corresponding lexicode. For $n \geq 8$ the situation is different as the next theorem will show.

Before proving this theorem, we remark that column permutations and permutations of the non-zero elements of the field will not affect the mutual Hamming distance and the orthogonality of codewords.

Theorem 7.2.6. *For code length at least* 8*, minimum distance* 3*, and with respect to a triangular basis, a self-orthogonal lexicode produced by Algorithm 7.1 cannot be a subcode of the corresponding lexicode produced by Algorithm 6.1.*

Proof. We shall denote the lexicode of dimension i by C_i, and the self-orthogonal lexicode of dimension j, produced by Algorithm 7.1, by C_j^o, $j \leq i$. For a (0-1)-triangular basis, it is easy to see that the first codeword selected by the greedy algorithm is the codeword $\mathbf{a}_1 = 0000011a_1^1$ of weight 3. So, the intermediate lexicode C_1 has list

$$\begin{array}{l} 00000000 \\ 0000011a_1^1 \\ 0000022a_1^{1'} \end{array}$$

with $a_1^{1'} = 2a_1^1$. Due to the previous remark, without loss of generality, we can take for C_1 the list

$$\begin{array}{l} 00000000 \\ 00000111 \\ 00000222 \end{array}$$

So, we assume that $\mathbf{a}_1 = 00000111$, and hence $\mathbf{a}_1{}' = 00000222$. Let $\mathbf{x} = 00000x_3x_2x_1$. If $||\mathbf{x}|| = 3$, then it follows immediately that $d(\mathbf{x}, C_1) \leq 1$. Assume now that $||\mathbf{x}|| = 2$. If the two non-zero digits of $x_3x_2x_1$ are the same, then it is clear that $d(\mathbf{x}, C_1) = 1$. If the two non-zero digits are different, like in $\mathbf{x} = 00000102$ for instance, then $\mathbf{x}$ is at Hamming distance 2 from C_1.

Let $\rho(\mathbf{v}) = d(\mathbf{v}, C)$, for every $\mathbf{v} \in V$(see the definition right before Lemma 7.2.7). For the code C_1, we may conclude that $max\{\rho(\mathbf{v})|\mathbf{v} = 00000v_3v_2v_1\} = 2$, and that the vector $\mathbf{x} = 0000x_3x_2x_1$ with $x_3x_2x_1 \in \{012, 021, 102, 120, 201, 210\}$, is the only vector which is at Hamming distance 2 from C_1. This implies that the next selected vector $\mathbf{a}^2 = 000a_4^2a_3^2a_2^2a_1^2$ is a vector of weight 3 with $a_4^2 \neq 0$ and $a_3^2a_2^2a_1^2 \in \{012, 021, 102, 120, 201, 210\}$. Whatever the string $a_3^2a_2^2a_1^2$ is, the next six codewords will consist of the codewords $0000a_4^2\mathbf{a}_1$, $0000a_4^2\mathbf{a}_2$, $0000a_4^2\mathbf{a}_3$, $0000a_4^2\mathbf{a}_4$, $0000a_4^2\mathbf{a}_5$, $0000a_4^2\mathbf{a}_6$, with $\mathbf{a}_i \in \{012, 021, 102, 120, 201, 210\}$. Without loss of generality, by applying some column permutations if necessary, we may assume C_2 to be the list

00000000
00000111
00000222
00001012
00001120
00001201
00002021
00002102
00002210

Of course, $\mathbf{a}^2$ is orthogonal to itself and $\mathbf{a}^2$ is also orthogonal to C_1. It implies that $C_2 = span(\mathbf{a}_2, C_1)$ is self-orthogonal. Without considering the zero columns (the first four columns from the left), C_2 is even self-dual. Notice that Algorithm 7.1 will also select the vector $\mathbf{a}^2$ as the second basis vector. At this point, the lexicode and the self-orthogonal lexicode are indeed the same. For the sake of convenience, we sometimes consider codewords in C_2 as strings of length 4, when counting from right to left.

For the code C_2, we can verify that $max\{\rho(\mathbf{b})|\mathbf{b} = b_4b_3b_2b_1\} = 1$. So, Algorithm 6.1 will select as next basis vector for C_3 a vector $\mathbf{a}^3 = 00a_6^3a_5^3a_4^3a_3^3a_2^3a_1^3$ with $a_6^3 \neq 0$, $a_5^3 \neq 0$ and $d(a_4^3a_3^3a_2^3a_1^3, C_2) = 1$. Thus, $C_3 = span(\mathbf{a}^3, C_2)$. Again, we can easily see that $\mathbf{a}^3 = 0011a_4^3a_3^3a_2^3a_1^3$ with $d(0000a_4^3a_3^3a_2^3a_1^3, C_2) = 1$. The possibilities for the weight of the vector $\mathbf{a}^3$ are $3, 4, 5$, and 6, since the string $a_4^3a_3^3a_2^3a_1^3$ is possibly equal to 0001, 0011, 0122, or 1111. Since C_2 is a self-dual code, apart from the first four zero-columns from the left, Algorithm 7.1 will not select a basis vector which has the same pattern as the vector $\mathbf{a}^3 = 00a_6^3a_5^3a_4^3a_3^3a_2^3a_1^3$, since this vector is not orthogonal to C_2. Instead, it will select a vector of the form $\mathbf{u} = 011u\mathbf{c}$, for some $\mathbf{c} \in C_2$, $u \neq 0$, as the next basis vector for C_3^o. If $u = 1$, i.e. $\mathbf{u} = 0111\mathbf{c}$, we obtain $0 < d(\mathbf{a}^3, 0111\mathbf{c}) < 3$ for some $\mathbf{c} \in C_2$. At this point, we conclude that the self-orthogonal lexicode which is equal to $C_3^o = span(\mathbf{u}, C_2)$ is not a subcode of the related lexicode C_3. What happens if $u = 2$? We can easily verify that $d(\mathbf{u}, C_3) = 3$ in that case. Could the vector $\mathbf{u}$ be selected as next basis vector of the lexicode? To satisfy the designed minimum distance requirement, Algorithm 6.1 must select a basis vector for C_4 of type $\mathbf{a}^4 = 01a_6^4a_5^4a_4^4a_3^4a_2^4a_1^4$, with $a_6^4 \neq a_5^4$ and $d(0000a_4^4a_3^4a_2^4a_1^4, C_2) = 1$. Let $\mathbf{c}'$ be a codeword in C_2 such that $d(a_4^4a_3^4a_2^4a_1^4, \mathbf{c}') = 1$, and let $\mathbf{c}''$ be also a codeword in C_2 such that $\mathbf{u}' = 0112\mathbf{c}' = \mathbf{u} + 0000\mathbf{c}''$. It is clear that $\mathbf{u}' \in C_3^o$. If $a_6^4 = 1$ and $a_5^4 = 2$ (or $a_6^4 = 2$ and $a_5^4 = 1$), then $d(\mathbf{a}^4, \mathbf{u}') = 2$. It implies that $\mathbf{u}'$ is rejected as a codeword of the lexicode, hence C_3^o is not a subcode of C_3. Assume that $a_6^4a_5^4 \in \{01, 02, 10, 20\}$. If we take some codeword $\mathbf{w} = \alpha\mathbf{a}^4 + \beta\mathbf{a}^3$, $\alpha, \beta \in \{1, 2\}$, then we have for any $a_6^4a_5^4 \in \{01, 02, 10, 20\}$, and for some $\alpha, \beta \in \{1, 2\}$, that $\mathbf{w} = 0w_712w_4w_3w_2w_1$ with $w_7 \neq 0$ and with $d(0000w_4w_3w_2w_1, C_2) \geq 1$. So, $d(\mathbf{w}, \mathbf{u}) < 3$. If $0 < d(\mathbf{w}, \mathbf{u}) < 3$, we conclude that the self-orthogonal lexicode cannot be the subcode of the corresponding lexicode.

Next we shall also show that if $d(\mathbf{u}, C_4) = 0$, i.e. if $\mathbf{a}^4 = 0101a_4^4a_3^4a_2^4a_1^4$ with $\alpha = 1$ and $\beta = 1$, or $\mathbf{a}^4 = 0120a_4^4a_3^4a_2^4a_1^4$ with $\alpha = 1$ and $\beta = 2$, then the self-orthogonal

lexicode cannot be the subcode of the corresponding lexicode either.

Assume that the case $d(\mathbf{u}, C_4) = 0$ occurs, for some $\alpha, \beta \in \{1, 2\}$. In other words, we can say that $\mathbf{a}^4 = 0101a_4^4a_3^4a_2^4a_1^4$ or $\mathbf{a}^4 = 0120a_4^4a_3^4a_2^4a_1^4$. Notice that the first four components of C_4 from the left will constitute strings 0000, 0011, 0022, 0101, 0112, 0120, 0202, 0210, 0221, if $\mathbf{a}^4 = 0101a_4^4a_3^4a_2^4a_1^4$, and strings 0011, 0022,0120,0101,0112,0210,0221,0202, if $\mathbf{a}^4 = 0120a_4^4a_3^4a_2^4a_1^4$. Since $d(\mathbf{w}, \mathbf{u}) = 0$ for some $\alpha, \beta \in \{1, 2\}$, the strings 0000,0112 and 0221 are followed by all $\mathbf{c} \in C_2$, and the others, i.e. the strings 0011,0022,0101,0120,0202, and 0210, are followed by strings $\mathbf{g}$ which are not in C_2, and some of these $\mathbf{g}$'s satisfy the property $d(\mathbf{g}, C_2) = 1$.

Next, we shall examine the type of the next basis vector - say $\mathbf{z}$ - which will be selected by Algorithm 7.1. Since this vector must be orthogonal to all previously selected codewords, and also must have a distance at least 3 to those codewords, it follows that the first four positions of the selected vector from the right constitute a string in C_2, and the remaining part of the vector has weight 3, and is orthogonal to both 0112 and 0221. We can verify that this vector must be equal to one of the vectors 1011$\mathbf{c}$, 1101$\mathbf{c}$, 1022$\mathbf{c}$, 1202$\mathbf{c}$, 1120$\mathbf{c}$ or 1210$\mathbf{c}$ with $\mathbf{c} \in C_2$. It appears that any linear combination of these vectors and 0112$\mathbf{c}$ will produce vectors having patterns $x120\mathbf{c}$ and $x210\mathbf{c}$, with $x \neq 0$ and $\mathbf{c} \in C_2$. So, $d(1120\mathbf{c}, 0120\mathbf{g}) = 2$ or $d(1210\mathbf{c}, 0210\mathbf{g}) = 2$ for some $\mathbf{c} \in C_2$. In this case, the vector 1120$\mathbf{c}$ or 1210$\mathbf{c}$ is rejected to be an element of C_4. Thus, in any case we proved that $C_3^o = span(\mathbf{u}, C_2)$ or $C_4^o = span(\mathbf{z}, C_3^o)$ can not be a subcode of C_4. □

If $d \equiv 0 (\text{mod } 9)$, $d \equiv 3 (\text{mod } 9)$ and $d \equiv 6 (\text{mod } 9)$, we can also consider $d = 3(3k)$, $d = 3(3k + 1)$, and $d = 3(3k + 2)$, $k \geq 1$ respectively. In the next three theorems we establish the first three basis vectors of self-orthogonal lexicodes for these d-values. The relevance of these theorems lies in minimizing the time complexity of the implementation of Algorithm 7.1. We need the following lemma to complete the proof of these theorems. This lemma and its proof are presented in [51]. The proof of this Lemma is referred to when proving Theorem 7.2.8, Theorem 7.2.9, and Theorem 7.2.10.

For some code C, we define for every $\mathbf{v} \in V$, $\rho(\mathbf{v}) = d(\mathbf{v}, C)$. The maximum value of $\rho(\mathbf{v})$ for all $\mathbf{v} \in V$, is called the *covering radius* of the code C

Lemma 7.2.7. [*51, Lemma*1.3.2] *The covering radius of a q-ary* $[n, 1, n]$ *repetition code is* $\lfloor \frac{q-1}{q} n \rfloor$.

Proof. Let C denote the q-ary $[n, 1, n]$ repetition code. So, $C = \{\mathbf{0}^n, \mathbf{1}^n, \ldots, (\mathbf{q-1})^n\}$, where $\mathbf{i}^n$ is the n-vector with all of its components equal to i. Let $\mathbf{x} = x_n x_{n-1} \cdots x_1$ be a vector in $GF(q)^n$ such that each element of $GF(q)$ occurs $\lfloor \frac{n}{q} \rfloor$ times as a coordinate of $\mathbf{x}$. So, $q\lfloor \frac{n}{q} \rfloor$ positions of $\mathbf{x}$ are filled. Let the $n - q\lfloor \frac{n}{q} \rfloor$ remaining positions of $\mathbf{x}$ be filled with the elements of $GF(q)$ such that the numbers of the digits $0, 1, \ldots, (q-1)$ differ at most one. By this condition, the occurrence of any element of $GF(q)$ in $\mathbf{x}$ is at most $\lceil \frac{n}{q} \rceil$ times. So, the vector $\mathbf{x}$ agrees with every element of C in at most $\lceil \frac{n}{q} \rceil$ positions, implying that the distance of $\mathbf{x}$ to C is equal to $n - \lceil \frac{n}{q} \rceil = \lfloor \frac{q-1}{q} n \rfloor$. This is the maximum distance a vector in $GF(q)^n$ can have to C. □

We shall frequently omit the first zero components when writing vectors explicitly. For example, the vector 00001111 will be written as 1111.

Theorem 7.2.8. *If $d = 3(3k)$, $k \geq 1$, and if the basis is a triangular ordered basis, then the first three basis vectors that will be selected by Algorithm 7.1 are equal to* $\mathbf{a}_1 = \mathbf{1}^d$, $\mathbf{a}_2 = \mathbf{1}^{3k}\mathbf{0}^{3k}\mathbf{1}^{3k}\mathbf{2}^{3k}$, *and* $\mathbf{a}_3 = \mathbf{1}^k\mathbf{z}^{12k}$.

Proof. Here, we denote the self-orthogonal lexicode of dimension i by C_i^o and the i-th basis vector of C_i^o by $\mathbf{a}_i$. We write $d = 3\delta$ with $\delta = 3k$, $k \geq 0$. The first basis vector of C_1^o that will be selected by Algorithm 7.1 is the vector $\mathbf{a}_1 = \mathbf{1}^d = \mathbf{1}^\delta\mathbf{1}^\delta\mathbf{1}^\delta$. This vector is also selected by Algorithm 6.1 as the first basis vector. According to Lemma 7.2.7, the next selected vector, is the vector $\mathbf{a}_2 = \mathbf{1}^\delta\mathbf{a}^\delta\mathbf{b}^\delta\mathbf{c}^\delta$, where the numbers of 0's, 1's and 2's in the string $\mathbf{a}^\delta\mathbf{b}^\delta\mathbf{c}^\delta$ are equal (See the proof of Lemma 7.2.7). Without loss of generality we may consider $\mathbf{a}_2$ as the vector $\mathbf{1}^\delta\mathbf{0}^\delta\mathbf{1}^\delta\mathbf{2}^\delta$. One can verify that the vector $\mathbf{1}^\delta\mathbf{0}^\delta\mathbf{1}^\delta\mathbf{2}^\delta + \mathbf{c}$ is orthogonal to itself for all previous codewords $\mathbf{c}$ in C_1^o. The list of this code has the following form

$$
\begin{array}{cccc}
\overbrace{0\cdots0}^{\delta} & \overbrace{0\cdots0}^{\delta} & \overbrace{0\cdots0}^{\delta} & \overbrace{0\cdots0}^{\delta} \\
0\cdots0 & 1\cdots1 & 1\cdots1 & 1\cdots1 \\
0\cdots0 & 2\cdots2 & 2\cdots2 & 2\cdots2 \\
1\cdots1 & 0\cdots0 & 1\cdots1 & 2\cdots2 \\
1\cdots1 & 1\cdots1 & 2\cdots2 & 0\cdots0 \\
1\cdots1 & 2\cdots2 & 0\cdots0 & 1\cdots1 \\
2\cdots2 & 0\cdots0 & 2\cdots2 & 1\cdots1 \\
2\cdots2 & 1\cdots1 & 0\cdots0 & 2\cdots2 \\
2\cdots2 & 2\cdots2 & 1\cdots1 & 0\cdots0
\end{array}
$$

Let us denote the next selected vector by $\mathbf{1}^\triangle\mathbf{x}^\delta\mathbf{o}^\delta\mathbf{p}^\delta\mathbf{q}^\delta$. Again, due to Lemma 7.2.7, we can infer that the vector $\mathbf{x}^\delta\mathbf{o}^\delta\mathbf{p}^\delta\mathbf{q}^\delta$ which contains digits 0, 1, and 2 in a balanced way, i.e. the differences between the numbers of digits 0, 1 and 2, respectively, are at most one, has a distance to C_2^o equal to the covering radius of C_2^o. Without loss of generality, the vector $\mathbf{1}^\triangle\mathbf{x}^\delta\mathbf{o}^\delta\mathbf{p}^\delta\mathbf{q}^\delta$ can be written as

$$
\overbrace{1\cdots1}^{\triangle}\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}
\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}. \tag{7.1}
$$

By adding vector (7.1) to all previously generated codewords, we have that all next 18 codewords will have the same weight $8k+\ \triangle$. To satisfy the minimum distance requirement, we have that $\triangle = 9k - 8k = k$. With this value of $\triangle$, the 18 codewords will be orthogonal to themselves, and hence Algorithm 7.1 may also select a vector of the same type with (7.1) as $\mathbf{a}_3$. Of course, Algorithm 7.1 may also select a vector which has a pattern different from (7.1), but still with the value of $\triangle$ equal to k. □

Theorem 7.2.9. *If $d = 3(3k+1)$, $k \geq 1$, and if the basis is a triangular ordered basis, then Algorithm 7.1 will select the first three basis vectors equal to $\mathbf{a}_1 = \mathbf{1}^d$, $\mathbf{a}_2 = \mathbf{1}^{3k+1}\mathbf{0}^{3k+1}\mathbf{1}^{3k+1}\mathbf{2}^{3k+1}$, and $\mathbf{a}_3 = \mathbf{1}^{\triangle}\mathbf{z}^{12k+4}$ with $\triangle \geq k+2$.*

Proof. As before, we denote the self-orthogonal lexicode of dimension i by C_i^o and the i-th basis vector of C_i^o by $\mathbf{a}_i$. We write $d = 3\delta$ with $\delta = (3k+1)$, $k \geq 0$. Like in the proof of Theorem 7.2.8, let $\mathbf{a}_1 = \mathbf{1}^d = \mathbf{1}^\delta\mathbf{1}^\delta\mathbf{1}^\delta$ and $\mathbf{a}_2 = \mathbf{1}^\delta\mathbf{a}^\delta\mathbf{b}^\delta\mathbf{c}^\delta$, where the numbers of 0's, 1's and 2's in the string $\mathbf{a}^\delta\mathbf{b}^\delta\mathbf{c}^\delta$ are equal. Assume again the list of $C_2^o = span(\mathbf{a}_1, \mathbf{a}_2)$ to be of the following form

$$\begin{array}{cccc}
\overbrace{0\cdots0}^{\delta} & \overbrace{0\cdots0}^{\delta} & \overbrace{0\cdots0}^{\delta} & \overbrace{0\cdots0}^{\delta} \\
0\cdots0 & 1\cdots1 & 1\cdots1 & 1\cdots1 \\
0\cdots0 & 2\cdots2 & 2\cdots2 & 2\cdots2 \\
1\cdots1 & 0\cdots0 & 1\cdots1 & 2\cdots2 \\
1\cdots1 & 1\cdots1 & 2\cdots2 & 0\cdots0 \\
1\cdots1 & 2\cdots2 & 0\cdots0 & 1\cdots1 \\
2\cdots2 & 0\cdots0 & 2\cdots2 & 1\cdots1 \\
2\cdots2 & 1\cdots1 & 0\cdots0 & 2\cdots2 \\
2\cdots2 & 2\cdots2 & 1\cdots1 & 0\cdots0
\end{array}$$

The vector $\mathbf{1}^{\triangle}\mathbf{x}^\delta\mathbf{o}^\delta\mathbf{p}^\delta\mathbf{q}^\delta$, where $\mathbf{x}^\delta$, $\mathbf{o}^\delta$, $\mathbf{p}^\delta$, and $\mathbf{q}^\delta$ contain digits $0, 1$, and 2 in a balanced way, will have maximum distance to C_2^o. Without loss of generality, the vector $\mathbf{1}^{\triangle}\mathbf{x}^\delta\mathbf{o}^\delta\mathbf{p}^\delta\mathbf{q}^\delta$ can be written as

$$\overbrace{1\cdots1}^{\triangle}\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}r\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}s\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}t\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}u. \tag{7.2}$$

with $k = (d-1)/3$, $r, s, t, u \in 0, 1, 2$. By adding vector (7.2) to all previously generated codewords, we have that all next 18 codewords will have the same weight $8k+\triangle$, apart from the columns which contain the digits r, s, t, and u.

Let us next consider these columns of C_2^o containing the digits r, s, t, and u. The digits of each codeword of C_2^o associated to these columns generate a vector of length 4. Let W be the set containing all these vectors. It will be clear that $W = \{0000, 0111, 0222, 1012, 1120, 1201, 2021, 2102, 2210\}$. As remarked in the proof of Theorem 7.2.6, we shall have $max\{\rho(v)|\mathbf{v} \text{ vector of length } 4\} = 1$ for the set W. It implies that the maximum distance vector (7.2) can have to C_2^o is equal to $8k+1+\triangle$. This occurs when $rstu$ is not in W. To meet the minimum distance requirement, we must have

$$\triangle = d - 8k - 1 = 3(3k+1) - 8k - 1 = k + 2.$$

Moreover, one can verify that for every vector $\mathbf{y}$ of length 4 which is not in W, there exists some vector $\mathbf{w} \in W$ such that $||\mathbf{y} + \mathbf{w}|| = 1, 2, 3$, and 4. It follows that the weights of the next 18 codewords, produced by adding vector (7.2) and its multiples to all codewords of C_2^o, are equal to $d = 3(3k+1)$, $d+1$, $d+2$, or $d+3$. Since not all these codewords are orthogonal to themselves, this type of vector will not be selected by Algorithm 7.1 as the next basis vector for C_3^o. If for this effective length, a vector exists satisfying the property prescribed by Algorithm 7.1, the algorithm will select a vector of a type different from (7.2). We finally may conclude that the selected vector is of the form $\mathbf{a}_3 = \mathbf{1}^{\triangle}\mathbf{z}^{12k+4}$ with $\triangle \geq k+2$. □

Theorem 7.2.10. *If $d = 3(3k+2)$, $k \geq 1$, and if the basis is a triangular ordered basis, then Algorithm 7.1 will select the first three basis vectors equal to $\mathbf{a}_1 = \mathbf{1}^d$, $\mathbf{a}_2 = \mathbf{1}^{3k+2}\mathbf{0}^{3k+2}\mathbf{1}^{3k+2}\mathbf{2}^{3k+2}$, and $\mathbf{a}_3 = \mathbf{1}^{\triangle}\mathbf{z}^{12k+8}$ with $\triangle \geq k+1$.*

Proof. Here again, we denote the self-orthogonal lexicode of dimension i by C_i^o and the i-th basis vector of C_i^o by $\mathbf{a}_i$. We write $d = 3\delta$ with $\delta = (3k+2)$, $k \geq 0$. The first vector basis of C_1^o that will be selected by Algorithm 7.1, is the vector $\mathbf{1}^d = \mathbf{1}^\delta\mathbf{1}^\delta\mathbf{1}^\delta$. So, $\mathbf{a}_1$ is equal to $\mathbf{1}^d$. According to Lemma 7.2.7, the vector of type $\mathbf{1}^\delta\mathbf{a}^\delta\mathbf{b}^\delta\mathbf{c}^\delta$, where the numbers of 0's, 1's and 2's in the string $\mathbf{a}^\delta\mathbf{b}^\delta\mathbf{c}^\delta$ are equal, has maximum distance to C_2^o. Without loss of generality, we may assume this vector to be $\mathbf{1}^\delta\mathbf{0}\mathbf{1}^\delta\mathbf{2}^\delta$. Since the vector $\mathbf{1}^\delta\mathbf{0}^\delta\mathbf{1}^\delta\mathbf{2}^\delta + \mathbf{c}$ is orthogonal to itself for all previous codewords $\mathbf{c}$ in C_1^o, the vector $\mathbf{1}^\delta\mathbf{0}^\delta\mathbf{1}^\delta\mathbf{2}^\delta$ will be selected as $\mathbf{a}_2$. We may assume again that C_2^o is the following list

$$
\begin{array}{cccc}
\overbrace{0\cdots0}^{\delta} & \overbrace{0\cdots0}^{\delta} & \overbrace{0\cdots0}^{\delta} & \overbrace{0\cdots0}^{\delta} \\
0\cdots0 & 1\cdots1 & 1\cdots1 & 1\cdots1 \\
0\cdots0 & 2\cdots2 & 2\cdots2 & 2\cdots2 \\
1\cdots1 & 0\cdots0 & 1\cdots1 & 2\cdots2 \\
1\cdots1 & 1\cdots1 & 2\cdots2 & 0\cdots0 \\
1\cdots1 & 2\cdots2 & 0\cdots0 & 1\cdots1 \\
2\cdots2 & 0\cdots0 & 2\cdots2 & 1\cdots1 \\
2\cdots2 & 1\cdots1 & 0\cdots0 & 2\cdots2 \\
2\cdots2 & 2\cdots2 & 1\cdots1 & 0\cdots0
\end{array}
$$

We denote the next selected vector by $\mathbf{1}^{\triangle}\mathbf{x}^\delta\mathbf{o}^\delta\mathbf{p}^\delta\mathbf{q}^\delta$. This vector will have maximum distance to C_2^o if $\mathbf{x}^\delta$, $\mathbf{o}^\delta$, $\mathbf{p}^\delta$, and $\mathbf{q}^\delta$ contain digits 0, 1, and 2 in a balanced way, i.e. if the differences between the numbers of digits 0, 1 and 2 are at most one (See again the proof of Lemma 7.2.7). Without loss of generality, the vector $\mathbf{1}^{\triangle}\mathbf{x}^\delta\mathbf{o}^\delta\mathbf{p}^\delta\mathbf{q}^\delta$ can be written as

$$
\overbrace{1\cdots1}^{\triangle}\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}00\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}11\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}22
$$

$$
\overbrace{\underbrace{0\cdots0}_{k}\underbrace{1\cdots1}_{k}\underbrace{2\cdots2}_{k}}^{\delta}01, \qquad (7.3)
$$

with $k = (\delta - 2)/3$. By adding the vector (7.3) to all previously generated codewords, we have that all next 18 codewords will have weight $8k + 5 + \triangle$ and $8k + 6 + \triangle$. To meet the minimum distance condition we must have that $\triangle = 9k + 6 - 8k - 5 = k + 1$. It is clear that not all corresponding vectors are orthogonal to itself, hence Algorithm 7.1 will not select this type of vector as $\mathbf{a}_3$. If some vector exists with this effective length, which satisfies the selection criterion, then Algorithm 7.1 will select a type of vector different from (7.3). It implies that the vector $\mathbf{a}_3 = \mathbf{1}^{\triangle}\mathbf{z}^{12k+8}$ which is selected by Algorithm 7.1 will satisfy the condition that $\triangle \geq k + 1$. □

Remark 7.2.1. We emphasize again that the relevance of Theorems 7.2.8, 7.2.10 and 7.2.9 lies in a possible implementation of Algorithm 7.1 in a computer program. This knowledge can be used to make a "jumping-loop" from the effective length d to length $d + \frac{d}{3}$, and from length $d + \frac{d}{3}$ to length $d + \frac{d}{3} + \triangle$, for some $\triangle$.

Appendix A

The list of an 8-bit ternary self-dual lexicode

00000000	11022012	22011021
00000111	11022120	22011102
00000222	11022201	22011210
00001102	11020111	22012120
00001210	11020222	22012201
00001021	11020000	22012012
00002201	11021210	22010222
00002012	11021021	22010000
00002120	11021102	22010111
01111021	12100000	20122012
01111102	12100111	20122120
01111210	12100222	20122201
01112120	12101102	20120111
01112201	12101210	20120222
01112012	12101021	20120000
01110222	12102201	20121210
01110000	12102012	20121021
01110111	12102120	20121102
02222012	10211021	21200000
02222120	10211102	21200111
02222201	10211210	21200222
02220111	10212120	21201102
02220222	10212201	21201210
02220000	10212012	21201021
02221210	10210222	21202201
02221021	10210000	21202012
02221102	10210111	21202120

Appendix B

The lists of $G_4(3)$, $G_4(3)^2$, and $C_4(2;4)$

$G_4(3)$		$G_4(3)^2$		$C_4(2;4)$	
000	220	200	020	2000	0200
001	221	201	021	2013	0213
002	222	202	022	2022	0222
003	223	203	023	2031	0231
013	233	213	033	2130	0330
010	230	210	030	2103	0303
011	231	211	031	2112	0312
012	232	212	032	2121	0321
022	202	222	002	2220	0020
023	203	223	003	2233	0033
020	200	220	000	2202	0002
021	201	221	001	2211	0011
031	211	231	011	2310	0110
032	212	232	012	2323	0123
033	213	233	013	2332	0132
030	210	230	010	2301	0101
130	310	330	110	3300	1100
131	311	331	111	3313	1113
132	312	332	112	3322	1122
133	313	333	113	3331	1131
103	323	303	123	3030	1230
100	320	300	120	3003	1203
101	321	301	121	3012	1212
102	322	302	122	3021	1221
112	332	312	132	3120	1320
113	333	313	133	3133	1333
110	330	310	130	3102	1302
111	331	311	131	3111	1311
121	301	321	101	3210	1010
122	302	322	102	3223	1023
123	303	323	103	3232	1032
120	300	320	100	3201	1001

Appendix C

The lists of $G_5(3)$, $G_5(3)^2$, and $C_5(3;4)$

$G_5(3)$

000	140	230	320	410
001	141	231	321	411
002	142	232	322	412
003	143	233	323	413
004	144	234	324	414
014	104	244	334	424
010	100	240	330	420
011	101	241	331	421
012	102	242	332	422
013	103	243	333	423
023	113	203	343	433
024	114	204	344	434
020	110	200	340	430
021	111	201	341	431
022	112	202	342	432
032	122	212	302	442
033	123	213	303	443
034	124	214	304	444
030	120	210	300	440
031	121	211	301	441
041	131	221	311	401
042	132	222	312	402
043	133	223	313	403
044	134	224	314	404
040	130	220	310	400

$G_5(3)^2$

300	440	030	120	210
301	441	031	121	211
302	442	032	122	212
303	443	033	123	213
304	444	034	124	214
314	404	044	134	224
310	400	040	130	220
311	401	041	131	221
312	402	042	132	222
313	403	043	133	223
323	413	003	143	233
324	414	004	144	234
320	410	000	140	230
321	411	001	141	231
322	412	002	142	232
332	422	012	102	242
333	423	013	103	243
334	424	014	104	244
330	420	010	100	240
331	421	011	101	241
341	431	021	111	201
342	432	022	112	202
343	433	023	113	203
344	434	024	114	204
340	430	020	110	200

$C_5(3;4)$

3000	4400	0300	1200	2100
3014	4414	0314	1214	2114
3023	4423	0323	1223	2123
3032	4432	0332	1232	2132
3041	4441	0341	1241	2141
3140	4040	0440	1340	2240
3104	4004	0404	1304	2204
3113	4013	0413	1313	2213
3122	4022	0422	1322	2222
3131	4031	0431	1331	2231
3230	4130	0030	1430	2330
3244	4144	0044	1444	2344
3203	4103	0003	1403	2303
3212	4112	0012	1412	2312
3221	4121	0021	1421	2321
3320	4220	0120	1020	2420
3334	4234	0134	1034	2434
3343	4243	0143	1043	2443
3302	4202	0102	1002	2402
3311	4211	0111	1011	2411
3410	4310	0210	1110	2010
3424	4324	0224	1124	2024
3433	4333	0233	1133	2033
3442	4342	0242	1142	2042
3401	4301	0201	1101	2001

Bibliography

[1] Ádám A., *Truth Functions and the Problem of their Realization by Two-Terminal Graphs,* Akadémiai Kiadó, Budapest, 1968.

[2] Agrell, E., J. Lassing, E.G. Ström, and T. Ottosson, "On the optimality of the binary reflected Gray code," *IEEE Trans. Inform. Theory,* vol. 50, 2004.

[3] Bhat, G.S., and C. D. Savage, "Balanced Gray codes," *The Electronic Journal of Combinatorics,* vol. 3, paper R25, 1996.

[4] Bonn, J.T., "Forcing linearity on greedy codes," *Designs, Codes and Crypthography,* vol. 9, pp. 39-49, 1996.

[5] Bose, B. and B. Broeg, "Lee distance Gray codes," *Proceeding of International Symposium on Information Theory,* Sept 17-22, 1995.

[6] Bose, B., B. Broeg, Y. Kwon, and Y Ashir, "Lee distance and topological properties of k-ary n-cubes," *IEEE Trans. of Computers,* vol. 44, pp. 1021-1030, 1995.

[7] Brualdi, R.A. and V. S. Pless, "Greedy codes," *J. Combin. Theory Ser. A,* vol. 64, pp. 10-30, 1993.

[8] Bultena, B. and F. Ruskey, "Transition restricted Gray codes," *The Elect. J. Combin.,* vol. 3 Paper R11, 1996.

[9] Cavior, S.R., "An upper bound associated with errors in Gray code," *IEEE Trans. Inform. Theory,* vol. IT-21, p. 596, 1975.

[10] Cohn, M., "Affine m-ary Gray codes," *Information and control*, vol. 6, pp. 70-78, 1963.

[11] Conder, M., "Explicit definition of the binary reflected Gray codes," *Discrete Mathematics,* vol. 195, pp. 245-249, 1999.

[12] Conway, H., "Integral lexicographic codes," *Discrete Math.,* vol. 83, pp. 219-235, 1990.

[13] Conway, J.H. and N.J.A. Sloane, "Lexicographic codes: Error-correcting codes from game theory," *IEEE Trans. Inform. Theory,* vol. IT-32, No. 3, pp. 337-348, 1986.

[14] Dodunekov, S.M., Bulgarian Academy of Sciences, Bulgaria, *private communication.*

[15] Er, M.C. "On generating the N-ary reflected Gray codes," *IEEE Trans. Computers,* vol. C-33, pp. 739 - 741, 1984.

[16] Evdokimov, A.A., Novosibirsk University, Russia, *private communication.*

[17] Farrell, P.G., "Linear binary anticodes," *Electronics Letters,* vol. 6, pp. 419-421, 1970.

[18] Flores, I., "Reflected number system," *IRE Trans. Electron. Computer,* vol. EC-5, pp. 79-82, 1956.

[19] Fon-Der-Flaass, D., "A note on greedy codes," *J. Combin. Theory Ser. A,* vol. 76, pp. 156-159, 1996.

[20] Gilbert, E.N., "Gray codes and paths on the n-cube," *Bell Syst. Tech. J.,* vol. 37, pp. 815-826, 1958.

[21] Goddyn, L. and P. Gvozdjak, "Binary Gray codes with long bit runs," *The Elect. J. Combin.,* vol. 10, Paper R27, 2003.

[22] Goddyn, L., G.M. Lawrence, and E. Nemeth, "Gray codes with run length," *Utilitas Mathematica,* vol. 34, pp. 179-192, 1988.

[23] Gray, F., "Pulse code communications," U.S. Patent 2632058, 1953.

[24] Gulliver, T.A., V.K. Bhargava, and J.M. Stein, "Q-ary Gray codes and weight distributions," *Applied Math. and Comput.,* vol. 103, pp. 97-109, 1999.

[25] Hayes, J.P., "Generation of optimal transition count tests," *IEEE Trans. Computers,* C-27, pp. 36-41, 1978.

[26] He, M.X., S.V. Petoukhov, and P.E. Ricci, "Genetic code, Hamming distance and stochastic matrices," *Bulletin of Mathematical Biology,* vol. 66, pp. 1405-1421, 2004.

[27] Irshid, I.M., "Gray code weighting system," *IEEE Trans. Inform. Theory,* vol. IT-33, pp. 930-931, 1987.

[28] Johnsson, S.L and C.T. Ho, "On the conversion between binary code and binary-reflected Gray code on binary cubes," vol. C-44, pp. 47-53, 1995.

[29] Kautz, W.H., "A readily implemented single-error-correcting unit-distance counting code," *IEEE Trans. Comput.,* vol. 19, pp. 972-975, 1970.

[30] Killian, C.E. and C.D. Savage, "Antipodal Gray codes," *Discrete Math.,* vol. 281, pp. 221-236, 2004.

[31] Kim, H.J. and J.G. Lee, "Partial sum problem mapping into a hypercube," *Inform. Processing Letters,* vol. 36, pp. 221-224, 1990.

[32] Knuth, D.E., *The Art of Computer Programming, Volume 4*, Addison-Wesley as part of "fascicle" 2, USA, 2005.

[33] Kobayashi, Z. and T. Sekiguchi, "On a characterization of the standard Gray code by using its edge type on a hypercube," *Inform. Proc. Letters,* vol. 81, pp. 231-237, 2002.

[34] Lassing, J., E.G. Ström, E. Agrell, and T. Ottosson, "Computation of the exact bit-error rate of coherent M-ary PSK with Gray code bit mapping," *IEEE Trans. Comm.,* vol. Com-51, 2003.

[35] Lee, C.Y., "Some properties of Non-binary error-correcting codes," *IRE Trans. Inform. Theory,* vol. IT-4, pp. 77-82, 1958.

[36] Lee, P.J., "Computation of the bit error rate of coherent M-ary PSK with Gray bit mapping," *IEEE Trans. Comm.,* vol. COM-34, 1986.

[37] Levenstein, V.I., "A class of systematic codes," *Dokl. Akad. Nauk,* vol. 1, pp. 368-371, 1960.

[38] Lichtner, J. "Iterating an α-ary Gray code," *SIAM J. Discrete Math.,* vol. 11 pp. 381-386, 1998.

[39] Liu, X. and G.F. Schrack, "A heuristic approach for constructing symmetric Gray codes," *Appl. Math. Comput.,* vol. 155, pp. 55-63, 2004.

[40] Lu, C.J. and S.C. Tsai, "A note on iterating an α-ary Gray code," *SIAM J. Discrete Math.,* vol. 14, pp. 237-239, 2001.

[41] Ludman, J.E., "Gray code generation for MPSK signals," *IEEE Trans. Comm.,* vol. C-29, pp. 1519-1522, 1981.

[42] Ludman, J.E. and J.L. Sampson, "A technique for generating Gray codes," *J. Stat. Plan. and Inference,* vol. 5, pp. 171-180, 1981.

[43] MacWilliams, F.J. and N.J.A. Sloane, *The theory of error-correcting codes,* North-Holland, Amsterdam 1988.

[44] Mallows, C.L., V. Pless and N.J.A. Sloane, "Self-dual codes over GF(3)," *J. Appl. Math.* vol. 31, pp. 649-665, 1976.

[45] Mansour, I.I., "Gray code weighting system," *IEEE Trans. Inform. Theory,* vol. IT-21, p. 596, 1975.

[46] Mills, W.H., "Some complete cycles on the n-cube," *Amer. Math. Society,* vol. 14, pp. 640-643, 1963.

[47] Monroe, L., "Self-orthogonal Greedy codes," *Designs, Codes and Crypthography,* vol. 9, pp. 79-83, 1996.

[48] Monroe, L., *Greedy Codes over Binary and Non-binary Fields,* PhD. thesis, University of Illinois at Chicago, 1995.

[49] Monroe, L., "Binary greedy codes," *Congressus Numerantium,* vol. 104, pp. 49-63, 1994.

[50] Moore, J.A. and M.J. Quinn, "Generating an efficient broadcast sequence using reflected Gray codes," *IEEE TRans. Paral. and Dist. System,* vol 8, pp. 1117-1122, 1997.

[51] O'Brien, K., *Construction and properties of greedy codes,* PhD Thesis, Dept. Math. University College Cork, Ireland, 2003.

[52] Park, J.P. and B. Bose, "Separabilities of binary Gray codes designed over $\mathbf{Z}_4$," *IEEE International Symposium of Inform. Theory,* June 29, - July 4, 2003.

[53] Pless, V.S. and W.C. Huffman, Eds., *Handbook of coding theory,* Amsterdam, The Netherlands: Elsevier, 1998.

[54] Pless, V. S., *Introduction to the Theory of Error-Correcting Codes,* 2nd Ed., USA: John Willey and Sons, 1989.

[55] Pless, V. S., "On the uniqueness of the Golay codes," *J. Combin. Theory,* vol. 5, pp. 215-228, 1968.

[56] Porat, D.I. and S. Wojcicki, "Fast Synchronous Gray counter," *Nuclear Instruments and Methods,* vol. 169, pp. 243-244, 1980.

[57] Preparata, F.P. and J. Nievergelt, "Difference-preserving codes," *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 643-649, 1974.

[58] Ramras, M., "A new method of generating Hamiltonian cycle on the n-cube," *Discrete Math.,* vol. 85, pp. 329-331, 1990.

[59] Reingold, E.M., J. Nievergelt, and N. Deo, *Combinatorial Algoriths: Theory and Practice,* Englewood Cliffs, NJ: Prentice-Hall, 1977.

[60] Robinson, J.P. and M. Cohn, *"Counting sequences," IEEE Trans. Computers,* vol. C-30, pp. 17-23, 1981.

[61] Savage, C., "A survey of combinatorial Gray codes," *SIAM Rev.,* vol. 39, pp. 605-629, 1997.

[62] Savage, C.D. and P. Winkler, "Monotone Gray codes and the middle levels problems," *J. Combinatorial Theory, Series A,* vol. 70, pp. 230-248, 1995.

[63] Sharma, B.D. and R. K. Khanna, "On m-ary Gray codes," *Information Sciences,* vol. 15, pp. 31-43, 1978.

[64] Sundberg, C-E., "Bit error probability properties of Gray-coded M.P.S.K. signals," *Elect. Letters,* vol. 11, pp. 542-544. 1975.

[65] Suparta, I N. and A.J. van Zanten, "Balanced maximum counting sequences," to appear in *IEEE Trans. Inform. Theory.*

[66] Suparta, I N., "A simple proof for the existence of exponentially balanced Gray codes," *The Elect. J. Combinatorics,* vol. 12, Paper N19, 2005.

[67] Suparta, I N. and A.J. van Zanten, "A construction of Gray codes inducing complete graphs," submitted for publication to *Discrete Mathematics.*

[68] Suparta, I N. and A.J. van Zanten, "On balanced uniform counting sequences," *Proceedings of the International Conference on Applied Mathematics(ICAM05)*, pp. 305-310, Bandung-Indonesia, Aug. 2005.

[69] Suparta, I N. and A.J. van Zanten, *On balanced maximal counting sequences and balanced uniform counting sequences,* rept. CS 05-01, Institute for Knowledge and Agent Technology, University Maastricht, Maastricht, The Netherlands (2005).

[70] Suparta, I N. and A.J. van Zanten, "On a class of Gray codes with maximum crossover Hamming distance," Proceeding of the Ninth International Workshop ACCT, Kranevo-Bulgaria, pp. 362-367, June 2004.

[71] Suparta, I N. and A.J. van Zanten, "On cyclic N-ary Gray codes," *Proceedings of the SEAMS2003 conference,* Yogyakarta-Indonesia, July 2003.

[72] Suparta, I N. and A.J. van Zanten, *Balanced Gray codes,* rept. CS 03-03, Institute for Knowledge and Agent Technology, Universiteit Maastricht, Maastricht, The Netherlands, March 2003.

[73] Suparta, I N. and A. J. van Zanten, *On the list distance in cyclic N-ary Gray codes,* Rept. CS 02-01, Institute for Knowledge and Agent Technology, Universiteit Maastricht, Jan. 2002.

[74] Tang, D.T. and C.N. Liu, "Distance-2 cyclic chaining of constant-weight codes," *IEEE Trans. Computers,* vol. 22, pp. 176-180, 1973.

[75] Trachtenberg, A., "Designing Lexicographyc codes with a given trellis complexity," *IEEE Trans. Inform. Theory,* vol. 48, pp. 89-100, 2002.

[76] van Zanten, A.J. and I N. Suparta, "On the construction of linear q-ary lexicodes," *Designs, codes and Cryptography,* vol. 37, pp. 15-29, 2005.

[77] van Zanten, A.J. and I N. Suparta, "Totally balanced and exponentially balanced Gray codes," *Discrete Analysis and Operation Research,* vol. 11, No. 4, pp. 81-98, 2004. (Russian Journal)

[78] van Zanten, A.J. and I N. Suparta, *An algorithm for a large class of linear q-ary lexicodes,* rept. CS 03-01, Institute for Knowledge and Agent Technology, Universiteit Maastricht, Maastricht, The Netherlands, 2003.

[79] van Zanten, A.J., and I N. Suparta, "The separability of standard cyclic N-ary Gray codes," *IEEE Trans. on Inform. Theory,* vol. 49, pp. 485-487, 2003.

[80] van Zanten, A.J. and I N. Suparta, "The separability function of the standard cyclic N-ary Gray code," *Proceeding of the Eight International workshop ACCT,* pp. 485-487, Tsarskoe Selo-Russia, Sept. 2002.

[81] van Zanten, A.J., "Cyclic distance-preserving codes on constant-weight basis," *Discrete Appl. Math.,* vol. 114, pp. 289-294, 2001.

[82] van Zanten, A.J. and A. Lukito, "Construction of certain cyclic distance-preserving codes having linear-algebraic characteristics," *Designs, Codes, and Cryptography,* vol. 16, pp. 185-199, 1999.

[83] van Zanten, A.J., "Lexicographic order and linearity," *Design, Codes and Cryptography,* vol. 10, pp. 85-97, 1997.

[84] van Zanten, A.J., "Minimal-change order and separability in linear codes," *IEEE Trans. Inform. Theory,* vol. IT-39, pp. 1988-1989, 1993.

[85] van Zanten, A.J., "Lexicodes over fields of characteristic 2," *rept. 93-126, Faculty of Technical Mathematics and Informatics,* Delft Univ. of Techology, Delft, The Netherlands 1993.

[86] van Zanten, A.J., "Index system and separability of constant weight Gray codes," *IEEE Trans. Inform. Theory,* vol. IT-37, pp. 1229-1233, 1991.

[87] Vickers, V.E. and J. Silverman, "A technique for generating specialized Gray codes," *IEEE Trans. Computers,* vol. C-29, pp. 329-331, 1980.

[88] Wagner, D.J. and J. West, "Construction of uniform Gray codes," *Congressus Numerantium,* vol. 80, pp. 217-223, 1991.

[89] Wilmer, E.L. and M.D. Ernst, "Graphs induced by Gray codes," *Discrete Mathematics*, vol. 257, pp. 585-598, 2002.

[90] Yuen, C.K., "The separability of Gray code," *IEEE Trans. Inform. Theory,* vol. IT-20, p. 668, 1974.

Summary

The topics which we discuss in this thesis can be categorized as part of the theory of ordered codes. These topics can roughly be subdivided into two classes: Counting sequences and Gray codes on the one hand, and Lexicodes on the other. The discussion of the first topic is referred to as the first part of this thesis. The second part of the thesis contains the remaining topic.

Generally a *counting sequence* is a list of (all) 2^n binary n-tuples(or binary words of length n). Counting sequences have applications in logic-circuits. Two types of special counting sequences constitute the main issue of our study in Part I. The first type deals with counting sequences having the property that the number of bit changes from one codeword to its successor is as *large* as possible. In our thesis, this kind of counting sequence is called *maximum counting sequence*. An instance where maximum counting sequences are applied is the testing of physical circuits for reliable behavior in worst-case conditions (see e.g. [32, Exercise 67, p. 35]). The second interesting type of counting sequence we study, is called *uniform counting sequence*, i.e. a counting sequence having the property that all pairs of successive words in the sequence have the same Hamming distance. Among these two types of counting sequences, *balanced counting sequences* are of considerable interest in combinatorial logic-circuits. The discussion of maximum and uniform counting sequences mainly takes place in Chapter 5.

Binary Gray codes which constitute a special type of uniform counting sequence is a well-known topic. In this type of counting sequence any two successive codewords differ in precisely one bit, i.e. their Hamming distance is 1. Among all kinds of Gray codes, the *binary reflected Gray code*, also known as the *standard Gray code*, is the best known(cf. [59, 86]). This code was a patented invention due to Frank Gray in 1953, and was used to reduce the coding errors in a pulse code communication system [23]. The code itself however, was demonstrated already in 1878 as an application in a telegraph device by the French engineer Émile Baudot(cf. [26]).

Although the binary reflected Gray code has found its widespread applications, for instance in algebraic coding theory (cf. [84]), in the design of combinatorial algorithms (cf. [59]), and is even found to be optimal with respect to various other applications(cf. [2]), sometimes Gray codes with additional properties are requested for special applications. For instance, when designing experiments, or when designing and testing electrical circuits and information systems, *balanced Gray codes* are needed(cf. [3, 39, 41, 42, 87, 88]). Moreover, applications of the *N-ary n-cube* can be found in the design of several concurrent computers including the Ametek 2020,

the J-Machine, the Mosaic, the iWarp, and the Cray T3D(see [6] and references), which asks for N-ary Gray codes. This constitutes one reason why the topic of N-ary Gray codes is interesting. Moreover, if Q is a power of a prime, Q-ary Gray codes also have applications when determining the weight distribution of a linear code(see [24]). There are still many other types of Gray codes, depending on their applications(cf. e.g. [21, 22, 41]). For a more extended survey on Gray codes we refer to [61]. Of course, apart from their various widespread applications, Gray codes and their generalizations are also worthwhile to be studied in their own right, being nice mathematical structures with elegant properties.

Chapters 2, 3, and 4 of this thesis are dedicated to some special types of Gray codes. Chapter 2 mainly deals with the separability property of codes. In a Gray code, or in any ordered code, a question of theoretical as well as of practical relevance is the following. If two codewords in a code differ in m positions, how far are they separated from each other in the list of codewords? The larger this list distance in the code, the smaller the number of bit errors will be when transmitting codewords by means of analog signals (cf. [86]). In this chapter we derive the *separability function* of the *N-ary reflected Gray code*. We also introduce a slightly simpler construction for the related *cyclic N-ary Gray code* which was defined by Sharma and Khanna in [63]. The separability problem for this type of cyclic Gray codes is solved as well. Furthermore, we study a certain class of so-called *constant weight Gray codes*, and introduce a simple construction to obtain such codes, which appear to be subcodes of the aforementioned cyclic Gray codes with respect to the list order. The separability and the *index problem* or *ranking problem*(the relationship between a codeword and its index in the list) of all these codes are solved as well.
Moreover, we introduce in Chapter 2 a class of binary Gray codes having a separability capacity higher than that of the standard Gray codes, and we derive a formulism for its index system.

Chapter 3 focuses the discussion on the transition count spectra of Gray codes. *Balanced* and *exponentially balanced Gray codes* are major issues in this chapter. We develop a technique how to construct balanced Gray codes based on Bakos' Gray construction in [1]. By applying this technique, we can produce balanced Gray codes which can not be achieved by the methods of Robinson-Cohn [60], Bhat-Savage[3], or even by Bakos' method itself [1]. Furthermore, we prove a long standing conjecture of Wagner and West about the existence of exponentially balanced Gray codes.

More binary Gray codes with special properties are studied in Chapter 4. A class of Gray codes with *maximum crossover Hamming distance*(MCHD-*codes*) is constructed. This type of Gray codes is of considerable interest when dealing with communication systems using multi phase shift-keyed(MPSK) signals(see [41, 64]). The separability function and the index system of these codes are derived. Moreover, we introduce a method to construct Gray codes which induce *complete graphs*, and hence solve an open problem posed by Wilmer and Ernst in [89].

Chapter 6 is devoted to *lexicodes*, also known by the term *greedy codes*. Let V be a list of all vectors of $GF(q)^n$, lexicograpyically ordered with respect to some basis. Algorithms which search list V from top to bottom, any time selecting a codeword which satisfies some criterion, are called *greedy algorithms* and the resulting ordered set of codewords is called a *lexicode*. When $q = 2$, such a lexicode turns out to be linear for a wide variety of selection criteria. In this chapter we present a greedy algorithm for the construction of a large family of *linear q-ary lexicodes* which generalizes the algorithms of several other papers and puts these into a wider framework. By applying this new method, one can produce linear lexicodes which cannot be constructed by previous algorithms, because the characteristics or the underlying field of the codes do not meet the conditions of those algorithms.

Chapter 7 focuses on self-orthogonal ternary lexicodes. We derive some interesting properties of these lexicodes, and formulate a characterization for a self-orthogonal ternary lexicode which *cannot* be a subcode of a corresponding lexicode. Finally, we derive self-orthogonal $[n, 3, d]$-codes, with $d = 3(3k), 3(3k + 1)$ or $3(3k + 2)$, $k \geq 1$, which can be used as seed codes when applying a self-orthogonal greedy algorithm.

Samenvatting

De onderwerpen die in dit proefschrift aan de orde komen kunnen geclassificeerd worden als behorende tot de theorie van de geordende codes. Deze onderwerpen kunnen ruwweg worden onderverdeeld in twee klassen: Telrijen en Gray codes enerzijds en Lexicodes anderzijds. De bespreking van het eerste onderwerp zal als het eerste deel van dit proefschrift aangeduid worden. Het tweede deel van het proefschrift bevat het andere onderwerp.

In het algemeen is een *telrij* een lijst van (alle) 2^n binaire n-tallen(ofwel binaire woorden van lengte n). Telrijen worden toegepast in logische-circuit schakelingen. Twee speciale typen telrijen vormen het hoofdonderwerp van onze studie in Deel I. Het eerste type telrij heeft de eigenschap dat het aantal bits die veranderen als men van een codewoord naar de opvolger ervan overgaat, zo *groot* mogelijk is. In dit proefschrift wordt dit type telrij een *maximale telrij* genoemd. Een voorbeeld waar maximale telrijen worden toegepast is het testen van circuits voor betrouwbaar gedrag in het allerslechtste geval (zie bijv. [32, Exercise 67, p. 35]). Het tweede interessante type telrijen die we bestuderen, zijn de *uniforme telrijen*, d.w.z. telrijen die de eigenschap hebben dat alle paren opeenvolgende woorden dezelfde Hamming afstand hebben. Wat deze twee typen telrijen betreft, zijn de *gebalanceerde* telrijen van groot belang in combinatorische logische circuits. Maximale en uniforme telrijen worden voornamelijk in Hoofdstuk 5 besproken.

Binaire Gray codes, die een bijzonder soort uniforme telrij vormen, is een zeer bekend research onderwerp. In dit type verschillen elke twee opeenvogende codewoorden in precies één bit, d.w.z. hun Hamming afstand is 1. Van alle soorten Gray codes is de *binair gespiegelde Gray code*, ook bekend als de *standarard Gray code*, de meest bekende(zie [59, 86]). Deze code was een gepatenteerde uitvinding van Frank Gray in 1953, en werd gebruikt om het aantal coderingsfouten in een pulsecode systeem te verminderen. De code zelf werd echter al gedemonstreerd in 1878, als een toepassing in een telegraaf-installatie, door de Franse ingenieur Émile Baudot(zie [26]).

Ofschoon de binair gespiegelde Gray code een wijdverbreid toepassingsgebied heeft, zoals in de algebraische coderingstheorie (zie [84]), bij het ontwerpen van combinatorische algoritmen (zie [59]), en zelfs optimaal is bevonden met betrekking tot verscheidene andere toepassingen (zie [2]), zijn er soms Gray codes met extra eigenschappen vereist voor specifieke toepassingen. Bijvoorbeeld heeft men *gebalanceerde Gray codes* nodig bij het ontwerpen en testen van elektrische circuits en informatiesystemen(zie [3, 39, 41, 42, 87, 88]). Bovendien kan men toepassingen van de n-dimensionale kubus waarbij de hoekpunten in het N-tallig getalstelsel zijn uitge-

drukt, terugvinden in de ontwerpen van verscheidene parallelle computers waaronder de Ametek 2020, de J-machine, de Mosaic, de iWarp en de Cray T3D(zie [6] en de referenties aldaar), die N-aire Gray codes vereisen. Dit verklaart voor een flink deel het belang van deze N-aire codes. Stelt bovendien Q een priemmacht voor, dan kunnen Q-aire Gray codes toegepast worden bij het bepalen van de gewichtsverdeling van een lineaire code(zie [24]). Er zijn nog vele andere typen Gray codes, afhankelijk van hun toegepassingen. Voor een uitgebreider overzicht van Gray codes verwijzen we naar [61]. Natuurlijk zijn Gray codes, behalve om hun gevarieerde en wijdverspreide toepassingen, het bestuderen ook waard louter vanwege zichzelf, omdat ze mooie mathematische strukturen voorstellen met elegante eigenschappen.

De Hoofdstukken 2, 3, and 4 van dit proefschrift zijn gewijd aan enkele bijzondere typen Gray codes. Het grootste gedeelte van Hoofdstuk 2 gaat over de separabiliteits eigenschap van codes. Ten aanzien van een Gray code, of van elke andere geordende code, is de volgende vraag zowel van theoretisch alsook van praktisch belang. Wanneer twee codewoorden van een code in m posities verschillen, hoe ver liggen ze dan uit elkaar in de lijst van codewoorden? Hoe groter deze lijstafstand in de code, hoe kleiner het aantal bitfouten zal zijn wanneer men codewoorden door middel van analoogsignalen verstuurt(zie [86]). In dit hoofdstuk leiden we de *separabiliteitsfunctie* of voor de *N-aire gespiegelde Gray code*. We voeren ook een iets eenvoudiger constructie in voor de verwante cyclische N-aire Gray code die in [63] werd gedefinieerd door Sharma en Khanna. Tevens wordt het separabiliteitsprobleem voor dit type cyclische Gray codes opgelost.

Voorts bestuderen we een zekere klasse van zogenaamde *constant-gewicht Gray codes* en introduceren een eenvoudige constructie voor het verkrijgen van zulke codes, die subcodes blijken te zijn van de eerder genoemde Gray codes met betrekking tot de lijstorde. Het separabiliteitsprobleem (het verband tussen een codewoord en zijn index in de lijst) voor al deze codes wordt eveneens behandeld. Bovendien introduceren we in Hoofdstuk 2 een klasse van binaire Gray codes die een hogere separabiliteit hebben dan de standaard Gray codes en we leiden een formalisme af voor hun indexsysteem.

Hoofdstuk 3 concentreert zich op de overgangsspectra van Gray codes. *Gebalanceerde* en *exponentieel gebalanceerde Gray codes* zijn belangrijke onderwerpen in dit hoofdstuk. We ontwikkelen een methode om gebalanceerde Gray codes te construeren die gebaseerd is op de Gray constructie van Bakos in [1]. Met deze methode kunnen we Gray codes produceren die niet voortgebracht kunnen worden met de methoden van Robinson-Cohn [60], Bhat-Savage [3], en zelfs niet met de methode van Bakos zelf [1]. Verder bewijzen we een oud vermoeden van Wagner en West betreffende het bestaan van exponentieel gebalanceerde Gray codes.

Nog meer binaire Gray codes met speciale eigenschappen worden bestudeerd in Hoofdstuk 4. Er wordt een klasse van Gray codes geconstrueerd met *maximale diametrale Hamming afstand*(MCHD-*codes*). Dit type Gray codes in van groot belang wanneer men te doen heeft met communicatiesystemen die zogenaamde MPSK(multi phase shift-keyed) signalen gebruiken(zie [41, 64]). De separabiliteitsfunctie en het indexsysteem van deze codes worden bepaald. Bovendien introduceren we een methode voor de constructie van Gray codes die *volledige grafen* genereren en lossen daarmee

een open probleem op van Wilmer en Ernst in [89].

Hoofdstuk 6 is gewijd aan *lexicodes*, ook bekend onder de naam *gulzige codes*. Laat V een lijst zijn van alle vektoren van $GF(q)^n$ die lexicografisch is geordend ten opzichte van een of andere basis. Algoritmen die lijst V doorzoeken van boven naar beneden en die elke keer een codewoord selekteren dat aan een of ander kriterium voldoet, worden *gulzige algoritmen* genoemd en de resulterende geordende verzameling codewoorden heet een *lexicode*. Als $q = 2$, dan blijkt zo'n lexicode lineair te zijn voor een grote klasse van selectiecriteria. In dit hoofdstuk presenteren we een gulzige algoritme voor de constructie van een grote familie van *lineaire q-aire lexicodes* die een generalisatie is van de algoritmen uit diverse andere publicaties en die deze binnen een breder kader plaatst. Met behulp van deze nieuwe methode kan men lineaire lexicodes produceren die niet met vorige algoritmen geconstrueerd konden worden, omdat ofwel de karakteristieken van deze codes ofwel het getallenlichaan wat aan deze codes ten grondslag ligt, niet voldoen aan de voorwaarden van die constructies.

In Hoofdstuk 7 wordt de aandacht gevestigd op zelforthogonale ternaire lexicodes. We leiden een aantal interessante eigenschappen af voor deze lexicodes en we formuleren een karakterisering van een zelforthogonale ternaire lexicode die *niet* een subcode kan zijn van een bijbehorende lexicode. Tenslotte construeren we zelforthogonale $[n, 3, d]$-codes met $d = 3(3k), 3(3k+1)$ of $3(3k+2)$, $k \geq 1$, die gebruikt kunnen worden als basiscodes bij het toepassen van gulzige algoritmen voor de constructie van nieuwe zelforthogonale lexicodes.

Ringkasan

Topik-topik yang dibahas dalam thesis ini dapat digolongkan sebagai bagian dari teori tentang kode terurut. Secara garis besar topik-topik tersebut digolongkan dalam dua bagian: bagian pertama menyangkut barisan-barisan penghitung dan kode-kode Gray, sedangkan bagian lainnya mengenai *lexicodes*. Pembahasan topik pertama dikemas pada bagian pertama dari thesis ini, sedangkan bagian kedua dari thesis membahas topik sisanya.

Secara umum sebuah barisan penghitung adalah lis dari semua 2^n n-tuple biner (kata-kata biner dengan panjang n). Salah satu kegunaan dari barisan-barisan penghitung dapat ditemukan pada sirkuit-sirkuit logika. Dari berbagai barisan penghitung, pada thesis ini utamanya akan dibahas dua jenis barisan penghitung. Jenis yang pertama adalah barisan penghitung yang mempunyai sifat bahwa setiap dua kata berurutan di lis mempunyai perbedaan bit *semaksimal* mungkin. Pada thesis ini barisan penghitung jenis ini disebut *barisan penghitung maksimum*. Pengujian sirkuit-sirkuit fisik untuk kelakuan yang dapat dipercaya dalam kondisi-kondisi kasus terburuk adalah salah satu contoh dimana barisan penghitung maksimum itu digunakan(lihat misalnya [32, Exercise 67, p. 35]). Barisan penghitung lainnya dinamakan *barisan penghitung seragam*, yakni barisan penghitung yang bersifat bahwa semua pasangan dari kata-kata berurutan di lis mempunyai jarak Hamming sama. Diantara dua jenis barisan penghitung tersebut tadi, *barisan penghitung seimbang* layak dipertimbangkan karena kegunaannya dalam sirkuit-sirkuit logika kombinatorik. Pembahasan mengenai dua topik ini utamanya diletakan dalam Bab 5.

Kode Gray biner yang merupakan salah satu jenis khusus dari barisan penghitung seragam adalah topik yang sangat dikenal. Diantara kode-kode Gray, kode Gray tercermin biner, yang juga diketahui sebagai kode Gray baku, adalah yang paling dikenal(lihat [59, 86]). Kode ini dipatenkan oleh Frank Gray pada Tahun 1953, yang digunakan untuk menurunkan galat penyandian dalam sistim komunikasi kode berpulsa(lihat [26]). Akan tetapi kode itu sendiri telah didemonstrasikan penerapannya di Tahun 1878 pada peralatan telegraf oleh insinyur Perancis Émile Baudot(lihat [26]).

Walaupun kode Gray tercermin biner telah menemukan bidang terapannya secara luas, sebagai contoh pada teori kode aljabar(lihat [84]), pada perancangan dari algoritma-algoritma kombinatorik(lihat [59]), dan bahkan diketahui optimal berkaitan dengan berbagai aplikasi lainnya(lihat [2]), terkadang kode-kode Gray yang dilengkapi dengan sifat tambahan sangat dibutuhkan untuk terapan-terapan khusus. Sebagai contohnya, ketikan merancang percobaan-percobaan, atau ketika merancang dan menguji sirkuit-sirkuit listrik dan sistim-sistim informasi, kode-kode Gray *seimbang*

sangat dibutuhkan(lihat [3, 39, 41, 42, 87, 88]). Lebih jauh lagi, aplikasi dari *n-kubus N-er* dapat ditemukan pada rancangan dari beberapa komputer sistem beriringan (paralel) yang meliputi Ametek 2020, J-Machine, Mosaic, iWarp, dan Cray T3D(lihat [6] dan rujukannya). Ini merupakan alasan mengapa topik kode-kode Gray N-er menjadi menarik. Apabila Q adalah bilangan prima berpangkat, kode-kode Gray Q-er juga mempunyai terapan dalam menentukan distribusi bobot dari suatu kode linier(lihat [24]). Masih banyak lagi jenis dari kode Gray, bergantung dari bidang dimana dia digunakan(lihat e.g. [21, 22, 41]). Untuk yang tertarik mengetahui lebih mendalam tentang kode Gray kami rujuk pada [61]. Tapi terlepas dari berbagai jenis dan luasnya terapan dari kode Gray, kode Gray dan perumumannya sangat layak untuk dipelajari karena struktur matematikanya yang mengandung sifat-sifat yang menarik.

Bab 2, 3 dan 4 dari thesis ini diabdikan untuk membahas beberapa jenis khusus dari kode Gray. Bab 2 utamanya berkaitan dengan sifat keterpisahan dari kode. Pada sebuah kode Gray, atau pada sembarang kode terurut, pertanyaan yang bersifat teori maupun praktis yang relevan adalah sebagai berikut. Jika dua kata dalam sebuah kode berbeda pada m posisi, seberapa jauhkah mereka terpisah satu dengan lainnya di lis? Semakin jauh jarak lisnya, akan semakin kecil galat bitnya ketika kata-kata yang ditransmisikan menggunakan sinyal-sinyal analog(lihat [86]). Pada bab ini kita turunkan fungsi keterpisahan dari kode-kode Gray tercermin N-er. Kita juga perkenalkan suatu bangun yang sedikit lebih sederhana untuk mendapatkan *kode Gray N-er siklik* yang terkait dengan kode siklik yg dibangun oleh Sharma dan Khanna[63]. Masalah keterpisahan dari kode siklik jenis ini dipecahkan. Selanjutnya kita mempelajari sebuah kelas yg disebut *kode Gray bobot tetap*, dan memperkenalkan sebuah cara sederhana untuk mendapatkan kode tersebut. Dalam kaitannya dengan urutan, kode Gray bobot tetap yang diperoleh merupakan *kodebagian* dari kode siklik yang disebutkan sebelumnya. Masalah keterpisahan dan perankingan dari semua kode tersebut juga dipecahkan.
Selanjutnya, kita perkenalkan pada Bab 2 sebuah kelas dari kode-kode Gray biner dengan kapasitas pemisahnya lebih tinggi dari pada kapasitas yang dimiliki kode Gray baku, dan kita juga turunkan formulasi untuk sistim pengindekkannya.

Bab 3 berfokus pada pembahasan spektrum hitungan transisi dari kode-kode Gray. *Kode Gray seimbang* dan *kode Gray seimbang eksponen* merupakan isu mayor dari bab ini. Kita kembangkan sebuah tehnik bagaimana membangun kode-kode Gray seimbang berdasarkan pada tehniknya Bakos membangun kode Gray[1]. Dengan menggunakan tehnik itu kita dapat membangun kode Gray seimbang yang tidak bisa dihasilkan oleh metode Robinson-Cohn[60], Bhat-Savage[3], atau bahkan oleh metode Bakos sendiri[1]. Selebihnya, kita buktikan terkaan lama oleh Wagner dan West tentang adanya kode-kode Gray seimbang eksponen.

Kode-kode Gray biner lainnya yang mempunyai sifat-sifat khusus dibahas pada Bab 4. Sebuah kelas kode-kode Gray dengan *jarak Hamming bersebrangan maksimum* dibangun. Jenis kode-kode Gray ini menarik untuk dibicarakan terkait dengan sistem-sistem komunikasi yang menggunakan sinyal-sinyal multi fase shift-keyed(MPSK)(lihat [41, 64]). Sifat keterpisahan dan sistim pengindek dari kode-kode ini diturunkan. Lebih jauh, kita perkenalkan sebuah metode untuk membangun kode-kode Gray yang

menginduksi grafik-grafik lengkap, dan oleh karena itu memecahkan masalah terbuka yang disuguhkan oleh Wilmer dan Ernst[89].

Bab 6 diabdikan untuk pembahasan *lexicodes*, yang juga diketahui sebagai *kode-kode rakus*. Misalkan V sebuah lis dari semua vektor di $GF(q)^n$, yang terurut secara leksikografi sesuai dengan suatu basis. Algoritma yg mencari dari atas lis sampai ke bawah kata-kata yang memenuhi kriteria yang ditetapkan sebelumnya disebut *algoritma rakus*, dan himpunan kata yang dihasilkan disebut sebuah *lexicode*. Apabila $q = 2$, lexicode yang dihasilkan adalah linier untuk banyak jenis kriteria yang ditentukan. Pada bab ini kita sajikan algoritma rakus untuk membangun sebuah kelas yang luas dari *lexicodes q-er linier* yang memperumum algoritma-algoritma dari beberapa paper dan meletakkannya kedalam kerangka kerja yang lebih luas. Dengan menerapkan metode baru ini, seseorang dapat menghasilkan lexicodes yang tidak dapat dihasilkan menggunakan algoritma-algoritma sebelumnya, karena karakteristik atau medan yang disandari tidak memenuhi syarat-syarat algoritma-algoritma tersebut.

Bab 7 berfokus pada lexicodes self-orthogonal terner. Kita turunkan beberapa sifat menarik dari lexicodes ini, dan merumuskan sebuah karakterisasi untuk sebuah lexicode self-orthogonal terner sehingga tidak dapat merupakan kodebagian dari lexicode terkait. Lebih jauh, kita turunkan kode-kode self-orthogonal $[n, 3, d]$, dengan $d = 3(3k), d = 3(3k + 1)$ atau $d = 3(3k + 2)$, $1 \leq k$, yang dapat digunakan sebagai kode-kode inti ketika menerapkan suatu algoritma rakus self-orthogonal.

List of Publications

This thesis is based on the following publications which are written during the period from September 2002 until March 2006.

1. Suparta, I N. and A.J. van Zanten, "Balanced maximum counting sequences," to appear in *IEEE Trans. Inform. Theory.*
2. Suparta, I N., "A simple proof for the existence of exponentially balanced Gray codes," *The Elect. J. Combinatorics,* vol. 12, Paper N19, 2005.
3. Suparta, I N. and A.J. van Zanten, "A construction of Gray codes inducing complete graphs," submitted for publication to *Discrete Mathematics.*
4. Suparta, I N. and A.J. van Zanten, "On balanced uniform counting sequences", *Proceedings of the International Conference on Applied Mathematics(ICAM05)*, pp. 305-310, Bandung-Indonesia, Aug. 2005.
5. Suparta, I N. and A.J. van Zanten, *On balanced maximal counting sequences and balanced uniform counting sequences,* rept. CS 05-01, Institute for Knowledge and Agent Technology, University Maastricht, Maastricht, The Netherlands (2005).
6. Suparta, I N. and A.J. van Zanten, "On a class of Gray codes with maximum crossover Hamming distance," Proceeding of the Ninth International Workshop ACCT, Kranevo-Bulgaria, pp. 362-367, June 2004.
7. Suparta, I N. and A.J. van Zanten, "On cyclic N-ary Gray codes," *Proceedings of the SEAMS2003 conference,* Yogyakarta-Indonesia, July 2003.
8. Suparta, I N. and A.J. van Zanten, *Balanced Gray codes,* rept. CS 03-03, Institute for Knowledge and Agent Technology, Universiteit Maastricht, Maastricht, The Netherlands, March 2003.
9. Suparta, I N. and A. J. van Zanten, *On the list distance in cyclic N-ary Gray codes,* Rept. CS 02-01, Institute for Knowledge and Agent Technology, Universiteit Maastricht, Jan. 2002.
10. van Zanten, A.J. and I N. Suparta, "On the construction of linear q-ary lexicodes," *Designs, codes and Cryptography,* vol. 37, pp. 15-29, 2005.
11. van Zanten, A.J. and I N. Suparta, "Totally balanced and exponentially balanced Gray codes", *Discrete Analysis and Operation Research,* vol. 11, No. 4, pp. 81-98, 2004. (Russian Journal)
12. van Zanten, A.J. and I N. Suparta, *An algorithm for a large class of linear q-ary lexicodes,* rept. CS 03-01, Institute for Knowledge and Agent Technology, Universiteit Maastricht, Maastricht, The Netherlands, 2003.
13. van Zanten, A.J., and I N. Suparta, "The separability of standard cyclic N-ary Gray codes," *IEEE Trans. on Inform. Theory,* vol. 49, pp. 485-487, 2003.

14. van Zanten, A.J. and I N. Suparta, "The separability function of the standard cyclic N-ary Gray codes," *Proceeding of the Eight International workshop ACCT,* pp. 485-487, Tsarskoe Selo-Russia, Sept. 2002.

Curriculum Vitae

I Nengah Suparta was born in Buleleng regency, Bali - Indonesia, in 1965. After finishing his secondary school at Sekolah Menengah Atas Negeri 1 Amlapura in 1984, he continued his study at Mathematics Education Department, Udayana University, Denpasar - Indonesia. In 1998 he obtained his bachelor degree in Mathematics Education from this University. From 1988 till 1991 he taught Mathematics in a couple of secondary high schools. He was also a Mathematics tutor in a few private colleges to guide pupils who are preparing themselves for further study at University level.
In September 1991 he received a scholarship from DIKTI to attend a one-year pre-master program in Mathematics at ITB Bandung. Still funded by DIKTI, he started in September 1992 the master program at the same department and finished this program on December 12, 1994, with a M.Sc degree in Mathematics. His master thesis was titled *Integral Polyhedrons*. In 1995 he participated in a two-month workshop on coding theory at Gajah Mada University, Yogyakarta.
He followed an intensive course in French language at CCF Jakarta-Indonesia, from November 3, 1997 to May 14, 1998. Ho took a similar course at CUEF de Nancy - France, from June 11, 1998 to September 25, 1998. From October until November he attended a number of didactic-methodic courses at Lyon 1 University, Lyon-France. In November 27, 1998, he embarked on a PhD research program at Grenoble Institute of Applied Mathematics, Joseph Fourier, Grenoble - France. Unfortunately, because of important family reasons he had to quit this program on May 9, 1999.
After finishing a two-week course on coding theory in July 2001 at ITB Bandung, he was invited to visit the Department of Applied Mathematics, Delft University of Technology, The Netherlands, from November 18, 2001 to February 15, 2002. As a follow-up of this visit, he started his PhD research at the same department on September 11, 2002, under supervision of Prof. Dr. A.J. van Zanten. All results he obtained during the next four years form the content of his current thesis entitled "Counting Sequences, Gray Codes and Lexicodes".
Now he will return to the position which he occupied since November 1990, as a staff member of the Department of Mathematics Education, Singaraja Institute for Teachers Training and Education, Singaraja - Bali, Indonesia.

Index