

On perfect deletion-correcting codes

Andreas Klein

August 22, 2003

Abstract

In this article we present constructions for perfect deletion-correcting codes. The first construction uses perfect deletion-correcting codes without repetition of letters to construct other perfect deletion-correcting codes. This is a generalization of the construction shown in [1]. In the third section we investigate several constructions of perfect deletion-correcting codes using designs. In the last section we investigate perfect deletion-correcting codes containing few codewords.

1 Introduction

A word w' is said to be a subword of a word w , if w' can be obtained from w by deleting some letters from w . (The definition of "subword" is taken from [1]. We remark that sometimes "subword" is used for a subsequence of successive letters in mathematical literature, but not in the context of deletion-correcting codes.) In this article we want to reconstruct words from their subwords, i.e. we correct errors due to the deletion of some letters.

In the sequel we will use $A(v) := \{1, \dots, v\}$ as our alphabet. The set of all words of length k over $A(v)$ will be denoted by $A_k^*(v)$. Furthermore $A_k(v)$ will denote the set of all words of length k over $A(v)$ with different coordinates, i.e. if $x \in A_k(v)$ then $x_i \neq x_j$ for $i \neq j$. It is obvious that $A_k(v)$ is the empty set if $k > v$.

Definition 1

Let C (resp. C^*) be a subset of $A_k(v)$ (resp. $A_k^*(v)$) with the following property:

Every word of $A_{k-t}(v)$ (resp. $A_{k-t}^*(v)$) is subword of at most one word in C (resp. C^*).

Then C (resp. C^*) is said to be a t -deletion-correcting code.

If furthermore every word of $A_{k-t}(v)$ (resp. $A_{k-t}^*(v)$) is subword of exactly one word in C (resp. C^*), then C (resp. C^*) is said to be a perfect t -deletion-correcting code. We speak of a $T(k-t, k, v)$ (resp. $T^*(k-t, k, v)$) code.

Example 1

Consider the codes

$$\begin{aligned} C_1 &= \{(1, 2, 3), (3, 2, 1)\} \\ C_2 &= \{(1, 2, 3), (3, 2, 1), (1, 1, 1), (2, 2, 2), (3, 3, 3)\} \\ C_3 &= \{(1, 1, 2), (2, 2, 3), (1, 3, 3), (3, 2, 1)\} \end{aligned}$$

C_1 is a $T(2, 3, 3)$ code. C_2 and C_3 are $T^*(2, 3, 3)$ codes. This example proves that different $T^*(k-t, k, v)$ codes may be of different size. On the other hand it is easy to prove that the number of words in a $T(k-t, k, v)$ code is determined by its parameters. We find:

Each $T(k-t, k, v)$ code contains exactly

$$\frac{v}{k} \frac{(k-t)! \binom{v-1}{k-t-1}}{\binom{k-1}{k-t-1}}$$

different codewords (see Lemma 4.3 in [4]).

$T(t, k, v)$ codes are also known as directed designs. Other results on deletion-correction codes can be found in [6, 8].

This article is organized as follows: In the next section we construct $T^*(2, k, v')$ codes from $T(2, k, v)$ codes ($v' \leq v$). In the third section we investigate the construction of $T(t, k, v)$ codes from $T(t, k, k)$ codes. Finally we study perfect deletion-correcting codes with few codewords.

2 Construction of deletion-correcting codes from codes without repetition

In this section we show how to use $T(2, k, v)$ codes (e.g. codes without repetition) to construct perfect deletion-correcting codes over a smaller alphabet.

Lemma 1

Let C be a $T(2, k, v)$ code and let $r \leq v$. Let C_r be the set of all codewords in C with at least one coordinate in $\{v-r+1, \dots, v\}$. If there exists an injective mapping $f : C_r \rightarrow \{1, \dots, v-r\}$ where $f(c)$ is a coordinate of c then there exists a $T^*(2, k, v-r)$ code.

Proof

For each codeword $c \in C_r$ we delete all characters in $\{v-r+1, \dots, v\}$ and obtain a shorter codeword c' . Now we use c' to construct a codeword c'' of length k by repeating the character $f(c)$. (Example: $v = 4, r = 2, k = 4, c = (3, 2, 4, 1)$ and $f(c) = 1$. Then $c' = (2, 1)$ and $c'' = (2, 1, 1, 1)$.)

The code C^* contains

1. all words of type c'' with $c \in C_r$,
2. all words in $C \setminus C_r$ and

3. all words of type (c, c, \dots, c) with $c \in A(v-r)$ and c is not in the image of f .

We claim that C^* is a $T^*(2, k, v-r)$ code.

Since each word in $A_2(v)$ is a subword of exactly one word in C , we find that each word in $A_2(v-r)$ is a subword in exactly one word in C^* . The word (z, z) with $z \in A(v-r)$ is either a subword of c'' (if $f(c) = z$) or it is a subword of (z, z, \dots, z) .

This proves that every word of length 2 is a subword of exactly one word in C^* , i.e. C^* is a $T^*(2, k, v-r)$ code. \square

The next theorem is a generalization of the corresponding theorems in [1].

Theorem 1

Assume $k \geq 2r+1$ and a $T(2, k, v)$ code exists. Then there exists a $T^*(2, k, v-r)$ code.

Proof

We will use Lemma 1. Therefore we have to construct the mapping f .

Let $X \subset C_r$ and let X_f be the set of all characters in $\{1, \dots, v-r\}$ which are contained in at least one codeword of X . We are going to prove $|X_f| \geq |X|$.

For each character $z \in \{1, \dots, v-r\}$ and each codeword $c \in C_r$ we define the weight $w(z, c)$ by:

$$w(z, c) = \begin{cases} 0 & \text{if } z \text{ is not contained in } c. \\ \frac{a(c)}{2r} & \text{if } z \text{ is contained in } c \text{ and } a(c) \text{ is the number of characters} \\ & \text{in } \{v-r+1, \dots, v\} \text{ that are contained in } c. \end{cases}$$

Since all pairs of the form (z, x) and (x, z) with $x \in \{v-r+1, \dots, v\}$ are subwords of exactly one word in C_r , we find

$$\sum_{c \in C_r} w(z, c) = 1$$

for all $z \in \{1, \dots, v-r\}$.

Now we have:

$$\begin{aligned} |X_f| &\geq \sum_{c \in X} \sum_{z \in \{1, \dots, v-r\}} w(z, c) \\ &= \sum_{c \in X} (k - a(c)) \frac{a(c)}{2r} \\ &\geq |X| \frac{k-1}{2r} \geq |X| \end{aligned}$$

This proves $|X_f| \geq |X|$ for all $X \subset C_r$. With the wedding theorem (Hall 1954) this proves that an injective mapping f with the properties needed for the application of Lemma 1 exists. \square

In [1] this theorem was only proven for $r \leq 3$.

3 Construction from designs

In this section we will use combinatorial designs and linear spaces for the construction of deletion-correcting codes.

First we recall the definitions:

Definition 2

A $t - (v, k, 1)$ design is a family of k -subsets (called blocks) of $\{1, \dots, v\}$, with each t -subset of $\{1, \dots, v\}$ is contained in exactly one block.

A linear space is a family of subsets of $\{1, \dots, v\}$ (called lines) where every 2-subset is contained in exactly one line.

The next theorem shows how to use designs for the construction of deletion-correcting codes. (This is a special cases of Theorem 4.3 in [4], but the proof given here is much simpler.)

Theorem 2

If a $t - (v, k, 1)$ design and a $T(t, k, k)$ code exist, then a $T(t, k, v)$ code exists.

Proof

We replace every block by the elements of a $T(t, k, k)$ code over the corresponding alphabet. Now we find:

For every t -tuple of characters there exists exactly one block in the $t - (v, k, 1)$ design that contains these t characters. Since this block was replaced by a $T(t, k, k)$ code, we find exactly one codeword that contains the t characters in the given order. \square

This theorem shows that $T(t, k, k)$ codes are of special interest. In the following we will recall what is known about $T(t, k, k)$ codes.

Up to isomorphy, $\{(1, 2, \dots, k), (k, k - 1, \dots, 1)\}$ is the only $T(2, k, k)$ code. Levenstein [4] proves that a $T(t, t + 1, t + 1)$ code exists for each t . Mathon and van Trung [5] construct a $T(4, 6, 6)$ code and prove by exhaustive search that a $T(4, 7, 7)$ code does not exist.

At this point give a combinatoric prove for the non-existence of a $T(4, 7, 7)$ code without the use of computer calculations.

Theorem 3

A $T(4, 7, 7)$ code does not exist.

Proof

Assume C is a $T(4, 7, 7)$ code. Without loss of generality we can assume that $(1, 2, 3, 4, 5, 7)$ is a codeword of C .

If $(1, 3, 2, 4)$ is a subword of a codeword c , we can conclude that the characters 5, 6 and 7 must stand before the character 4. (Otherwise c and $(1, 2, 3, 4, 5, 6, 7)$ would contain a subword of type $(1, 3, 4, x)$.) Corresponding to the above argument we conclude that characters 5, 6 and 7 must stand before the character 4 in each codeword that contains $(2, 1, 3, 4)$, $(3, 1, 2, 4)$, $(2, 3, 1, 4)$, $(1, 2, 4, 3)$, $(1, 3, 4, 2)$ or $(2, 3, 4, 1)$ as a subword.

Thus there exist 7 codewords that contain the characters 5, 6 and 7 before the character 4. But there are only 6 permutations of 5, 6, 7, i.e. two codewords must share a four letter subword of type $(x, y, z, 4)$ with $\{x, y, z\} = \{5, 6, 7\}$. \square

Moreover we know that, for $k > \binom{t+1}{2}$, no $T(t, k, k)$ code exists (see [5] Theorem 2.1). For the special case $t = 5$ and $t = 6$ Mathom and van Trung found better bounds by exhaustive search.

Now we come to applications of Theorem 2.

1. For each $2 - (v, k, 1)$ design, there exists a $T(2, k, v)$ code.
2. For each $k \equiv 2, 4 \pmod{6}$ we can find a $T(3, 4, k)$ code (For all such k a $3 - (k, 4, 1)$ designs exist see [2]).
3. Application of Theorem 2 to the small Witt design [7] reveals a $T(5, 6, 12)$ code.
4. Other examples for the application of Theorem 2 are: There exists a $T(4, 5, 11)$, a $T(4, 5, 15)$, a $T(4, 5, 17)$, a $T(4, 5, 23)$, a $T(4, 5, 27)$, a $T(5, 6, 16)$, a $T(5, 6, 18)$, a $T(5, 6, 24)$, a $T(5, 6, 28)$, a $T(6, 7, 17)$, a $T(7, 8, 18)$, a $T(8, 9, 19)$ and a $T(9, 10, 20)$ code (see [3] and the references given there).

The next theorem can be viewed as a combination of Theorem 2 and Lemma 1.

Theorem 4

Let L be linear space with v points. Let m be the maximal number of points on a line of L .

Assume that two injective mappings f_1, f_2 exist which map each line with less than m points onto a point on that line. Further assume that the images of f_1 and f_2 are disjoint. Then a $T^*(2, m, v)$ code exists.

Proof

For each line with m points $\{P_1, \dots, P_m\}$ the code contains the words (P_1, \dots, P_m) and (P_m, \dots, P_1) .

For each other line l with the points $\{Q_1, \dots, Q_n\}$ let $f_1(l) = Q_i$ and $f_2(l) = Q_j$. Then the code contains the words $(Q_1, \dots, Q_i, Q_i, \dots, Q_i, Q_{i+1}, \dots, Q_n)$ and $(Q_n, \dots, Q_j, Q_j, \dots, Q_j, Q_{j-1}, \dots, Q_1)$. (The words contains $m - n + 1$ repetitions of Q_i respectively Q_j .)

Since L is a linear space, each word (P, P') with $P \neq P'$ is a subword of exactly one codeword. Since f_1 and f_2 are injective and have disjoint images, we find that each word of the form (P, P) is the subword of at most one codeword.

Now we can extend the code to a perfect deletion-correcting code, by adding words of the form (P, \dots, P) . \square

Examples for the application of Theorem 4 are:

1. For $k = q + 1$ and $q^2 + \frac{2}{3}q + 1 \leq v \leq q^2 + q + 1$ a $T^*(2, k, v)$ code exists. (Let L be a projective plane where $q^2 + q + 1 - v$ points on a line are deleted.)
2. For $k = q$ and $q^2 - \frac{1}{3}q \leq v \leq q^2$ a $T^*(2, k, v)$ code exists. (Application of Theorem 4 for an affine plane.)

4 Codes with few codewords

If $tv \leq k$, then the code that contains only the codeword $(1, 2, \dots, v, 1, 2, \dots, v, 1, 2, \dots)$ is a perfect deletion-correcting code with parameters t, k, v , since the sequence $1, \dots, v$ is repeated at least t times.

The proof that these are the only parameters for which a perfect deletion-correcting code with only one codeword exists, is shown in the next theorem.

Theorem 5

A $T^*(t, k, v)$ code which contains only one codeword exists if and only if $tv \leq k$.

Proof

We already know that for $tv \leq k$ a $T^*(t, k, v)$ code which contains only one codeword exists. This proves one direction of the theorem. For the proof of the other direction we assume that a $T^*(t, k, v)$ code exists which contains only one codeword c .

We can find a character c_1 such that the first occurrence of c_1 in c is at position $p_1 \geq v$. Analogously we can find a character c_2 such that the first occurrence of c_2 in c after position p_1 is at position p_2 with $p_2 - p_1 \geq v$. Repeating this argument we find characters c_3, \dots, c_t at positions p_3, \dots, p_t with $p_{i+1} - p_i \geq v$.

Thus c contains at least $p_t = (p_t - p_{t-1}) + \dots + (p_2 - p_1) + p_1 \geq tv$ points. Thus $k \geq tv$. \square

Now we investigate codes with $k < tv$.

Theorem 6

For each $t, v \in \mathbb{N}$ a $T^*(t + 1, tv, v)$ code exists.

Proof

We construct a code with $v + 2$ codewords.

The first v codewords are the words which contain no different characters. The remaining two codewords have the form

$$c = (1, \dots, 1, 2, \dots, 2, \dots, v, \dots, v)$$

where each character is repeated t times and

$$c' = (v, v - 1, \dots, 1, v, v - 1, \dots, 1, \dots, v, v - 1, \dots, 1)$$

where the sequence $v, v - 1, \dots, 1$ is repeated t times.

That this code is a $T^*(t + 1, tv, v)$ code can be seen as follows:

Each subword of length $t + 1$ of c respectively c' contains at least two different characters. Thus words of length $t + 1$ that contain no different characters are subwords of exactly one codeword.

Each subword of c has monotone increasing characters, but each subword of length $t + 1$ of c' contains at least two adjacent characters a, b with $a > b$. This guarantees that the code is a $t + 1$ deletion-correcting code.

It is now easy to check that indeed each word of length $t + 1$ is a subword of a codeword, i.e. the code is perfect. \square

References

- [1] P. A. H. Bours. On the Construction of Perfect Deletion-Correcting Codes using Design Theory. *Designs, Codes and Cryptography*, 6:5–20, 1995.
- [2] H. Hanani. On quadruple systems. *Can. J. Math*, 12:145–157, 1960.
- [3] D. L. Kreher. t -Designs, $t \geq 3$. In J. H. Dinitz C. J. Colbourn, editor, *Handbook of Combinatorial Designs*, volume 4 of *Discrete Mathematics and Its Applications*, chapter 1.3, pages 47–66. CRC Press, 1996.
- [4] V. I. Levenstein. On Perfect Codes in Deletion/Insertion Metric. *Discrete Math. Appl.*, 2(3):241–258, 1992. Translation from *Discretnaya Matematika*, 3(2):3-20, 1992.
- [5] R. Mathon and T. van Trung. Directed t -Packings and Directed t -Steiner Systems. *Designs, Codes and Cryptography*, 18:187–198, 1999.
- [6] N. Shalaby, J. Wang, and J. Yin. Existence of perfect 4-deletion-correcting codes with length six. *Designs, Codes and Cryptography*, 23:99–110, 2001.
- [7] E. Witt. Über Steinersche Systeme. *Abh. Math. Sem. Hamburg*, 12:265–275, 1938.
- [8] J. Yin. A combinatorial construction for perfect deletion-correcting codes. *Designs, Codes and Cryptography*, 27:145–156, 2002.

Andreas Klein
Universität Kassel
Fachbereich 17 (Mathematik und Informatik)
D-34109 Kassel
klein@mathematik.uni-kassel.de