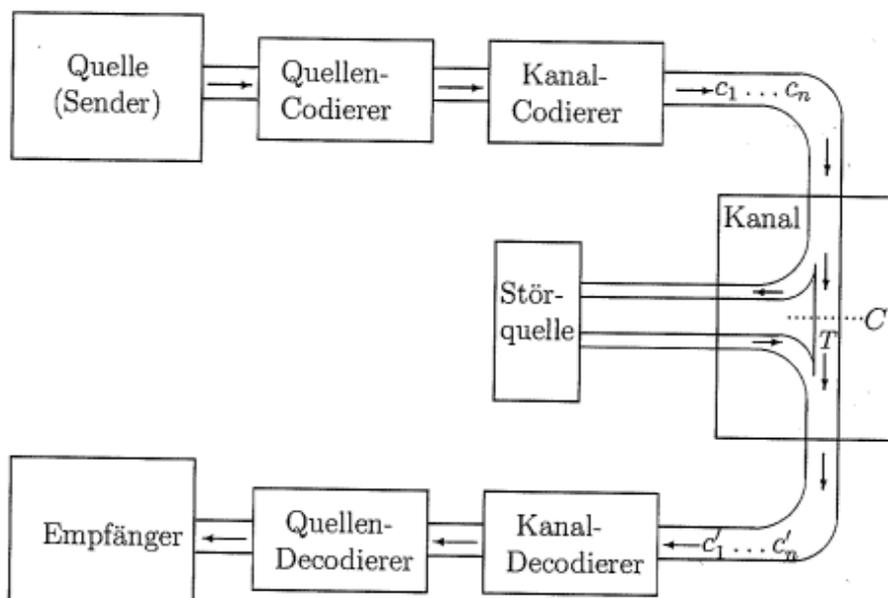


## E: Anwendungsbeispiele in der Codierungstheorie

### E 1.0 Schema eines Nachrichtenübertragungssystems

Das Schema eines Nachrichtenübertragungssystems zeigt Figur E 1:



Figur E.1: Schema eines Nachrichtenübertragungssystems mit gestörtem Kanal

Wir beschreiben zunächst ein Codier- und ein Decodierverfahren. Dass dieses eine Korrektur von Übertragungsfehlern gestattet, erläutern wir dann im Anschluss.

### 1.1 Definition Wort

In der Codierungstheorie schreitet man ein  $m$ -Tupel  $(\alpha_1, \dots, \alpha_m) \in A^m$  (mit  $m \in \mathbb{N}$ ) über der Menge  $A$  meist ohne Klammern und Kommata und nennt es **Wort** (Plural: Wörter, nicht Worte) der Länge  $m$  über dem **Alphabet**  $A$ . Nach der Quellencodierung seien die Nachrichten Wörter der Länge  $k$  über einem Körper  $K$ , also von der Form

$$\alpha_1 \dots \alpha_k := (\alpha_1, \dots, \alpha_k) \in K^k.$$

### 1.2 Lineare Codierung

Unter einer **linearen Codierung** dieser Nachrichten versteht man die durch Multiplikation mit einer Matrix  $G \in K^{(k,n)}$  vom Rang  $k$  gegebene lineare Abbildung:

$$c_G : \begin{cases} K^k & \longrightarrow K^n \\ (\alpha_1, \dots, \alpha_k) & \mapsto (\alpha_1, \dots, \alpha_k) \cdot G = \sum_{i=1}^k \alpha_i g_{i\bullet} \end{cases}$$

(Hierbei bezeichnet  $g_{i\bullet}$  die  $i$ -te Zeile von  $G$ .)

### 1.3 Beispiel

Seien  $K = \text{GF}(2) = (\{0, 1\}, \oplus, \odot) = \mathbb{Z}_2$ , ferner  $k = 4, n = 7$  und

$$G_1 := \left( \begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array} \right) \in \text{GF}(2)^{(4,7)}.$$

Die Nachricht 1100 wird dann codiert zu

$$1100 \cdot G_1 = g_{1\bullet} \oplus g_{2\bullet} = 1100010.$$

Da  $G_1$  als linke  $4 \times 4$ -Untermatrix die  $4 \times 4$ - Einheitsmatrix  $I_4$  hat, ist beim Codewort

$$c_G(\alpha_1 \alpha_2 \alpha_3 \alpha_4) = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_1 + \alpha_2 + \alpha_3, \alpha_2 + \alpha_3 + \alpha_4, \alpha_1 + \alpha_2 + \alpha_4)$$

das Anfangsstück (*Präfix*) der Länge 4 gleich der ursprünglichen Nachricht. (Zu diesen **Informationsbits** kommen noch drei **Kontrollbits**.) Eine solche Codierung heißt **systematische Codierung**.

## 1.4 Generatormatrix eines linearen Codes

Der Code  $\mathcal{C}$ , also die Menge der Codewörter

$$\text{Bild } c_G = \{ (\alpha_1, \dots, \alpha_k) \cdot G \mid \alpha_1, \dots, \alpha_k \in K \} \subseteq K^n$$

ist ein Unterraum von  $K^n$ . Daher heißt  $\mathcal{C}$  **linearer Code** oder **Linearcode**. Die Bilder der Einheitsvektoren von  $K^k$  unter  $c_G$  sind die Zeilen von  $G$ ; diese erzeugen daher  $\mathcal{C}$ :

$$\mathcal{C} = \langle g_{1*}, \dots, g_{k*} \rangle.$$

$G$  heißt dementsprechend eine **Generatormatrix**, oft auch **Basismatrix** von  $\mathcal{C}$ .

**Beispiel (Fortsetzung):**

Der Code  $\mathcal{C}_1$  mit Generatormatrix  $G_1$  hat Dimension 4 und Wörter der Länge 7. Er heißt **(7, 4)-Hammingcode** (nach R.W. Hamming, dem Erfinder einer ganzen Serie von Codes).

*Richard Wesley Hamming  
1915 - 1998 amerik. Mathematiker*

## 1.5 Kontrollmatrix eines linearen Codes

$\mathcal{C}$  kann (wie jeder Unterraum von  $K^n$ ) als Lösungsraum eines homogenen linearen Gleichungssystems dargestellt werden:

$$c \in \mathcal{C} \Leftrightarrow H \cdot c^T = 0.$$

Dazu sucht man als Zeilen der Koeffizientenmatrix  $H$ , der sogenannten **Kontrollmatrix**,  $n - k$  linear unabhängige Vektoren  $h_1, \dots, h_{n-k}$ , die auf  $\mathcal{C}$  senkrecht stehen, für die also gilt

$$h_i \cdot g_j^T = 0 \quad (i = 1, \dots, n - k, j = 1, \dots, k).$$

Wegen der Linearität der Abbildung

$$S_H: \begin{cases} K^n & \rightarrow K^{(n-k,1)} \\ v & \mapsto H \cdot v^T \end{cases}$$

wird durch  $S_H$  dann jeder Codevektor (als Linearkombination von  $\{g_{1*}, \dots, g_{k*}\}$ ) annulliert. Aus Dimensionsgründen gilt sogar

$$\mathcal{C} = \text{Kern } S_H.$$

**Spezialfall:** Ist  $G = (I_k \mid B)$ , so kann man  $H = (-B^T \mid I_{n-k})$  wählen.

**Beispiel (Fortsetzung)**

Ist  $G = G_1$ , so wählen wir z.B. die  $(3 \times 7)$ -Matrix

$$H_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

$\underbrace{\hspace{10em}}_{B_1^T}$

Für diese gilt insbesondere  $H_1 \cdot G_1^T = 0$  und  $H_1 \cdot c^T = 0$  für jedes  $c \in \mathcal{C}_1$ .  
Z.B. folgt:

$$H_1 \cdot (1, 1, 0, 0, 0, 1, 0)^T = h_{\bullet 1} \oplus h_{\bullet 2} \oplus h_{\bullet 6} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

## 1.6 Fehlerkorrektur

Nach der Übertragung über einen Kanal seien die

|                      |     |   |            |     |            |
|----------------------|-----|---|------------|-----|------------|
| empfangene Nachricht | $y$ | = | $\eta_1$   | ... | $\eta_n$   |
| gesendete Nachricht  | $c$ | = | $\gamma_1$ | ... | $\gamma_n$ |

und der sogenannte **Fehlervektor**  $e := y - c = \eta_1 - \gamma_1 \dots \eta_n - \gamma_n$ .

Wieder wegen der Linearität von  $S_H$  und wegen  $S_H(c) = 0$  gilt:

$$S_H(y) = S_H(c + e) = S_H(c) + S_H(e) = S_H(e).$$

Der Vektor  $S_H(y)$  ist daher unabhängig von der gesendeten Nachricht  $c$  und nur vom Fehlervektor abhängig; er heißt **Syndrom** von  $y$ . Die Abbildung  $S_H$  wird daher **Syndrom-Abbildung** genannt.

Sind nun die Syndrome  $S_H(e_1), \dots, S_H(e_n)$ , d.h. die Spalten von  $H$  paarweise verschieden, so kann man unter der Voraussetzung, dass ein Fehler in nur einer Komponente von  $c$  vorgekommen ist, diese Komponente bestimmen und im Falle  $K = \text{GF}(2)$  korrigieren. Ein solcher Code heißt 1-fehlererkennend bzw. **1-fehlerkorrigierend**. (Sind allerdings in einem Codewort mehrere Stellen verfälscht worden, so kann die Fehlerkorrektur sogar deren Anzahl vergrößern.)

**Beispiel: (7, 4)–Hammingcode** (Fortsetzung):

In  $H_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$  sind die 7 Spalten paarweise verschieden. Daher

ist der (7, 4)–Hammingcode  $\mathcal{C}_1 = \text{Kern } S_{H_1}$  1-fehlerkorrigierend.

Ist etwa das übertragene Codewort  $c_1 = 1100010$  zu  $y_1 = 1101010$  verfälscht, so liefert  $y_1$  das Syndrom

$$\begin{aligned} S_{H_1}(y_1) &= H_1 \cdot y_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = h_{\bullet 1} \oplus h_{\bullet 2} \oplus h_{\bullet 4} \oplus h_{\bullet 6} \\ &= \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = h_{\bullet 4}; \end{aligned}$$

## Der Sequenzraum: Codes und Kugelpackungen

Wir betrachten im Folgenden Codes fester Länge  $n$ , sogenannte **Blockcodes**, also Teilmengen von  $A^n$ . Wir führen einen *Abstandsbegriff* ein:

### E1.7 Hamming-Abstand, Sequenzraum

(a) **Definition: Hamming-Abstand**

Seien  $\mathbf{a} = a_1 \dots a_n$  und  $\mathbf{b} = b_1 \dots b_n$  Wörter der Länge  $n$  über dem Alphabet  $A$ ; dann bezeichnet  $d_H(\mathbf{a}, \mathbf{b})$  die Anzahl der unterschiedlichen Komponenten von  $\mathbf{a}$  und  $\mathbf{b}$ . Diese Zahl  $d_H(\mathbf{a}, \mathbf{b}) = |\{i \mid a_i \neq b_i\}|$  heißt *Hamming-Abstand (Hamming-Distanz)* zwischen  $\mathbf{a}$  und  $\mathbf{b}$ . (Wir schreiben oft nur  $d$  statt  $d_H$ ).

**Beispiel:**

$$\begin{aligned}d(00010, 00011) &= 1 = d(00010, 00000). \\d(00011, 00000) &= 2 = d(10001, 00000).\end{aligned}$$

(b) *Eigenschaft:* Der Hamming-Abstand  $d_H : A^n \times A^n \rightarrow \mathbb{N}_0$  ist eine *Metrik* auf  $A^n$ ,

d.h. es gilt (für alle  $\mathbf{x}, \mathbf{y}, \mathbf{z} \in A^n$ ):

- (i)  $d(\mathbf{x}, \mathbf{y}) \geq 0$  (Positivität)    (ii)  $d(\mathbf{x}, \mathbf{y}) = 0 \implies \mathbf{x} = \mathbf{y}$
- (iii)  $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$  (Symmetrie)
- (iv)  $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{z})$  (Dreiecksungleichung) .

**Beweisskizze**

(i), (ii) und (iii) sind offensichtlich: Die Dreiecksungleichung (iv) sieht man folgendermaßen ein: Ist  $i$  die Nummer einer Komponente, in der sich  $\mathbf{x}$  und  $\mathbf{z}$  unterscheiden ( $x_i \neq z_i$ ), so gilt mindestens eine der Beziehungen  $x_i \neq y_i$  oder  $y_i \neq z_i$ . Eine Komponente, die einen Beitrag 1 zu  $d(\mathbf{x}, \mathbf{z})$  liefert, gibt einen solchen auch für  $d(\mathbf{x}, \mathbf{y})$  oder für  $d(\mathbf{y}, \mathbf{z})$ .  $\square$

(c) **Definition: Sequenzraum**

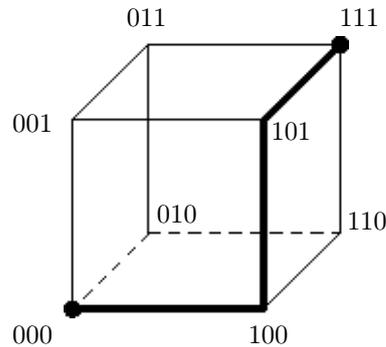
Der metrische Raum  $(A^n, d_H)$ , bestehend aus der Menge aller Wörter der Länge  $n$  über  $A$  und dem Abstand  $d_H$ , heißt *Sequenzraum*. Ist speziell  $A = \{0, 1\}$ , so spricht man auch vom  $n$ -dimensionalen **Einheitswürfel**.

(Der Sequenzraum spielt u.a. auch in der Evolutionstheorie eine Rolle; die Hamming-Distanz zwischen zwei Genen ist dort die minimale Anzahl der Punktmutationen, die nötig ist, um das Muster eines der beiden Gene in das des anderen umzuwandeln.)

### E 1.8 Beispiel:

$(\{0, 1\}^3, d_H)$  lässt sich mittels eines Würfels der Kantenlänge 1 im 3-dim reellen Raum darstellen; die Wörter von  $\{0, 1\}^3$  sind den Ecken des Würfels zugeordnet; der Abstand

zweier Wörter ist die Länge des kürzesten Weges zwischen den zugeordneten Ecken längs einer Würfelkante (s. Figur E.2).



Figur E.2: Darstellung des Sequenzraums  $(\{0, 1\}^3, d_H)$ .  
 (Markiert ist ein kürzester Weg von 000 zu 111;  
 seine Länge ist 3 – entsprechend dem Hamming-Abstand.)

### Aufgabe E.1

Zeigen Sie: Zu gegebenem Wort  $\mathbf{z} \in A^n$  gibt es  $\binom{n}{m}(|A| - 1)^m$  Wörter in  $A^n$ , die Abstand  $m$  von  $\mathbf{z}$  haben (für  $0 \leq m \leq n$ ).

Das Begehen von  $t$  Fehlern bei Übermittlung eines Codeworts  $\mathbf{c}$  führt zu einem Wort  $\mathbf{x}$  des Sequenzraums, das sich von  $\mathbf{c}$  in  $t$  Komponenten unterscheidet. Dieser Sachverhalt legt es nahe, Sphären oder Kugeln um Codewörter zu betrachten; dabei sind Kugeln wie in metrischen Räumen üblich definiert:

#### E 1.9 Definition: Kugel

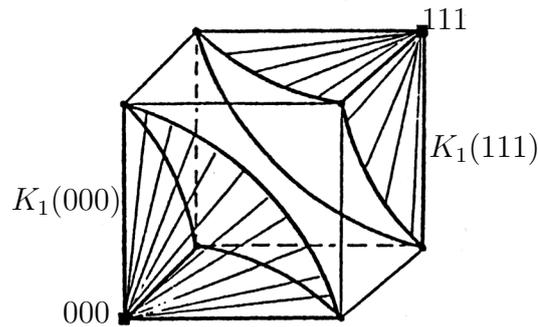
Ist  $\mathbf{z} \in A^n$  und  $t \in \mathbb{N} \cup \{0\}$ , dann heißt

$$K_t(\mathbf{z}) := \{\mathbf{x} \in A^n \mid d_H(\mathbf{x}, \mathbf{z}) \leq t\}$$

*Kugel vom Radius  $t$  um den Mittelpunkt  $\mathbf{z}$ .*

#### E 1.10 Beispiel (Fortsetzung):

Figur E.3 skizziert die Kugeln  $K_1(000) = \{000, 100, 010, 001\}$  und  $K_1(111) = \{111, 011, 101, 110\}$  vom Radius 1 in  $(\{0, 1\}^3, d_H)$ .



Figur E.3: Zwei Kugeln vom Radius 1 in  $(\{0, 1\}^3, d_H)$ .

### E 1.11 Bemerkungen: Kugeln und Fehlerkorrektur

- (i) Die Elemente von  $A^n$ , die aus  $\mathbf{z} \in A^n$  durch Fehler in bis zu  $t$  Komponenten entstehen, liegen in  $K_t(\mathbf{z})$ . Umgekehrt kann jedes Wort von  $K_t(\mathbf{z})$  aus  $\mathbf{z}$  durch  $t$  oder weniger Fehler entstanden sein.
- (ii) Ist  $\mathbf{b} \in K_t(\mathbf{z}_1) \cap K_t(\mathbf{z}_2)$ , so kann  $\mathbf{b}$  durch bis zu  $t$  Fehler aus  $\mathbf{z}_1$  oder aus  $\mathbf{z}_2$  entstanden sein.
- (iii) Ist  $\mathcal{C}$  ein Blockcode mit der Eigenschaft

$$(*) \quad K_t(\mathbf{c}_i) \cap K_t(\mathbf{c}_j) = \emptyset \quad \text{für alle } \mathbf{c}_i, \mathbf{c}_j \in \mathcal{C} \text{ mit } \mathbf{c}_i \neq \mathbf{c}_j,$$

so lassen sich (durch Decodierung zum Mittelpunkt derjenigen Kugel, in der das fehlerhafte Wort liegt) Wörter mit bis zu  $t$  Fehlern korrigieren.

- (iv) Ein einfaches *Decodierschema* besteht dabei aus der Auflistung der Kugeln um Codewörter und ihrer Elemente mit der Vorschrift, zum Kugelmittelpunkt hin zu decodieren. Alle nicht erfassten Wörter gehören zum *Decodierungsausfall*, sind nicht decodierbar und werden als "fehlerhaft" gemeldet.

Durch (10.5)(ii) und (iii) wird die folgende Definition motiviert:

### E 1.12 Definition: Fehlerkorrigierender Code

Ein Code  $\mathcal{C}$ , der die Eigenschaft  $(*)$  von E 1.11(iii) besitzt, heißt **t-fehlerkorrigierender Code**.

Er hat die Eigenschaft, dass bei ihm fehlerhafte Wörter mit Fehlern in  $m$  Komponenten für  $m \leq t$  auf die in E 1.11 (iii) angegebene Weise richtig decodiert werden können.

Jeder  $t$ -fehlerkorrigierende Code ist also auch  $(t - 1)$ -fehlerkorrigierend,  $(t - 2)$ -fehlerkorrigierend, usw..

**Beispiel** (Fortsetzung):

$\mathcal{C}_1 = \{000, 111\}$  ist nach E 1.10 ein 1-fehlerkorrigierender Code. Die fehlerhaften Wörter (in Figur E.3 durch  $\bullet$  markiert) werden zum Kugelmittelpunkt (mit  $\blacksquare$  markiert) korrigiert.

### E 1.13 Kugelpackungen

Es ist also in unserem Zusammenhang sinnvoll, nach einer disjunkten Überdeckung von  $A^n$  ( bzw. der Überdeckung einer Teilmenge von  $A^n$ ) durch Kugeln vom Radius  $t$  zu fragen, nach sogenannten *Kugelpackungen*. Die Kugel-Mittelpunkte können dann als Elemente eines Codes  $\mathcal{C}$  gewählt werden, der  $t$ -fehlerkorrigierend ist. (Vgl. Figur E.4 !)

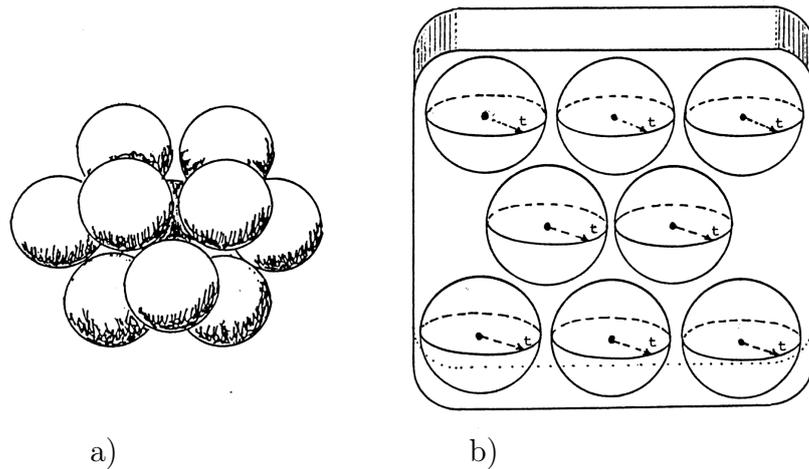


Bild E.4: Kugelpackung a) modellhaft b) schematisch

Wir behandeln *Schranken* für die Möglichkeit, solche Packungen zu finden. Dazu betrachten wir zunächst die Mächtigkeit einer Kugel und dann die Anzahl der durch die Kugeln abgedeckten Wörter.

### E 1.14 Anzahlen

#### (a) Kugel-Mächtigkeit

Für eine Kugel vom Radius  $t$  in  $A^n$  gilt (mit  $|A| = q$  und  $\mathbf{c} \in \mathcal{C}$ ):

$$|K_t(\mathbf{c})| = \sum_{i=0}^t \binom{n}{i} (q-1)^i$$

#### (b) Kugelpackungs-Schranke (Hamming-Volumenschranke)

Existiert ein  $t$ -fehlerkorrigierender Code  $\mathcal{C}$  in  $A^n$ , so gilt folgende Ungleichung (mit  $|A| = q$ ):

$$(**) \quad \sum_{i=0}^t \binom{n}{i} (q-1)^i |\mathcal{C}| \leq q^n.$$

**Beweis:**

- (a) Nach Aufgabe E.1 gibt es  $\binom{n}{i}(q-1)^i$  Wörter vom Abstand  $i$  von  $\mathbf{c}$ ; in  $K_t(\mathbf{c})$  liegen alle diese Wörter für  $i = 0$  bis  $t$ .
- (b) Definitionsgemäß sind die  $|\mathcal{C}|$  Kugeln vom Radius  $t$  um Codewörter disjunkt; es gilt also

$$\left| \bigcup_{\mathbf{c} \in \mathcal{C}} K_t(\mathbf{c}) \right| = \sum_{\mathbf{c} \in \mathcal{C}} |K_t(\mathbf{c})| = |\mathcal{C}| \cdot |K_t(\mathbf{c})| =: s.$$

Alle betrachteten Wörter liegen in  $A^n$ . Es gilt also  $s \leq |A^n| = q^n$ .  $\square$

(c) **Kugelüberdeckung**

In der Ungleichung (\*\*) *gilt die Gleichheit genau dann*, wenn die zum Code gehörige Kugelpackung die Eigenschaft hat, dass jedes Wort von  $A^n$  in einer (eindeutig bestimmten) Kugel vom Radius  $t$  liegt. Ein Code dieser Eigenschaft heißt *perfekter Code*.

Genauer:

**E 1.15 Definition: Perfekter Code**

Ein Code  $\mathcal{C} \subseteq A^n$  heißt  **$t$ -perfekt** (mit  $t \in \mathbb{N} \cup \{0\}$ ), falls gilt:

- (i)  $K_t(\mathbf{c}_1) \cap K_t(\mathbf{c}_2) = \emptyset$  für alle  $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}$  mit  $\mathbf{c}_1 \neq \mathbf{c}_2$  (d.h.  $\mathcal{C}$  ist  $t$ -fehlerkorrigierend) und
- (ii)  $A^n = \bigcup_{\mathbf{c} \in \mathcal{C}} K_t(\mathbf{c})$  (d.h. die Kugeln vom Radius  $t$  um Codewörter überdecken  $A^n$ ).
- Statt  $t$ -perfekt sagt man auch nur **perfekt**.

**E 1.16 Anmerkung zu perfekten Codes**

(a) **Beispiele von perfekten Codes:**

*Triviale* perfekte Codes über dem Alphabet  $A$  sind:

- $A^n$  (mit  $t = 0$ ),
- nur aus einem einzigen Codewort der Länge  $n$  bestehende Codes (mit  $t = n$ ) und
- im Falle  $A = \{0, 1\}$  die Wiederholungs-codes  $\{00 \dots 0, 11 \dots 1\}$  der (ungeraden) Länge  $2m + 1$  (mit  $t = m$ ).

Wir erwähnen noch folgende perfekten Codes:

- einen binären Code der Länge  $n = 23$  mit  $2^{12}$  Wörtern (für  $t = 3$ ), den sogenannten *binären Golay-Code*  $\mathcal{G}_{23}$ ,
- einen ternären Code der Länge  $n = 11$  mit  $3^6$  Wörtern (für  $t = 2$ ), den sogenannten *ternären Golay-Code*,

- eine Serie von Codes  $\mathcal{H}_{q,r}$  mit  $n = (q^r - 1)/(q - 1)$ ,  $|\mathcal{H}_{q,r}| = q^{n-r}$  (mit  $t = 1$ ) für jede Primzahlpotenz  $q$ , die sogenannten *Hamming-Codes*
- (b) Perfekte Codes sind deswegen interessant, weil sie den Raum  $A^n$  gut ausschöpfen und weil die Decodierung für jedes Wort erklärt ist. Auch in anderem Zusammenhang gibt es Anwendungen; (s. Teil d)!).
- (c) Nach weitgehender Vorarbeit von J.H. van Lint (s. auch [1971]) ist es A. Tietäväinen (und unabhängig davon V.A. Zinovjev und K.V. Leontjev) gelungen, die  $t$ -perfekten Codes über Alphabeten von Primzahlpotenz-Ordnungen zu kennzeichnen, (s. van Lint [1982], Pless [1982]):

**Satz über perfekte Codes:**

*Ist  $\mathcal{C}$  ein nicht-trivialer perfekter Code über einem Alphabet  $A$  mit  $|A| = p^m$ ,  $p$  prim, dann hat  $\mathcal{C}$  die gleichen Parameter wie ein Hamming- oder ein Golay-Code.*

Für  $t > 2, t \neq 6$  ist nach M.R. Best die Voraussetzung “ $q$  Primzahlpotenz” überflüssig.

- (d) Mit Hilfe eines perfekten Codes über dem Alphabet  $\{0, 1, 2\}$  lässt sich das Problem von **Garantiesystemen bei Toto-Wetten** behandeln. Sei der Ausgang von 13 Fußballspielen vorauszusagen; unter Verwendung der Codierung:

$0 \hat{=} \text{Unentschieden}, \quad 1 \hat{=} \text{Heimsieg}, \quad 2 \hat{=} \text{Heimniederlage}$

wird das Ergebnis einer Ausspielung zu einem Wort  $\mathbf{v}$  aus  $\{0, 1, 2\}^{13}$ . Von einem Tipp  $\mathbf{c}$  mit 12 Richtigen hat  $\mathbf{v}$  den Abstand 1; die richtige Tippreihe  $\mathbf{v}$  liegt daher in  $K_1(\mathbf{c})$ . Wie aus Teil (a) zu ersehen, gibt es einen perfekten Code in  $\{0, 1, 2\}^{13}$ , den *ternären Hamming-Code  $\mathcal{H}_{3,3}$  der Länge 13*. Man kann dann (theoretisch) die Mittelpunkte der (den Raum überdeckenden) Kugeln vom Radius 1 um Codewörter als Tippreihen wählen und hat dann mit Sicherheit 12 Richtige zu erwarten. Nach 10.8(c) gibt es jedoch  $3^{13}/(1 + 13 \cdot 2) = 3^{10}$  solcher Punkte, für praktische Zwecke eine zu große Anzahl.

*Allgemein* kann man bei einer Toto-Wette mit  $n$  Tippreihen fragen, ob es ein Wettsystem gibt, das  $n - t$  Richtige garantiert. Eine Lösung würden  $t$ -perfekte ternäre Codes mit Wortlänge  $n$  liefern. Für  $t = 1$  sind nicht wesentlich andere Zahlen zu erwarten als beim eben behandelten Beispiel. Für  $t > 1$  gibt es (nach Tietäväinen und Pless), im Wesentlichen nur einen einzigen nicht-trivialen ternären  $t$ -perfekten Code, und zwar mit  $n = 11$ ,  $|\mathcal{C}| = 3^6 = 729$  und  $t = 2$  (d.h. 9 Richtige garantiert), den bereits erwähnten *ternären Golay-Code*. Wegen der zu großen Anzahl von nötigen Tipps und lediglich garantiertem Nebengewinn ist das Verfahren nicht lukrativ. Somit ist das Toto-System noch nicht zusammengebrochen.

Zwar ist die Voraussetzung einer *disjunkten* Überdeckung mit Kugeln gleichen Radius nicht *notwendige* Voraussetzung für ein Garantiesystem. Jedoch ist nicht zu erwarten, dass andere geeignete Überdeckungen mit wesentlich weniger Mittelpunkten auskommen als die der erwähnten perfekten Codes.

- (e) Wie bemerkt, ist bei obigen Beispielen die Disjunktheit der Kugeln nicht wesentlich (ausser für deren Anzahl). Wir definieren in diesem Zusammenhang:

Ist  $\mathcal{C} \subseteq A$ , dann heißt  $\rho(\mathcal{C})$  **Überdeckungsradius** von  $\mathcal{C}$ , falls  $\rho(\mathcal{C})$  der kleinste Radius  $\rho$  ist mit der Eigenschaft, dass die Kugeln vom Radius  $\rho$  um Codewörter die Menge  $A^n$  überdecken, also  $A^n = \bigcup_{\mathbf{c} \in \mathcal{C}} K_\rho(\mathbf{c})$  gilt. Bei einem  $t$ -perfekten Code

$\mathcal{C}$  ist  $\rho(\mathcal{C}) = t$ , und zusätzlich sind die erwähnten Kugeln disjunkt. Ja es gilt sogar:  $\mathcal{C}$  ist  $t$ -perfekt genau dann, wenn  $\rho(\mathcal{C}) = t$  ist und je zwei Codewörter mindestens den Hamming-Abstand  $2t + 1$  haben.

Um bei gegebenen Code  $\mathcal{C}$  dessen Fehlerkorrektur-Eigenschaften zu bestimmen, ist es nicht nötig, die Kugeln vom Radius  $t$  um Codewörter auf Disjunktheit zu untersuchen. Es reicht dazu, den Abstand je zweier Codewörter zu kennen, genauer den minimalen Wert dieses Abstands:

### E 1.17 Minimalabstand

- (a) *Definition:* Sei  $\mathcal{C} \subseteq A^n$ ; dann ist der Minimalabstand von  $\mathcal{C}$  definiert als

$$d_{\min}(\mathcal{C}) := \min_{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}, \mathbf{c}_1 \neq \mathbf{c}_2} d_H(\mathbf{c}_1, \mathbf{c}_2).$$

- (b) *Eigenschaften:*

$\mathcal{C} \subseteq A^n$  ist  $t$ -fehlerkorrigierend genau dann, wenn gilt:  $d_{\min}(\mathcal{C}) \geq 2t + 1$ .

**Beweis:** “ $\Rightarrow$ ” Sei  $d_{\min}(\mathcal{C}) =: s \leq 2t$ . Dann existieren  $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}$  mit  $d_H(\mathbf{c}_1, \mathbf{c}_2) = s$ . Ist  $s < t$ , so gilt  $\mathbf{c}_2 \in K_t(\mathbf{c}_1)$ . Ist  $s \geq t$ , so kann man bei  $\mathbf{c}_1$  unter den  $s$  Komponenten, in denen sich  $\mathbf{c}_1$  und  $\mathbf{c}_2$  unterscheiden,  $t$  Komponenten in solche von  $\mathbf{c}_2$  umändern. Es entsteht ein Wort  $\mathbf{z}$  mit  $d_H(\mathbf{c}_1, \mathbf{z}) = t$  und  $d_H(\mathbf{c}_2, \mathbf{z}) = s - t$ . Wegen  $s - t \leq t$  folgt  $K_t(\mathbf{c}_1) \cap K_t(\mathbf{c}_2) \neq \emptyset$ . Damit ist  $\mathcal{C}$  nicht  $t$ -fehlerkorrigierend (s. 10.6).

“ $\Leftarrow$ ” Ist  $\mathcal{C}$  nicht  $t$ -fehlerkorrigierend, so existieren  $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}$  und  $\mathbf{v} \in A^n$  mit  $\mathbf{v} \in K_t(\mathbf{c}_1) \cap K_t(\mathbf{c}_2)$ . Aus der Dreiecksungleichung (10.1b) ergibt sich

$$d_{\min}(\mathcal{C}) \leq d_H(\mathbf{c}_1, \mathbf{c}_2) \leq d_H(\mathbf{c}_1, \mathbf{v}) + d_H(\mathbf{v}, \mathbf{c}_2) \leq t + t. \quad \square$$

- (c) **Beispiel:** Beim Wiederholungscode  $\mathcal{C}_1 = \{000, 111\}$  aus Beispiel E 1.10 ist trivialerweise  $d_{\min}$  gleich 3, in Übereinstimmung damit, dass  $\mathcal{C}_1$  1-fehlerkorrigierend ist.