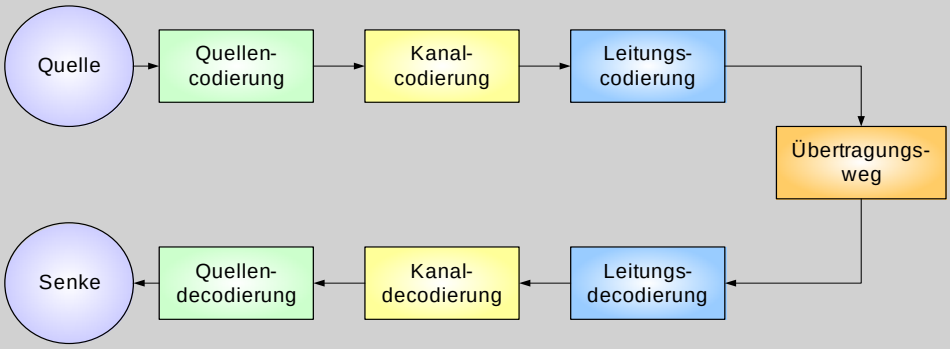
FernUniversität in Hagen



Praktikum

Versuch KS2: Digitale Übertragungstechnik I

Autoren:
Prof. Dr.-Ing. F. Kaderali
Dipl.-Ing. T. Kisner
Dipl.-Ing G. Steinkamp

Gliederung

Einführung und Überblick

Das gemeinsame Merkmal der Kommunikation ist die Übertragung von Nachrichten von einem Sender zu einem Empfänger, also von einer Nachrichtenquelle zu einer Nachrichtensenke. Bei der technischen Datenkommunikation werden diese Nachrichten als Zeichenfolgen dargestellt, wobei die Zeichen aus binären Digitalsignalen bestehen, d. h. die Signale können einen von zwei möglichen Zuständen einnehmen (z. B. High- und Low-Pegel).

Die zu übertragenden Nachrichten werden im Sender in Binärzeichen nach einer bestimmten Vorschrift umgesetzt. Diese Umsetzung nennt man Codierung. Auf der Empfängerseite werden die Signale dann wieder entsprechend zurückgewandelt (decodiert). Unter einer Codierung ist die Vorschrift für die eindeutige Zuordnung von Zeichen eines Zeichenvorrats zu denjenigen eines anderen Zeichenvorrats zu verstehen.

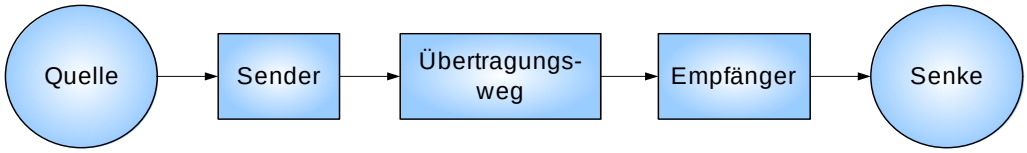


Abb. 1: Kommunikationsstrecke

Die allgemeine Aufgabe der Codierung besteht darin, die Daten in einer für die jeweiligen Anforderungen am besten geeignete Form darzustellen. Im wesentlichen sind hierbei zwei Aufgabengebiete zu unterscheiden. Zum einen soll durch die Codierung eine Reduktion der Quellenredundanz herbeigeführt werden, d. h. eine Darstellung der Daten mit einer geringstmöglichen Anzahl von Zeichen. Solche Codierungsverfahren werden als Quellencodierung bezeichnet. Zum anderen soll durch die Codierung ein einfaches und sicheres Übertragen von Daten einer Quelle zu einer Senke realisiert werden, d. h. es wird eine sogenannte Kanalcodierung durchgeführt. Bei der Kanalcodierung werden die Wörter einer Nachrichtenquelle so umcodiert, dass Codewörter entstehen, bei denen eine Verfälschung von Codesymbolen möglichst nicht zu einem neuen Codewort führt. Damit kann der Empfänger erkennen, dass eine Verfälschung aufgetreten ist und kann diese, bei geeigneter Codewahl, sogar beseitigen.

Die physikalische Umsetzung der binären Daten wird als Leitungscodierung bezeichnet. Damit ergibt sich folgende Darstellung der Kommunikationsstrecke.

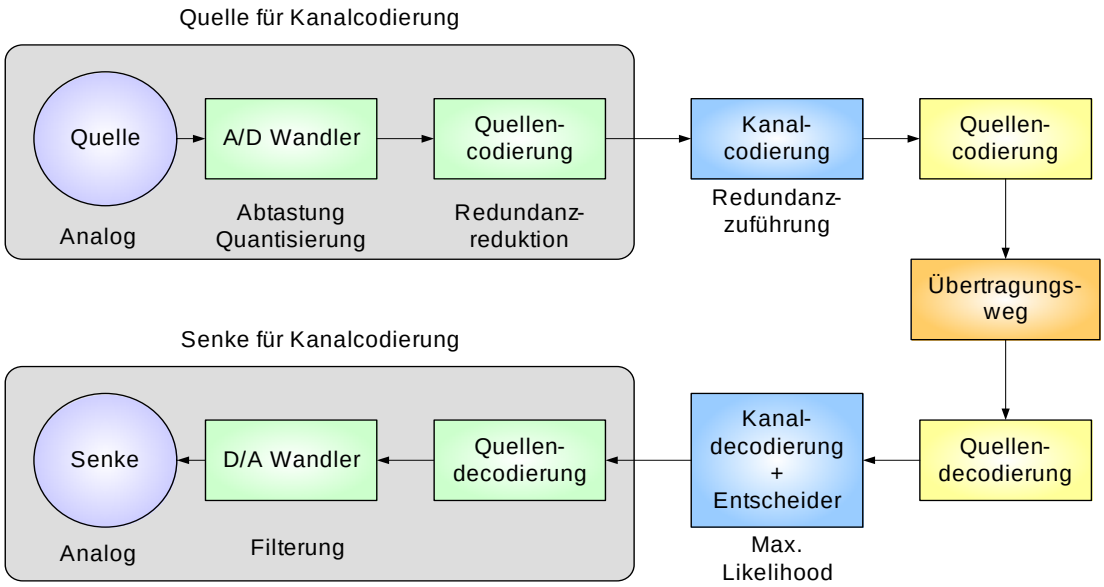


Abb. 2: Bedeutung der Kanalcodierung in der Kommunikationsstrecke

Gegenstand des Praktikums ist im wesentlichen der Einsatz von Codierungsmaßnahmen zur Sicherung der Übertragung, d. h. also die Kanalcodierung, unter Verwendung eines realen Übertragungssystems. Hierbei soll insbesonde das Verhältnis von Leistungsfähigkeit zum Aufwand der jeweils gewählten Codierung aufgezeigt werden.

In Kapitel Kapitel 1 dieser Praktikumsunterlagen ist der theoretische Hintergrund der Kanalcodierung umfassend dargestellt. Bei der Versuchsdurchführung werden nur einige der hier angesprochenen Aspekte in einer praktischen Anwendung realisiert.

Die Software- Konfiguration des Versuchsaufbaus wird in Kapitel 2 vorgestellt, die Beschreibung der Aufgabenstellung mit der Darstellung des Versuchsablaufs folgt in Kapitel 3.

Im letzten Kapitel sind Fragen und versuchsvorbereitende Aufgaben zusammengestellt. Eine Versuchsdurchführung ohne gründliche Vorbereitung ist nicht möglich. Die Beantwortung der Fragen und Lösung der Aufgaben wird vor Beginn des Versuchs mit dem Betreuer durchgesprochen.

Diese Seite bleibt aus technischen Gründen frei.

1 Kanalcodierung

In dem vorliegendem Kapitel wird die Kanalcodierung – d. h. die Codierung zur Erkennung und Korrektur von Fehlern – behandelt. Im ersten Abschnitt wird an Hand einfacher, in der Praxis üblicher Verfahren aufgezeigt, wie die einfache Wiederholung und die Paritätsprüfung zur Fehlererkennung angewandt werden. Der Begriff der Hamming-Distanz wird eingeführt und die Möglichkeit, Bündelfehler zu korrigieren, erörtert. Im nächsten Abschnitt werden lineare Codes behandelt. Es werden die Erzeugung und die Prüfung von linearen Codes an Hand von Matrizen dargestellt, die Eigenschaften der Matrizen diskutiert und der Hamming-Code sowie der erweiterte Hamming-Code behandelt. Für das Verständnis dieses Abschnittes ist erforderlich, dass der Student genügend Umgang mit der linearen Algebra hatte - insbesondere, dass er mit Begriffen wie Vektorraum, Basis, Dimension, lineare Unabhängigkeit vertraut ist. Die verwendeten Begriffe und Sätze sind im Anhang A zusammengestellt.

Um die Übersichtlichkeit zu gewähren, werden im Kurs einige Variablen weitgehend einheitlich verwendet. Diese sind:

Variable	Bedeutung
n	Anzahl der Informationssymbole (Rang der Generatormatrix G)
k	Anzahl der Prüfsymbole (Rang der Prüfmatrix)
r	Anzahl der Symbole im Codealphabet
$q = r^n$	Anzahl der Codewörter (Anzahl der Nachrichten)
$m = n + k$	Blocklänge.

1.1 Fehlererkennung und Fehlerkorrektur

Im ersten Abschnitt wird an Hand einfacher, in der Praxis üblicher Verfahren aufgezeigt, wie die einfache Wiederholung und die Paritätsprüfung zur Fehlererkennung angewandt werden. Der Begriff der Hamming-Distanz wird eingeführt und die Möglichkeit, Bündelfehler zu korrigieren, erörtert. Im nächsten Abschnitt werden lineare Codes behandelt. Es werden die Erzeugung und die Prüfung von linearen Codes an Hand von Matrizen dargestellt, die Eigenschaften der Matrizen diskutiert und der Hamming- Code sowie der erweiterte Hamming-Code behandelt. Für das Verständnis dieses Abschnittes ist erforderlich, dass der Student genügend Umgang mit der linearen Algebra hatte - insbesondere, dass er mit Begriffen wie Vektorraum, Basis, Dimension, lineare Unabhängigkeit vertraut ist. Die verwendeten Begriffe und Sätze sind im Anhang Anhang A zusammengestellt.

Wir betrachten eine aus k Symbolen bestehende Nachricht, die über einen gedächtnislosen Kanal mit der (Symbol-) Fehlerwahrscheinlichkeit p übertragen wird. Die Wahrscheinlichkeit, dass ein Symbol richtig übertragen wird, ist $(1-p)$, dass die ganze Nachricht richtig übertragen wird, $(1-p)^k$, dass sie fehlerhaft ist also $1-(1-p)^k$. Will man nun die Wahrscheinlichkeit, dass die Nachricht verfälscht wird, herunterdrücken, so wiederholt man sie einmal und vergleicht die empfangenen Nachrichten (Abb. 1.1-1a). Sind die empfangenen Nachrichten identisch, so nimmt man an, dass die Übertragung fehlerfrei war. Sind die Nachrichten verschieden, so verwirft man sie und veranlaßt eine**direkte Wiederholung** (negative Quittierung) oder eine **indirekte Wiederholung** (fehlende Quittierung). Alle Fehler bis auf identische Fehler in beiden Nachrichten werden bei diesem Verfahren entdeckt. Technisch günstig ist eine symbolweise Wiederholung und Vergleich der Nachricht, so dass im Fehlerfall unmittelbar eine Wiederholung veranlaßt werden kann (Abb. 1.1-1b).

Wiederholung zur Herabsetzung der Fehlerwahrscheinlichkeit

Die Wahrscheinlichkeit, dass beide Nachrichten in j bestimmten Stellen (d. h. j bestimmten Symbolen) verfälscht werden, ist $p^{2j}(1-p)^{2(k-j)}$, dass sie in j beliebigen, jedoch identischen Stellen verfälscht werden gleich

$$\left(\begin{array}{c} k \\ j \end{array} \right) p^{2j}(1-p)^{2(k-j)}. \qquad \text{1.1-1}$$

Die Wahrscheinlichkeit, dass unentdeckte Fehler auftreten, ist somit

$$\sum_{j=1}^{k} \left(\begin{array}{c} k \\ j \end{array} \right) p^{2j}(1-p)^{2(k-j)}. \qquad \text{1.1-2}$$

Im allgemeinen ist p klein, $(1-p)$ also nahe bei 1, so dass nur die ersten Werte zur Summe wesentlich beitragen. Der Preis, den man für die Erniedrigung der Wahrscheinlichkeit für unbemerkte Fehler bezahlt, besteht aus:

- Verdopplung der Nachrichtenlänge (und damit verbundenem längerem Verzug)
- schlechter Kanalausnutzung (d. h. niedrigere Informationsrate) und
- technischem Aufwand (für den Vergleich der Nachrichten und die Anforderung zur Wiederholung).

Eine dreifache Wiederholung ermöglicht es, die Fehlerwahrscheinlichkeit unbemerkter Fehler noch weiter herunterzudrücken. Die Entscheidungsregel lautet nun: Sind mindestens zwei der drei Nachrichten identisch, so werden diese als richtig bewertet, sonst verworfen. Im zweiten Fall wird eine Wiederholung veranlaßt.

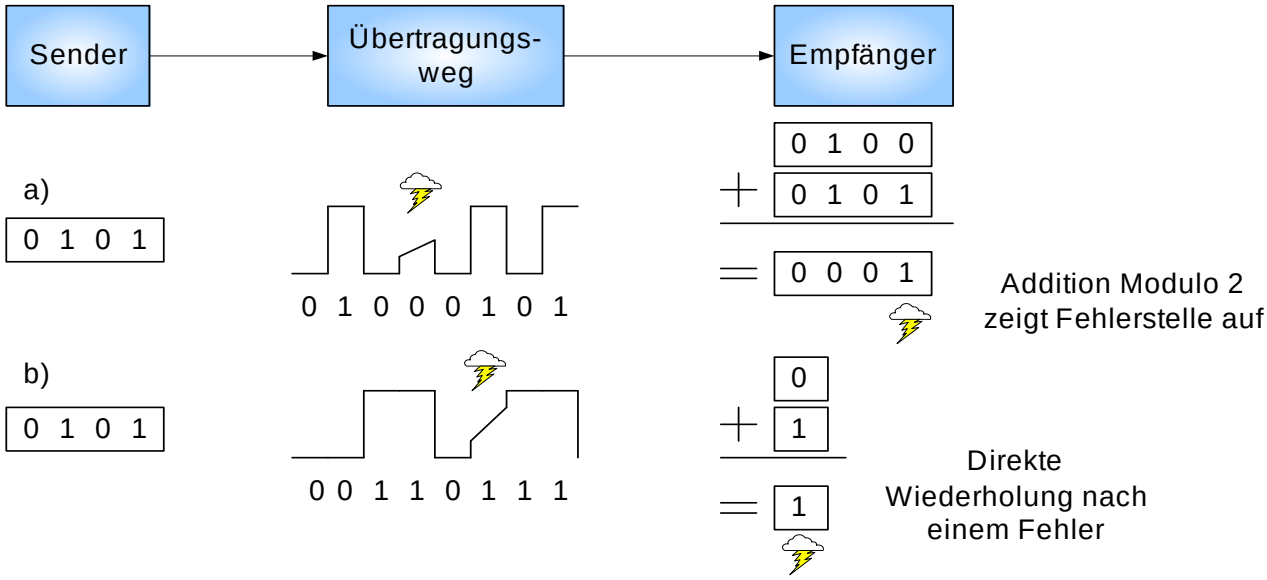


Abb. 1.1-1: Senden mit einmaliger Wiederholung

a. Wortweise Übertragung

b. Symbolweise Übertragung

Beispiel 1.1-1:
Wir betrachten die Übertragung einer Nachricht mit 10 Symbolen über einen gedächtnislosen Kanal mit der Fehlerwahrscheinlichkeit $p = 10^{-3}$.

Die Wahrscheinlichkeit, dass eine Nachricht mit $k = 10$ Symbolen unverfälscht ankommt, ist gleich $p^0(1-p)^k \approx 1-kp = 1-10\cdot 10^{-3} = 0,99$. Die Wahrscheinlichkeit, dass sie falsch ankommt, ist also 10^{-2}.

überträgt man nun mit einer Wiederholung, so ist die Wahrscheinlichkeit, dass beide Nachrichten unverfälscht ankommen, geringer, nämlich $p^0(1-p)^{2k} \approx 1-2kp = 1-20\cdot 10^{-3} = 0,98$.

Die Wahrscheinlichkeit, dass ein unentdeckter Fehler vorliegt, ist

$$\sum_{j=1}^{k} \binom{k}{j} \quad p^{2j} \quad (1-p)^{2(k-j)}.$$

Mit $p = 10^{-3}$ und $k = 10$ erhalten wir im einzelnen:

$$j \begin{pmatrix} k \\ j \end{pmatrix} p^{2j}(1-p)^{2(k-j)} \qquad \text{1.1-3}$$

$$\begin{aligned}
1 &\approx 9,82 \cdot 10^{-6} \\
2 &\approx 4,43 \cdot 10^{-11} \\
3 &\approx 1,18 \cdot 10^{-16} \\
4 &\approx 2,07 \cdot 10^{-22} \\
5 &\approx 2,49 \cdot 10^{-28} \\
6 &\approx 2,08 \cdot 10^{-34} \\
7 &\approx 1,19 \cdot 10^{-40} \\
8 &\approx 4,48 \cdot 10^{-47} \\
9 &\approx 9,98 \cdot 10^{-54} \\
10 &\approx 1,00 \cdot 10^{-60}
\end{aligned}$$

und somit

$$\sum \approx 9,82 \cdot 10^{-6}.$$

Die Wahrscheinlichkeit, dass unbemerkte Fehler vorliegen, konnte also um mehrere Zehnerpotenzen erniedrigt werden.

Paritätsprüfung

Das bei der Datenübertragung am häufigsten angewandte Verfahren ist die **Paritätsprüfung**. Zu einer vorgegebenen Anzahl von binären Codezeichen (z. B. einem Wort) wird ein Binärzeichen hinzugefügt, um ein Codewort mit gerader oder ungerader Parität (Quersumme Modulo 2) zu ergeben. Treten nun eine ungerade Anzahl von Verfälschungen im Codewort auf, so wird die Parität verletzt und der Fehler erkannt (Abb. 1.1-2).

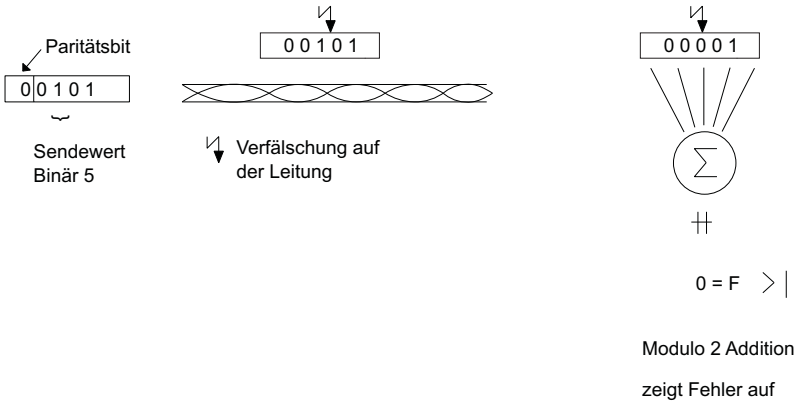


Abb. 1.1-2: Senden mit gerader Parität

Beispiel 1.1-2:
Der folgende 2-aus-5 Code

$$\begin{array}{ll} 1 \rightarrow \texttt{11000} & 6 \rightarrow \texttt{00110} \\ 2 \rightarrow \texttt{10100} & 7 \rightarrow \texttt{10001} \\ 3 \rightarrow \texttt{01100} & 8 \rightarrow \texttt{01001} \\ 4 \rightarrow \texttt{10010} & 9 \rightarrow \texttt{00101} \\ 5 \rightarrow \texttt{01010} & 10 \rightarrow \texttt{00011} \end{array} \qquad \text{1.1-4}$$

hat eine gerade Parität, denn jedes Codewort hat genau zwei Einsen. Tritt ein einfacher Fehler auf, z. B. an der zweiten Stelle der codierten Ziffer 5, so wird aus $5 \hat{=} \texttt{01010}$ ein unzulässiges Codewort `00010` mit ungerader Parität. Tritt jedoch ein weiterer Fehler z. B. an der dritten Stelle auf, so wird nun hieraus $\texttt{00110} \hat{=} 3$. Der Fehler ist nun nicht mehr erkennbar, denn es entsteht wieder ein zulässiges Codewort.

Das Paritätsprüfungsverfahren unterteilt alle möglichen Symbolkombinationen auf einfache Weise in zwei Klassen: unzulässige Symbolkombination mit ungerader Parität und (zulässige) Codewörter mit gerader Parität. Stets, wenn Fehler zu einer neuen Symbolfolge führen, die unzulässig ist, wird der Fehler erkannt. Führen sie zu einem (zulässigen) Codewort, ist eine Fehlererkennung nicht möglich.

Abstand zwischen zwei Codewörtern

Der **Abstand zwischen zwei Codewörtern** ist definiert als die Anzahl der Stellen, in denen sich die Codewörter unterscheiden. Wir betrachten nun einen Code mit nur zwei Codewörtern, die sich in a Stellen unterscheiden. Genau a Fehler an den entsprechenden Stellen führen das eine Codewort in das andere Codewort über. $(a-1)$ Fehler können also stets erkannt werden, denn sie führen zu unzulässigen Kombinationen. Treten f Fehler auf, wobei

$$f \leq \frac{a-1}{2}$$

ist, so ist es möglich, eindeutig auf das gesendete Wort zu schließen, denn der Abstand zwischen dem anderen Codewort und der entstandenen Symbolkombination muß (wegen $2f \leq a-1$) größer als f sein (Abb. 1.1-3).

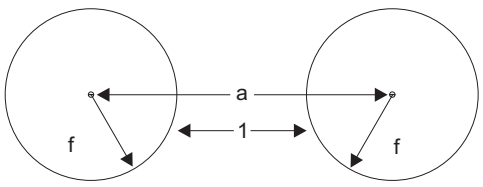


Abb. 1.1-3: Sphären mit Radius f um Codewörter A und B im Abstand $a = 2f + 1$. f Fehler sind noch korrigierbar.

Die Überlegungen sind auf mehrere Codewörter übertragbar, wobei an Stelle des Abstandes a nunmehr der Abstand $d = min\, a$ über alle Paare von Codewörtern gebildet wird.

Hamming-Distanz

Die **Hamming-Distanz** eines Codes ist definiert als der Mindestabstand zwischen zwei Codewörtern - sie ist gleich der Mindestanzahl der unterschiedlichen Symbole zweier Codewörter eines Codes. Bei einem Code mit der Hamming-Distanz d können $d-1$ Fehler erkannt oder

$$t \leq \frac{d-1}{2}$$

Fehler korrigiert werden.

Beispiel 1.1-3:
Wir betrachten den folgenden 4-aus-7 Code, mit 8 Codewörtern und der Blocklänge 7.

$A \rightarrow$ `0000000`	$E \rightarrow$ `1001110`	1.1-5
$B \rightarrow$ `1110100`	$F \rightarrow$ `0100111`	
$C \rightarrow$ `0111010`	$G \rightarrow$ `1010011`	
$D \rightarrow$ `0011101`	$H \rightarrow$ `1101001`	

Die Hamming-Distanz ist gleich 4. Es können 3 Fehler stets erkannt oder 1 Fehler stets korrigiert werden. Dies schließt nicht aus, dass im Einzelfall auch mehr Fehler erkannt bzw. korrigiert werden können. Tritt z. B. bei jedem Symbol des Codewortes D ein Fehler auf, so resultiert das Komplementärwort $\bar{D} \equiv$ `1100010`, also eine unzulässige Kombination - obwohl 7 Verfälschungen vorlagen, wird der Fehler erkannt. Werden lediglich das erste, dritte, sechste und siebte Symbol verfälscht, so erhalten wir statt D das (zulässige) Codewort $E \equiv$ `1001110`, vier Fehler werden also nicht erkannt. Tritt ein Fehler z. B. in der zweiten Stelle auf, so erhält man die unzulässige Kombination `0111101`. Diese hat den Abstand ≥ 3 von jedem Codewort $\neq D$ und den Abstand 1 von D, so dass bei maximal einem Fehler sicher auf D zurückgeschlossen werden kann. Das Maximum-Likelihood-Verfahren verwendet das Kriterium "geringste Fehlerwahrscheinlichkeit bei gleichverteilten Symbolen"; dies liefert dieselben Ergebnisse wie das Kriterium "minimaler Abstand", denn beide sind einander proportional - je größer der Abstand, den ein unzulässiges Wort von einem Codewort hat, desto geringer die Wahrscheinlichkeit, dass das unzulässige Wort aus dem Codewort hervorging.

Die bisherigen Überlegungen zeigen: je weiter Codewörter auseinanderliegen, bzw. je mehr unzulässige Kombinationen zwischen zwei Codewörtern

liegen, desto besser kann die Redundanz für die Fehlererkennung bzw. -korrektur ausgenutzt werden. Bei einem Blockcode der Länge n hat man insgesamt 2^n Symbolkombinationen. Hat man q Codewörter, so sind $(2^n - q)$ redundante Kombinationen vorhanden. Es gilt, die q Codewörter so zu wählen, dass der Abstand zwischen zwei beliebigen Codewörtern möglichst groß wird. Eine triviale Folgerung dieser Aussage für die Benennung von Dateien oder Variablen bei der Programmierung ist z. B., dass die Bezeichnungen so gewählt werden, dass sie sich in möglichst vielen Stellen unterscheiden.

Eine weitere Folgerung für die Codierung von Daten ist z. B., dass sie nicht geordnet, sondern besser zufällig codiert werden. Hat man z. B. 100 gleichwahrscheinliche Nachrichten und 8 binäre Symbole (d. h. 256 Wörter insgesamt) für ihre Codierung, so sollten sie nicht von binär 1 (`00000001`) bis binär 100 (`01100100`) durchcodiert, sondern möglichst gleich verteilt werden. Eine Zufallscodierung gewährleistet dies annähernd.

Beispiel 1.1-4:
Es werden 4096 gleichwahrscheinliche Nachrichten in Codewörter der Länge 16 binär codiert. Die geordnete Codierung liefert Codewörter von binär `0` bis binär `0000111111111111`. Tritt nun ein Fehler auf, so ist die Wahrscheinlichkeit, dass dies unerkannt bleibt, gleich

$$\frac{12}{16} = 0,75.$$

Tritt bei der zufälligen Codierung ein Fehler auf, so ist die Wahrscheinlichkeit , dass die Kombination ein Codewort ist und damit als unerkannter Fehler bleibt, ungefähr gleich $2^{12}/2^{16} = 0,0625$.

Fordern wir bei einem Blockcode der Länge m mit $r-$närem Alphabet und r^n vielen Codewörtern, dass t Fehler pro Wort korrigiert werden können, so können wir die erforderliche Redundanz leicht abschätzen. Im Abstand i von einem $r-$nären Wort der Länge m liegen

$$\binom{m}{i} (r-1)^i$$

Wörter. In der Kugel (vom Abstand t) liegen also

$$\sum_{i=0}^{t} \binom{m}{i} (r-1)^i$$

Wörter. Für die Korrekturfähigkeit müssen alle Kugeln um die Codewörter disjunkt sein, m also so groß gewählt werden, dass mindestens die Anzahl

aller Kombinationen größer oder gleich ist als die Anzahl aller Wörter in den disjunkten Kugeln, d. h.

$$r^m \geq \sum_{i=0}^{t} \left(\begin{array}{c} m \\ i \end{array} \right) (r-1)^i \cdot r^n,$$

oder

$$r^{m-n} \geq \sum_{i=0}^{t} \left(\begin{array}{c} m \\ i \end{array} \right) (r-1)^i. \qquad \text{1.1-6}$$

Bedingung für die Korrekturfähigkeit von t Fehlern

Gleichung (1.1-6) stellt eine notwendige **Bedingung** dar um die **Korrekturfähigkeit von t Fehlern** zu gewährleisten.

Beispiel 1.1-5:
Ein Quellenalphabet mit 2^3 Symbolen wird binär codiert. Es wird die Korrekturfähigkeit von $t = 3$ Fehlern pro Wort gefordert. Mit einem binären Blockcode mit $m = 10$ Symbolen pro Codewort ist wegen

$$2^7 = 128 \not\geq \sum_{i=0}^{3} \left(\begin{array}{c} 10 \\ i \end{array} \right) = 1 + 10 + \frac{10 \cdot 9}{1 \cdot 2} + \frac{10 \cdot 9 \cdot 8}{1 \cdot 2 \cdot 3} = 176$$

diese Forderung nicht erfüllbar. Mit $m = 11$ Symbolen ist sie wegen

$$2^8 = 256 \geq \sum_{i=0}^{3} \left(\begin{array}{c} 11 \\ i \end{array} \right) = 1 + 11 + \frac{11 \cdot 10}{1 \cdot 2} + \frac{11 \cdot 10 \cdot 9}{1 \cdot 2 \cdot 3} = 232$$

möglicherweise erfüllbar.

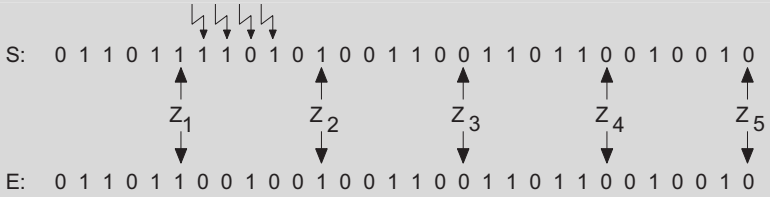
Bisher haben wir unsere Betrachtungen oft unter die Prämisse geringer Fehlerwahrscheinlichkeit bzw. von Einfach- oder wenigen Fehlern pro Codewort gestellt. In der Praxis ist es oft so, dass im allgemeinen die Fehlerwahrscheinlichkeit zwar gering ist, Fehler jedoch meist in Form von Bündelfehlern ("Bursts") auftreten. Pro Codewort treten dann Mehrfachfehler auf, und die einfache, wortweise Paritätsprüfung versagt. Eine einfache Abhilfe besteht darin, mehrere Wörter durch Untereinanderschreiben zu einem Block zusammenzufassen und diesen statt zeilenweise (bzw. wortweise) spaltenweise zu sichern, um damit eine Verteilung der Fehler auf die Paritätsbits zu erreichen. Verwendet man sowohl zeilen- als auch spaltenweise Paritätssicherung, so wird es möglich, bei Einfachfehlern (d. h. ein Fehler pro Zeile bzw. Spalte) die genaue Fehlerstelle anzugeben und damit zu korrigieren.

Beispiel 1.1-6:
Eine Nachricht besteht aus folgenden fünf Sendewörtern, SW1 bis SW5, wobei die Zeilen- und Spaltenparitätsbits eingetragen sind:

Sendewort		Zeilenparität
1	01101	1
2	11010	1
3	00110	0
4	11011	0
5	01001	0
Spaltenparität	00011	

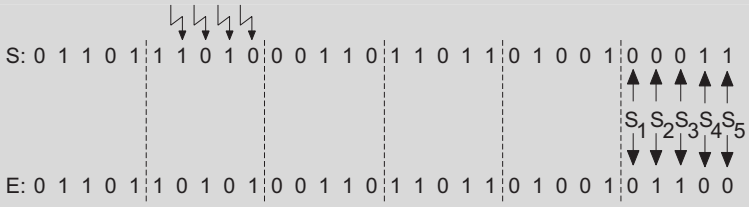
Bei der Übertragung tritt ein Bündelfehler der Länge 4 Bit ab dem 7. Symbol auf.

Wird eine wortweise (Zeilen-) Parität verwendet, so lautet die Sendefolge (S) und die empfangene Folge (E):



Der Fehler wird nicht erkannt, da die Paritäten alle stimmen, im Sendewort SW2 liegt jedoch eine vierfache Verfälschung vor!

Wird eine spaltenweise Parität verwendet, so lautet die Sendefolge (S) und die empfangene Folge(E):



Der Fehler wird nun erkannt, die Parität wird 4mal, nämlich bei $S_2S_3S_4S_5$ verletzt, so dass erkannt wird, dass vier Fehler vorliegen.

Hätte anstatt eines Bündelfehlers lediglich ein Einfachfehler am 7. Symbol vorgelegen, und wären sowohl Zeilen- als auch Spaltenparität geprüft worden, so wäre die Parität in der 2. Zeile und 2. Spalte verletzt, der Fehler hierdurch lokalisierbar und somit korrigierbar gewesen.

Anstatt nun die Parität über alle Symbole eines Wortes zu bilden, können wir auch differenzierter vorgehen und über ausgewählte Symbole die Parität bilden. Als Hilfsmittel zur Kennzeichnung der Stellen, die in der Paritätsprüfung einbezogen werden, verwenden wir ein Prüfwort, das aus `0` und `1`

besteht: durch 1 an einer Stelle wird angegeben, dass diese Stelle in die Prüfung einbezogen wird, durch 0, dass sie in die Prüfung nicht einbezogen wird. Liegt ein Wort vor, so bildet man die Parität über die Symbole, an deren Stelle im Prüfwort eine Eins steht - ist die Parität erfüllt, handelt es sich möglicherweise um ein Codewort, sonst sicher um eine unzulässige Kombination.

Beispiel 1.1-7:
Wir möchten die Paritätsprüfung über jeweils gerade und ungerade Symbole eines Wortes mit 8 Symbolen bilden. Die beiden Prüfwörter lauten:

$$\begin{aligned} \texttt{10101010} &= P_1 \quad \text{für die geraden und} \\ \texttt{01010101} &= P_2 \quad \text{für die ungeraden Symbole.} \end{aligned} \tag{1.1-7}$$

Das empfangene Wort $w = \texttt{00101011}$ bildet mit P_1 die (Modulo 2) Quersumme

$$0+0+1+0+1+0+1+0=1,$$

verletzt also die Parität. Mit P_2 bildet es die Summe

$$0+0+0+0+0+0+0+1=1,$$

verletzt wiederum die Parität.

Wir schließen daraus, dass sowohl in den geraden Stellen als auch in den ungeraden Stellen Fehler vorliegen.

Wir haben für die Quersummenbildung die Modulo 2 Addition, bei der differenzierten Auswahl der Stellen, die in eine Prüfung einbezogen werden, die Modulo 2 Multiplikation mit anschließender Modulo 2 Addition für Paritätsbildung verwendet. Es wird hier ersichtlich, dass einige Codes auf algebraischen Strukturen basieren - diese werden algebraische Codes genannt. Wir werden Codes, die auf linearen Räumen basieren, im nächsten Abschnitt behandeln - sie werden lineare Codes genannt. Hierzu werden wir einige mathematische Begriffe der linearen Algebra heranziehen. Diese sind im Anhang A.1 und Anhang A.2 zusammengestellt.

Selbsttestaufgabe 1.1-1:

a. *Was versteht man unter dem "Abstand zwischen zwei Codewörtern", und welcher Zusammenhang ergibt sich mit dem Begriff "Hamming-Distanz" eines Codes?*

b. Bestimmen Sie für die nachfolgend aufgeführten Codewörter die Hamming -Distanz und machen Sie eine Aussage, wieviele Fehler stets erkannt werden können.

A	00000	1.1-8
B	11010	
C	01101	
D	10110	

1.2 Lineare Codes

Wir nehmen an, dass dem zu betrachtenden Code gewisse algebraische Strukturen zugrunde liegen. Wir gehen von der Definition eines Codes im Abschnitt 6.1 aus. Da wir Blockcodes betrachten, sind die Wörter w nun Elemente aus B^m. Wir setzen zusätzlich voraus, dass die Menge $B = \{x_1, \ldots x_r\}$, die wir als Alphabet des Codes bezeichnet haben, einen endlichen Körper bildet. Dies bedeutet, dass für die Elemente der Menge eine Addition (+) und eine Multiplikation (·) so definiert sind, dass die Axiome der Addition $A1 - A3$, der Multiplikation $M1 - M3$ und die Distributivgesetze D (siehe Anhang A.1) gelten. Wir fassen ferner B^m (die Menge aller m-Tupel über B) als einen Vektorraum über dem Körper $(B, +, \cdot)$ auf; dies setzt voraus, dass die Addition von Vektoren und deren Multiplikation mit Elementen des Körpers so definiert sind, dass die Axiome $V1 - V4$ (siehe Anhang A.2) gelten.

linearer Code

Ein **linearer Code**C (genauer die Codewörter des Codes C) wird nun als Untervektorraum der Dimension n des Vektorraumes B^m definiert. Da wir ein Alphabet mit r Elementen für die Codierung angenommen haben, hat der Code $q = r^n$ Codewörter. Man spricht auch von einem (m, n)-Code, wobei m die Blocklänge und n die Ordnung des Untervektorraumes ist, die später als die Anzahl der (r-nären) Informationssymbole interpretiert wird.

Eine Basis des Untervektorraumes der Dimension n hat n Elemente, und wir können alle Codewörter aus Linearkombinationen der Basisvektoren erzeugen. Hierin liegt ein erheblicher Vorteil von linearen Codes: bei der Überprüfung, ob eine beliebige Kombination der Symbole ein zulässiges Codewort ist, braucht man nicht alle Codewörter gespeichert vorliegen zu haben, um einen Vergleich durchführen zu können; eine überprüfung, ob sie als Linearkombination der Basisvektoren zusammengestellt werden kann, genügt. Wir wollen dies weiter formalisieren und führen eine Matrizen-Darstellung von Codes ein.

C sei ein (m, n)-Code, $\{g_1, g_2, \ldots g_n\}$ eine Basis von C. Dann heißt

$$G = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_n \end{bmatrix} \qquad \text{1.2-1}$$

Basismatrix
Generatormatrix

eine **Basismatrix** oder eine **Generatormatrix** des linearen Codes C. G ist eine (n, m)-Matrix vom Rang n. Jedes Codewort (Vektor aus C) ist eindeutig darstellbar als Linearkombination aus den Basisvektoren $g_1, \ldots g_n$:

$$w = \sum_{i=1}^{n} \alpha_i g_i, \qquad \alpha_i \in B. \qquad \text{1.2-2}$$

Die Vektoren g_i werden wir auch in der Schreibweise

$$g_i = (g_{i1} g_{i2} \ldots g_{im})$$

darstellen, so dass G als Matrix geschrieben wird:

$$G = \begin{bmatrix} g_{11} & \cdots & g_{1m} \\ g_{21} & & \vdots \\ \vdots & & \\ g_{n1} & \cdots & g_{nm} \end{bmatrix}. \qquad \text{1.2-3}$$

Beispiel 1.2-1:
Das binäre Alphabet $B = \{0, 1\}$ mit der Addition (+) und der Multiplikation (·) entsprechend den Tabellen

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

bildet den Körper $F_2 = (B, +, \cdot)$.

Die $2^m m$-Tupel $v_i = (a_{i1} a_{i2} \ldots a_{im})$ $(a_{ij} \in B)$ bilden den Vektorraum B^m. Für $m = 7$ bildet die Basis

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \qquad \text{1.2-4}$$

einen Untervektorraum der Dimension 3. Er besteht aus den $2^3 = 8$ Codewörtern des Beispiels 7.3. Die Basis besteht aus den Codewör-

tern B, C und D. Jedes der anderen Codewörter kann als eine Linearkombination der Basis dargestellt werden. So ist z. B. $H = 1 \cdot B + 0 \cdot C + 1 \cdot D = B + D$. Die Koeffizienten der Basisvektoren (101) legen H eindeutig fest. Die Codewörter B, C, H bilden auch eine Basis G' von C:

$$G' = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}. \qquad 1.2\text{-}5$$

Die Koeffizienten (001) der Basis G' legen nun H eindeutig fest.

Da jedes Codewort eine Nachricht darstellt, können wir mit dem Code genau r^n Nachrichten übertragen. Wir können dabei die Nachrichten jeweils durch ein n-Tupel $a = (a_1 a_2 \ldots a_n)$ mit $a_i \in B$ festlegen. Wir gehen stets davon aus, dass die **Zuordnung von Codewörtern zu den Informations-n-Tupeln** (d. h. Nachrichten) durch eine lineare Abbildung $\varphi : B^n \to B^m$ mit

Zuordnung von Nachrichten zu Codewörtern

$$\varphi(a) = a \cdot G = \sum_{i=1}^{n} a_i g_i \qquad 1.2\text{-}6$$

beschrieben wird.

Beispiel 1.2-2:
Der lineare Code mit der Generatormatrix G' aus Beispiel 1.2-1 ermöglicht $2^3 = 8$ Nachrichten zu codieren bzw. zu übertragen. Seien diese Nachrichten binär durchgezählt:

$$\begin{array}{ll} N_1 & 0\,0\,0 \\ N_2 & 0\,0\,1 \\ N_3 & 0\,1\,0 \\ N_4 & 0\,1\,1 \\ N_5 & 1\,0\,0 \\ N_6 & 1\,0\,1 \\ N_7 & 1\,1\,0 \\ N_8 & 1\,1\,1. \end{array} \qquad 1.2\text{-}7$$

Legt man die Abbildung $\varphi(a) = a \cdot G'$ für die Zuordnung der Nachrichten zu den Codewörtern fest, so erhält man für die Nachricht N_4 das Codewort

$$\varphi = [011] \cdot \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [1010011]$$

Definieren wir das Skalarprodukt von Vektoren in der üblichen Weise, so können wir den zu C orthogonalen Vektorraum C^d definieren:

$$C^d = \{v \in B^m | v \cdot w = 0 \quad \text{für alle } w \in C\}. \tag{1.2-8}$$

dualer Code

C^d ist wieder Untervektorraum von B^m und wird deshalb der zu C **duale Code** genannt. Für die Dimension von C^d gilt (Anhang A.2.12)

$$dim\, C + dim\, C^d = m. \tag{1.2-9}$$

Es kann gezeigt werden, dass $\left(C^d\right)^d = C$ ist, und somit ist C auch der duale Code von C^d.

Sei H eine Basismatrix von C^d:

$$H = \begin{bmatrix} h_1 \\ \vdots \\ h_{m-n} \end{bmatrix}$$

Kontrollmatrix

H wird eine **Kontrollmatrix** (oder Paritätsmatrix) von C genannt. Wegen Gleichung (1.2-9) hat sie den Rang $(m-n)$.

Da durch die Basismatrix ein Vektorraum eindeutig bestimmt ist, ist durch die Generatormatrix der Code C, durch die Kontrollmatrix der Code C^d eindeutig bestimmt. Umgekehrt ist durch den Coderaum die Generator- oder Kontrollmatrix nicht eindeutig bestimmt.

Wegen Gleichung (1.2-8) gilt die Beziehung

$$GH^T = 0 \quad \text{bzw. } HG^T = 0. \tag{1.2-10}$$

Ist $v \in B^m$ und H eine Kontrollmatrix des linearen Codes C, so gilt die folgende, für die Paritätsprüfung wichtige Äquivalenz:

$$(v \in C) \Leftrightarrow (vH^T = 0) \Leftrightarrow (Hv^T = 0). \tag{1.2-11}$$

Wir wollen diese beweisen.

Es sei $v \in C$. Da C^d orthogonal zu C ist, gilt $v \cdot v' = 0$ für jeden Basisvektor v' jeder Basis von C^d. Es ist deshalb $vH^T = 0$. Umgekehrt sei $vH^T = 0$, dann gilt $vv' = 0$ für jeden Vektor v' einer Basis von C^d. v ist also orthogonal zu jedem Vektor aus C^d, v gehört zum Dualcode von C^d, also $v \in C$. Somit haben wir die erste Äquivalenz. Die zweite Äquivalenz gilt wegen $vH^T = 0 \Leftrightarrow (vH^T)^T = 0 \Leftrightarrow Hv^T = 0$.

Gleichung (1.2-11) liefert uns nun die Möglichkeit zu überprüfen, ob eine Kombination aus B^m ein Codewort ist. Dies ist genau dann der Fall, wenn das Produkt mit einer Kontrollmatrix $Hv^T = 0$ liefert. Hierin ist auch der Name Kontrollmatrix begründet.

Tritt bei der Übertragung eines Codewortes $v \in C$ ein Fehler auf, so erhält man beim Empfang eine Kombination $k \in B^m$. Der Fehler kann als Vektor

$e = (k - v) \in B^m$ dargestellt werden. Ist $e \notin C$, so kann der Fehler erkannt werden, denn es ist

$$s = kH^T = (v + e)H^T = vH^T + eH^T \neq 0. \qquad \text{1.2-12}$$

Syndrom

s nennt man das **Syndrom** des Vektors k bzw. e bezüglich der Kontrollmatrix H. Wir werden sehen, dass bei einer geschickten Wahl der Kontrollmatrix oft aus dem Syndrom noch weitere Hinweise abgeleitet werden können, z. B. über die Stelle, wo der Fehler im Codewort aufgetreten ist; somit erhält man die Möglichkeit, den Fehler zu korrigieren.

Beispiel 1.2-3:
Die Kontrollmatrix

$$H = \left[\begin{array}{cccc:ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{c} h_1 \\ h_2 \\ h_3 \\ h_4 \end{array}\right] \qquad \text{1.2-13}$$

hat die Dimension 4 und bildet eine Basis des zu C von Beispiel 1.2-1 dualen Codes C^d. C^d hat $2^4 = 16$ Codewörter. Wie man sieht, können Basisvektoren der Kontrollmatrix Codewörter von C sein (wie $h_2 = F$) oder auch nicht (wie h_1). Um zu entscheiden, ob ein m-Tupel $v = (\texttt{0010110}) \in B^7$ ein Codewort von C bildet, braucht man es nicht mit allen 8 Codewörtern von C zu vergleichen. Es genügt die Probe, ob $H \cdot v^T = 0$ ist. Da bereits $h_1 v \neq 0$ ist, ist v kein Codewort von C.

Hamming-Gewicht

Das **Hamming-Gewicht** eines Vektors $v = (v_1 \ldots v_m)$ aus B^m ist definiert als

$$W(v) = \sum_{i=1}^{m} \rho(v_i), \qquad \text{1.2-14}$$

wobei

$$\rho(v_i) = \begin{cases} 0 & falls\ v_i\ das\ Nullelement\ von\ B\ ist \\ 1 & sonst. \end{cases}$$

$W(v)$ ist damit genau die Anzahl der von Null verschiedenen Komponenten von v.

Abstand zwischen zwei Vektoren

Mit Hilfe von W können wir den **Abstand zwischen zwei Vektoren** $v, w \in B^m$ als

$$d(v, w) = W(v - w) \qquad \text{1.2-15}$$

definieren. Der Abstand $d(v, w)$ ist damit genau die Anzahl der Komponenten, in denen sich v und w unterscheiden, wie wir es in Abschnitt 1.1 bereits

definierten. Der Abstand $d()$ ist eine Metrik auf dem Vektorraum, denn es gilt

$$\begin{aligned} d(w,w) &= 0 \quad \text{für alle } w \in B^m \\ d(w,v) &= d(v,w) \quad \text{für alle } w,v \in B^m \text{ und} \\ d(v,w) &> 0 \quad \text{für } v \neq w. \end{aligned} \tag{1.2-16}$$

Wegen $W(x) + W(y) \geq W(x+y)$ gilt

$$d(u,v) + d(v,w) = W(u-v) + W(v-w) \geq W(u-w) = d(u,w). \tag{1.2-17}$$

Hamming-Distanz

Wir können nun die **Hamming-Distanz**, die wir als den Mindestabstand zwischen zwei Codewörtern eines Codes definierten, für einen linearen Code einfacher angeben. Sie ist genau gleich dem minimalen Hamming-Gewicht, d. h.

$$\min_{\substack{v,w \in C \\ v \neq w}} \{d(v,w)\} = \min_{\substack{u \in C \\ u \neq 0}} \{W(u)\}. \tag{1.2-18}$$

Denn ist für ein Paar $v, w\, d(v,w) =$ Min, so existiert ein Codewort $u = (v - w) \in C$ mit $W(u) = W_{\min}$. Umgekehrt: ist für ein Codewort $u\ W(u) = W_{\min}$, so ergibt sich mit dem Nullwort das Paar mit $d(u,0) = Min$.

Beispiel 1.2-4:
Wir betrachten den $(6, 4)$ Code, der durch die Generatormatrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix}$$

erzeugt wird. Er hat $2^4 = 16$ Codewörter, die man durch Linearkombinationen von $g_1, \ldots g_4$ erhält.

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ g_1 \\ g_2 \\ g_1 + g_2 \\ g_3 \\ g_3 + g_1 \\ g_3 + g_2 \\ g_3 + g_2 + g_1 \\ g_4 \\ g_4 + g_1 \\ g_4 + g_2 \\ g_4 + g_2 + g_1 \\ g_4 + g_3 \\ g_4 + g_3 + g_1 \\ g_4 + g_3 + g_2 \\ g_4 + g_3 + g_2 + g_1 \end{bmatrix}$$

Die Hamming-Distanz ist gleich $d = 2$.

Da für

$$H = \left[\begin{array}{cccc:cc} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right]$$

gilt

$$GH^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

und Rang $H = 2$, ist H eine Prüfmatrix.

H hat den Rang 2, der duale Code C^d hat somit 4 Codewörter:

$$C^d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

C^d hat die Hamming-Distanz $d^d = 3$.

Spalten einer Kontrollmatrix

Wir wollen nun den Zusammenhang zwischen dem Hamming-Gewicht und den **Spalten einer Kontrollmatrix** näher untersuchen.

Hat ein Codewort eines linearen Codes das Hamming-Gewicht W, so gibt es ein Codewort v mit W Elementen $\neq 0$. Wir können symbolisch das Codewort wie folgt schreiben

$$c = (00C_1 0 \ldots 0C_2 00 \ldots C_w 0),$$

wobei wir die (beliebig verteilten) W Symbole $\neq 0$ durch $C_1, C_2, \ldots C_w$ gekennzeichnet haben. Wegen $Hv^T = 0$, ausführlich

$$\begin{bmatrix} h_{11} & \ldots & h_{1m} \\ \vdots & & \vdots \\ h_{m-n,1} & \ldots & h_{m-n;m} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ C_1 \\ 0 \\ \vdots \\ C_2 \\ \vdots \\ C_w \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

bedeutet dies, dass W Spalten (nämlich die zu den Koeffizienten C_i gehörenden Spalten) linear abhängig sind. Umgekehrt: ergibt die Summe von k Spalten mit den Koeffizienten $C_i \neq 0$ Null, so kann man ein Codewort mit dem Hamming-Gewicht k angeben. Wir stellen somit fest, dass ein linearer Code nur dann die Hamming-Distanz W bzw. das Minimal-Gewicht W haben kann, wenn jede Kombination von $W - 1$ oder weniger Spalten von H linear unabhängig ist. Dies eröffnet uns eine Möglichkeit, Codes mit einer gewünschten Hamming-Distanz zu konstruieren.

Beispiel 1.2-5:
Wir betrachten den Code des Beispiel 1.2-4. Das Codewort $g_3 + g_2 = (\texttt{001010})$ hat das Hamming-Gewicht 2. Die Spalten 3 und 5 von H sind linear abhängig, denn es gilt

$$1 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Umgekehrt sind die Spalten 2 und 4 abhängig. Deswegen können wir ein Codewort $C = (\texttt{010100})$ mit dem Gewicht 2 angeben.

Wir wollen uns nun linearen Codes widmen, die eine Generatormatrix mit einer sehr einfachen Form haben:

$$G = [E_n \vdots P].$$

E_n ist dabei eine $(n \times n)$ Einheitsmatrix (d. h. Diagonalmatrix mit Eins aus dem Körper B als Diagonalelemente) und P eine beliebige $n \times (m - n)$ Matrix ohne Nullspalte. Man nennt einen Code mit einer solchen kanonischen Generatormatrix einen **systematischen Code**.

systematischer Code

Ist eine Basis eines Vektorraumes gegeben, so erhält man durch Elementaroperationen (Anhang A.2) an den Basisvektoren wieder eine neue Basis desselben Raumes. Für die Generatormatrix bedeutet dies, dass Elementaroperationen an den Zeilen der Generatormatrix wieder eine Generatormatrix desselben Codes liefern. Eine Vertauschung der Spalten einer Generatormatrix entspricht der Vertauschung von Symbolen bei der Codierung der Nachrichten. Diese spielt bei der Fehlererkennung bzw. Fehlerkorrektur und der Nachrichtenübermittlung oder -speicherung keine Rolle, wenn Symbolstörungen voneinander unabhängig sind. Codes, die Generatormatrizen haben, die durch elementare Zeilenoperationen und Spaltenvertauschungen ineinander überführt werden können, nennt man (kombinatorisch) äquivalent. Es kann nun gezeigt werden, dass jeder lineare Code einen äquivalenten systematischen Code besitzt. In diesem Sinne ist die Betrachtung von systematischen Codes keine Einschränkung. Wir wollen den Beweis jedoch nicht weiter ausführen.

Es sei $G = [E_n \vdots P]$ eine Generatormatrix eines systematischen Codes. Dann ist

$$H = [-P^T \vdots E_{m-n}] \qquad 1.2\text{-}19$$

eine Prüfmatrix[1], denn es gilt

$$GH^T = [E_n \vdots P] \begin{bmatrix} -P \\ \dots \\ E_{m-n} \end{bmatrix} = -E_n P + P E_{m-n} = 0.$$

kanonische Form der Prüfmatrix

Gleichung (1.2-19) nennt man die **kanonische Form der Prüfmatrix**.

In der kanonischen Form der Matrizen nennt man die ersten n Stellen der Codewörter die Informationsstellen, die restlichen $(m-n)$ Stellen die Prüf- oder Kontrollstellen. Sind die Informationsstellen einer Nachricht vorgegeben, so können wegen der einfachen Form der Generatormatrix die Prüfstellen unmittelbar als Linearkombination der Zeilen von P (entsprechend Gleichung (1.2-6) angegeben werden.

Beispiel 1.2-6:
Wir greifen wieder das Beispiel 1.2-4 auf. Die Generatormatrix in der kanonischen Form lautet:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & \vdots & 1 & 0 \\ 0 & 1 & 0 & 0 & \vdots & 0 & 1 \\ 0 & 0 & 1 & 0 & \vdots & 1 & 0 \\ 0 & 0 & 0 & 1 & \vdots & 0 & 1 \end{bmatrix},$$

somit ist

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}.$$

Da bei der Modulo 2 Addition $-1 = +1$, haben wir

$$H = \left[-P^T \vdots E_2 \right] = \begin{bmatrix} 1 & 0 & 1 & 0 & \vdots & 1 & 0 \\ 0 & 1 & 0 & 1 & \vdots & 0 & 1 \end{bmatrix},$$

1 $-$ bedeutet hierbei die Bildung des Inversen bezüglich der Addition; T bezeichnet die transponierte Matrix (d. h. die Matrix, in der die Zeilen und Spalten vertauscht wurden).

und es gilt

$$GH^T = \left[\begin{array}{cccc:cc} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right] \cdot \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ \dots & \dots \\ 1 & 0 \\ 0 & 1 \end{array}\right] \qquad \text{1.2-20}$$

$$= \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \cdot \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{array}\right] + \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{array}\right] \cdot \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right]$$

$$= \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{array}\right] + \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{array}\right] = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right].$$

Sind die Informationsstellen einer Nachricht v vorgegeben, $v = (\texttt{1001xy})$, so errechnen sich die Prüfbits zu

$$(xy) = (p_1) + (p_4) = (10) + (01) = (11).$$

Wir haben bereits gesehen, dass wir, um einen linearen Code mit der Hamming-Distanz W zu erhalten, lediglich dafür zu sorgen brauchen, dass die Kontrollmatrix H so gewählt wird, dass jede Kombination von $(W-1)$ oder weniger Spalten von H linear unabhängig ist. Dies ist im allgemeinen nicht einfach. Für einen linearen binären Code, dessen Kontrollmatrix k Zeilen haben soll, ist es jedoch besonders einfach, die Spalten von H so zu bestimmen, dass sie alle verschieden und somit paarweise unabhängig werden. Man braucht lediglich alle möglichen $2^k - 1$ von Null verschiedenen Kombinationen mit k Elementen aus $\{0, 1\}$ zu bilden und sie als Spalten zu nehmen. Durch eine geeignete Reihenfolge der Spalten kann man die Matrix in die kanonische Form bringen. Den so gewonnenen Code C_k nennt man den **binären Hamming-Code**. Er hat die Hamming-Distanz $d = 3$, denn 2 Spalten von H sind stets linear unabhängig, während es 3 Spalten gibt, die linear abhängig sind.

binärer Hamming-Code

Beispiel 1.2-7:
Wir erhalten C_4, den Hamming-Code mit 4 Kontrollzeilen, indem wir alle $2^4 - 1 = 15$ von Null verschiedenen Kombinationen mit 4 Elementen aus

$\{0,1\}$ bilden und die so gewonnenen Spalten in die kanonische Form bringen:

$$H_4 = \left[\begin{array}{cccccccccccc|cccc} 0&0&0&0&1&1&1&1&1&1&1&1&0&0&0\\ 0&1&1&1&0&0&0&1&1&1&1&0&1&0&0\\ 1&0&1&1&0&1&1&0&0&1&1&0&0&1&0\\ 1&1&0&1&1&0&1&0&1&0&1&0&0&0&1 \end{array}\right].$$

Somit ist

$$G_4 = \left[\begin{array}{cccccccccccc|cccc} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\\ 0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&1\\ 0&0&1&0&0&0&0&0&0&0&0&0&0&1&1&0\\ 0&0&0&1&0&0&0&0&0&0&0&0&0&1&1&1\\ 0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&1\\ 0&0&0&0&0&1&0&0&0&0&0&0&1&0&1&0\\ 0&0&0&0&0&0&1&0&0&0&0&0&1&0&1&1\\ 0&0&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&1\\ 0&0&0&0&0&0&0&0&0&1&0&0&1&1&1&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&1&1&1&1 \end{array}\right],$$

und der Code hat 2^{11} Codewörter.

$r-$närer Hamming-Code

Es sei an dieser Stelle davor gewarnt, für ein r**-näres** Alphabet den **Hamming-Code** durch alle $r^k - 1$ Kombinationen der Elemente angeben zu wollen. Für jeden Spaltenvektor erhält man nämlich durch die Multiplikation mit den $(r-1)$ von Null verschiedenen Elementen des Alphabetes $(r-1)$ verschiedene, jedoch abhängige Elemente, so dass insgesamt

$$\frac{r^k - 1}{r - 1}$$

linear unabhängige Spalten übrig bleiben. Der Hamming-Code hat dann die Länge

$$m = \frac{r^k - 1}{r - 1}. \qquad \text{1.2-23}$$

Beispiel 1.2-8:
Wir wollen eine Prüfmatrix des Hamming-Codes mit dem Alphabet aus 3 Elementen $\{0,1,2\}$, der Modulo 3 Addition und Multiplikation, und 3 Prüfstellen aufstellen. Wir bilden alle Kombinationen mit drei Elementen

aus $\{0,1,2\}$ und streichen die von den vorhergehenden linear abhängigen Kombinationen und erhalten im Einzelnen:

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

1.2-24

Der Code besteht somit aus Codewörtern der Länge

$$m = \frac{3^3 - 1}{3 - 1} = 13,$$

und die Prüfmatrix in der kanonischen Form lautet:

$$H = \left[\begin{array}{cccccccccc:ccc} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 0 & 1 & 0 \\ 1 & 2 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 0 & 1 \end{array}\right]. \qquad \text{1.2-25}$$

Der Hamming-Code hat die Hamming-Distanz $d = 3$, 3 Paritätsstellen und 10 Informationssymbole.

Effizienz der Kanalcodierung

Der Begriff **Effizienz** $E = \frac{H(X)}{l_m} \cdot \frac{1}{ld(r)}$ kann auch für die Kanalcodierung anwendet werden. Für die Effizienz des binären Hamming-Codes mit k Prüfstellen und der Blocklänge $m = 2^k - 1$ erhalten wir für eine Quelle mit 2^n gleichverteilten Nachrichten (Quelle mit maximaler Entropie $H(X) = H_{\max}(X) = n$)

$$E = \frac{H(X)}{l_m} \cdot \frac{1}{ld(r)} = \frac{n}{m} = \frac{m-k}{m} = 1 - \frac{k}{2^k - 1}. \qquad \text{1.2-26}$$

Die Effizienz steigt somit für große k auf 1.

Für den r-nären Hamming-Code mit k Prüfstellen und der Blocklänge m gilt entsprechend (siehe (1.2-23))

$$E = 1 - \frac{k(r-1)}{r^k - 1}. \qquad \text{1.2-27}$$

Wir hatten bereits in Abschnitt 1.1 gesehen, dass, um t Fehler pro Wort korrigieren zu können, die Ungleichung $r^{m-n} \geq \sum_{i=0}^{t} \binom{m}{i} (r-1)^i$ (1.1-6)

gelten muß. Für Hamming-Codes gilt diese Ungleichung mit Gleichheitszeichen, denn für $t = 1$ erhalten wir daraus

$$r^{m-n} \geq 1 + m(r-1)$$

oder

$$r^k \geq 1 + m(r-1)$$

bzw.

$$\frac{r^k - 1}{r-1} \geq m.$$

Wie wir gesehen haben, gilt (1.2-23)

$$\frac{r^k - 1}{r-1} = m.$$

Da wir einzelne Spalten der Kontrollmatrix weglassen können, ohne die Hamming-Distanz zu verringern, können wir bei der Suche nach einem Code mit $d \geq 3$ (bzw. $t \geq 1$) für die Codierung von n Informationssymbolen wie folgt verfahren:

Zunächst bestimmen wir $k = m - n$ bei vorgegebenem n, so dass die Ungleichung (1.1-6) mit $t = \frac{d-1}{2} = 1$ erfüllt wird. Dann bestimmen wir den Hamming-Code C_k. Anschließend streichen wir so viele Spalten der Kontrollmatrix H_k, bis n Informationssymbole verbleiben. Der so erhaltene Code wird als **verkürzter Hamming-Code** bezeichnet.

verkürzter Hamming-Code

Beispiel 1.2-9:
Es ist eine binäre Codierung für 10-Informationsbits (d. h. 2^{10} Nachrichten) mit der Hamming-Distanz $d \geq 3$ gesucht. Für verschiedene k und $m = k + n$ sowie $d = 3$ erhalten wir:

k	m	2^k	$m+1$
1	11	2	12
2	12	4	13
3	13	8	14
4	14	16	15

1.2-28

Es genügen also 4 Paritätsbits, um die Ungleichung (1.1-6) mit $t = 1$ zu erfüllen, d. h. für $2^k \geq 1 + m$. Wir nehmen den Hamming-Code C_4 (s.

Beispiel 1.2-6) als Ausgangscode und streichen eine Spalte (z. B. die erste), um die Kontrollmatrix

$$K = \left[\begin{array}{cccccccccc|cccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

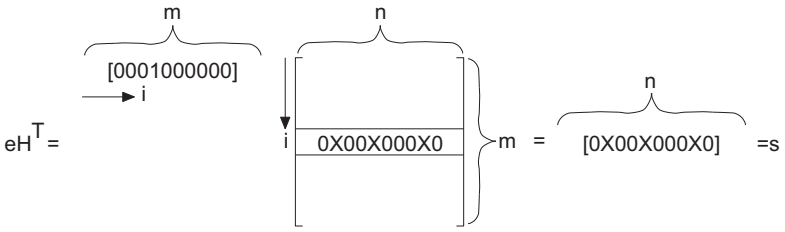
zu erhalten. K erfüllt unsere Anforderung $d \geq 3$.

Syndrom eines binären Hamming-Codes

Wir sehen uns nun das **Syndrom eines binären Hamming-Codes** etwas genauer an, und zwar bei einer fehlerhaften übertragung. $w \in C$ sei das gesendete Codewort, es liege ein einfacher Fehler an einer beliebigen Stelle vor. Der empfangene Code ist $w + e$, wobei e aus m Komponenten besteht, von denen alle bis auf eine Null sind. Wir haben

$$s = (w + e) \cdot H^T = wH^T + eH^T = eH^T \neq 0.$$

Symbolisch haben wir eine Gleichung der Form



Wir haben angenommen, dass der Fehler im i-ten Symbol auftritt, d. h. an der i-ten Stelle in e haben wir eine 1 gesetzt. Alle Elemente in der i-ten Spalte von H (i-te Zeile von H^T), die nicht gleich 0 sind, haben wir mit $\times$ bezeichnet. Das Syndrom weist genau an diesen Stellen von Null verschiedene Elemente auf. Da alle Spalten von H verschieden sind, ist damit die Stelle i genau lokalisierbar. Es ist die Spalte, die mit s identisch ist. Auf diese Weise können wir die Fehlerstelle lokalisieren und korrigieren. Falls wir H nicht in der kanonischen Form, sondern (binär) geordnet aufstellen, gibt das Syndrom die (binäre) Adresse der Fehlerstelle an.

Beispiel 1.2-10:
Wir konstruieren den Hamming-Code mit drei Kontrollstellen C_3 und ordnen die Spalten der Kontrollmatrix H_3 nach deren binärer Wertigkeit.

$$H_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$w = (\texttt{1011010})$ ist ein Codewort, denn es gilt

$$w \cdot H_3^T = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}. \quad 1.$$

Liegt ein Fehler an der 4-ten Stelle vor, d. h. $e = (\texttt{0001000})$, bzw. $v = w + e = (\texttt{1010010})$, so erhalten wir

$$s = v \cdot H^T = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}.$$

1.2-31

Es wird also die Fehlerstelle binär $\texttt{100} \equiv 4$ angezeigt.

Lägen zwei Fehler z. B. in der 4. und 5. Stelle mit $e = (\texttt{0001100})$ vor, so erhielte man als Syndrom $(\texttt{001})$. Der Fehler würde also erkannt. Falls man nicht weiß, dass ein Doppelfehler vorliegt, würde man schließen, dass ein Einfachfehler an der Stelle $\texttt{001} = 1$ vorläge. Das zeigt deutlich, dass man bei Hamming-Codes nicht gleichzeitig einen Fehler korrigieren und zwei Fehler erkennen kann, sondern man kann nur alternativ entweder zwei Fehler erkennen oder einen Fehler korrigieren - wobei im zweiten Fall schon vorher festliegen muß, dass mehr als ein Fehler nicht auftreten darf.

erweiterter Hamming-Code

Ist ein binärer Hamming-Code mit $d \geq 3$ bekannt, so ist es einfach, einen erweiterten Code, den **erweiterten** binären **Hamming-Code** C'_H mit $d \geq 4$ anzugeben. Man erweitere den Hamming-Code um eine Paritätsstelle und

bilde H', indem man H eine Spalte mit Nullen und eine weitere Zeile mit Einsen hinzufügt. Der Code C'_H mit der Kontrollmatrix

$$H' = \left[\begin{array}{cccc:cc} & & & & 0 \\ & & & & 0 \\ & H & & & 0 \\ & & & & 0 \\ \hdashline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & 1 & 1 & & 1 \end{array}\right] \qquad \text{1.2-32}$$

hat wegen der Kontruktionsvorschrift stets $d \geq 4$. Denn wie bisher sind zwei Spalten von H' stets unabhängig. Wegen der Einsen an der letzten Stelle bei allen Spalten sind auch 3 Spalten stets unabhängig.

Beispiel 1.2-11:
Wir betrachten H_3 und erweitern es zu H'_3.

$$H_3 = \left[\begin{array}{cccc:cccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \qquad \text{1.2-33}$$

$$H'_3 = \left[\begin{array}{cccc:cccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right].$$

Wir können nun auch die Kontrollmatrix in der kanonischen Form angeben

$$\tilde{H}_3 = \left[\begin{array}{cccc:cccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

und erhalten als Generatormatrix

$$\tilde{G}_3 = \left[\begin{array}{cccc:cccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right].$$

Selbsttestaufgabe 1.2-1:

a. Geben Sie die Generatormatrix G und die Prüfmatrix H eines systematischen Codes in der kanonischen Form an, und zeigen Sie den Zusammenhang zwischen ihnen auf.

b. Stellen Sie an einem Beispiel dar, wie sich ein Codewort aus einem gegebenen Informationswort und der Generatormatrix des Codes ergibt.

2 Versuchsaufbau

Der Versuch ist als Online-Versuch realisiert. Nach einer Anmeldung mit Benutzernamen und Kennwort wird eine Benutzerschnittstelle entsprechend Abb. 2-1 dargestellt. In der linken Menüleiste hat man die Möglichkeit zwischen einzelnen Versuchen zu wechseln. Die Aufgabenstellung eines Teilversuches ist unterhalb der jeweiligen Versuchsüberschrift dargestellt.

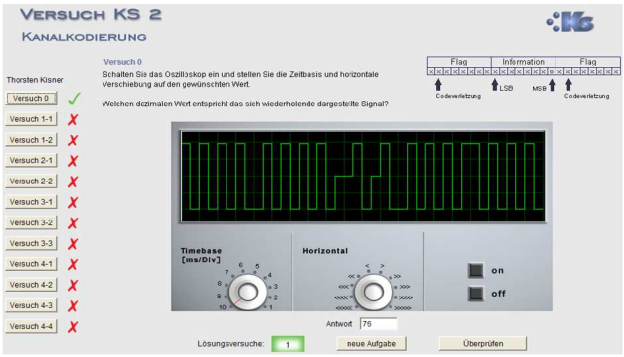


Abb. 2-1: Benutzerschnittstelle

Ist eine Aufgabe bearbeitet, kann mit dem Button `Überprüfen` eine Überprüfung der Lösung eingeleitet werden. Bei dieser Überprüfung wird die Lösung an einen Korrekturserver geschickt und dort geprüft. Ist die Antwort richtig, wird die Aufgabe als Bestanden gewertet und in der linken Menüleiste erscheint neben der Versuchsnummer ein Haken.

Eine in diesem Versuch simulierte Übertragungsstrecke prägt einem Sendesignal Störungen auf. Diese Störungen können verschiedene Ursachen haben. Beispiele sind das Rauschen der gesamten Strecke, das Nebensprechen bei mehradrigen Kabeln, das durch kapazitive oder induktive Kopplungen zwischen den Aderpaaren entsteht oder auch Stossbelastungen und Schaltvorgänge, die im Bereich der Strecke auftreten. Führt eine solche Störung zu einem Fehler, so bedeutet dies, daß bei den empfangenen Datenbits einzelne oder auch mehrere Bits umgekippt werden, d. h. im Empfänger wird auf den falschen, entgegengesetzten Zustand des Bits geschlossen. Auch bei der im Praktikumsversuch verwendeten Übertragungsstrecke treten Störungen, die Bitverfälschungen zur Folge haben, auf.

Bei der Leitungscodierung werden die diskreten Signale in zwei- oder mehrstufige Impulse für die physikalische Übertragung auf der Leitung gewandelt. Auf der Übertragungsstrecke des Versuchsaufbaus wird als Leitungscode ein sogenannter modifizierter AMI-Code (Alternate Mark Inversion) verwendet. Der ”normale” AMI-Code gehört zu der Klasse der pseudoternären Codes und hat die folgende Codiervorschrift:

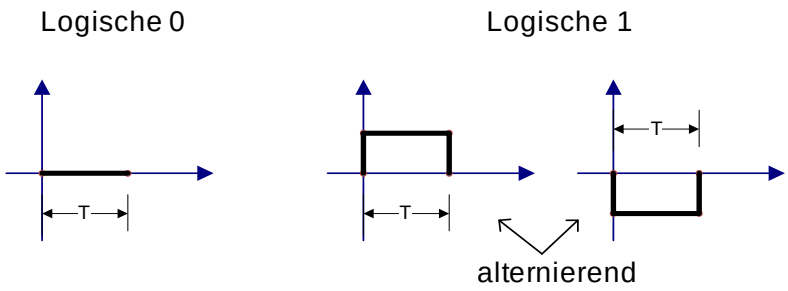


Abb. 2-2: AMI-Code

Pseudoternäre Codes sind Codes, die dreistufige (ternäre) Signale verwenden um binäre Signale zu codieren. Sie weisen deshalb stets Redundanz auf, die häufig dazu verwendet werden kann, auftretende Fehler zu erkennen. Beim AMI-Code wird die Eins alternierend als +A und -A codiert, es treten also nie +A, +A oder -A, -A (gegebenenfalls mit Nullen dazwischen) auf. Ein Fehler der +A in -A oder umgekehrt verwandelt, kann falls es sich um Einzelfehler handelt stets erkannt werden. Der AMI-Code hat keine Gleichstromkomponente und hat im allgemeinen einen ausreichenden Taktgehalt, wenn Nullfolgen vermieden werden. Der AMI-Code wird bei PCM-Systemen insbesondere PCM24 und PCM30 häufig angewandt. Auch im ISDN wird er auf der S-Schnittstelle genutzt. Hier wird ebenso wie bei dem Versuchsaufbau ein sogenannter modifizierter AMI-Code verwendet. Dies bedeutet, daß er mit umgekehrter Polarität gesendet wird, damit bei der Übertragung einer logischen Null, z. B. bei Sprachpausen, stets ±A gesendet wird.

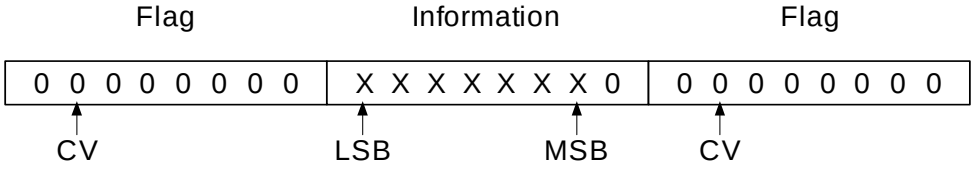
Tab. 2-1: *ASCII-Codierung*

hex	ASCII	hex	ASCII	hex	ASCII	hex	ASCII
20	Space	38	8	50	P	68	h
21	!	39	9	51	Q	69	i
22	"	3A	:	52	R	6A	j
23	CR	3B	;	53	S	6B	k
24	$	3C	<	54	T	6C	i
25	%	3D	=	55	U	6D	m
26	&	3E	>	56	V	6E	n
27	’	3F	?	57	W	6F	o
28	(	40	@	58	X	70	p
29	)	41	A	59	Y	71	q
2A	*	42	B	5A	Z	72	r
2B	+	43	C	5B	Ä	73	s
2C	’	44	D	5C	Ö	74	t
2D	-	45	E	5D	Ü	75	u
2E	.	46	F	5E	^	76	v
2F	/	47	G	5F	_	77	w
30	0	48	H	60	‘	78	x
31	1	49	I	61	a	79	y
32	2	4A	J	62	b	7A	z
33	3	4B	K	63	c	7B	ä
34	4	4C	L	64	d	7C	ö
35	5	4D	M	65	e	7D	ü
36	6	4E	N	66	f	7E	ß
37	7	4F	O	67	g		

3 Aufgabenstellung und Versuchsdurchführung

Die Versuchsdurchführung ist in mehrere Abschnitte eingeteilt. Im ersten Teil soll die digitale Übertragungsstrecke mit dem Leitungscode untersucht werden. Die Versuchsteile 1 bis 3 beschäftigen sich mit einfachen Sicherungsmassnahmen der Übertragungsstrecke und im Teil 4 werden Hamming-Codes behandelt.

Versuch 0 In diesem Versuch wird ein bestimmter ASCII-codierter Buchstabe (Tabelle 2-1) periodisch wiederkehrend übertragen. Der Beginn einer Datenübertragung wird dem Empfängerprozess über eine Codeverletzung des modifizierten AMI-Codes angezeigt. Informationen werden in folgendem Rahmen übertragen:



Auf dem Oszilloskop wird ein idealisiertes Rechecksignal dargestellt. Bei einer realen Übertragungsstrecke treten bedingt durch die physikalischen Eigenschaften der Sender- und Übertragungshardware und des Übertragungsweges Signalverformungen auf. Skizzieren Sie das so zu erwartende Signal für einen beliebigen Buchstaben. Bestimmen Sie aus dem Oszillosgraphenbild die Übertragungsgeschwindigkeit des Systems (bit/s) und den dezimalen Wert des gesendeten Buchstaben.

Versuch 1 Die Informationen werden zeilenweise übertragen und sind mit einer zeilenweisen Paritätsprüfung zu versehen. Danach ist das empfangene Signal bezüglich Fehlererkennung und -korrektur zu bewerten.

Versuch 2 Die Informationen werden nun spaltenweise übertragen, jedoch mit zeilenweiser Paritätsprüfung versehen. Paritätsprüfung und Bewertung zu Fehlererkennung und -korrektur sind in den jeweiligen Unteraufgaben durchzuführen.

Versuch 3 Nach einer zeilenweisen Übertragung sowohl mit Zeilen- als auch Spaltenparität müssen in den folgenden zwei Teilversuchen zwei empfangene Datenwörter zu Fehlererkennung und -korrektur bewertet werden.

Versuch 4 In diesem Versuch wird der Einsatz eines binären Hamming-Codes mit 4 Kontrollstellen untersucht. Zur Erzeugung der Kontrollstellen ist folgende Generatormatrix G zu verwenden:

$$G = \left[\begin{array}{ccccccc:cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right]$$

Nach dem Eintrag der Kontrollstellen als Sicherungsmaßnahme ist im nächsten Teil die zu G gehörige Kontrollmatrix H zu bestimmen.

In Teil 3 soll das Syndrom zu jeder empfangenen Zeile bestimmt werden und im letzten Teil soll aus dem Syndrom auf die Spalte geschlossen werden, in der ein Fehler aufgetaucht ist.

Untersuchen Sie bei jedem Teilversuch, ob Übertragungsfehler (Bitverfälschungen) aufgetreten sind und welche Auswirkungen sie ggf. auf den gesendeten Text haben. Bestimmen Sie, ob der aufgetretene Fehler durch das jeweilige Sicherungsverfahren erkannt und sogar ggf. korrigiert werden kann.

Der Versuch wird direkt nach der Durchführung von dem Betreuer als erfolgreich anerkannt, wenn von jedem einzelnen Teilnehmer die Lösungen der Versuchsvorbereitungsaufgaben und von der Gruppe ein gemeinsam erstelltes Protokoll der Versuchsdurchführung vorgelegt wird.

Diese Seite bleibt aus technischen Gründen frei.

4 Versuchsvorbereitung

Arbeiten Sie vorab die gesamte Versuchsbeschreibung sorgfältig durch, insbesondere sind die Ausführungen zur Kanalcodierung von zentraler Bedeutung für die Versuchsdurchführung. Wählen Sie bereits bei der Versuchsvorbereitung die Zeichenfolge aus, die Sie während der Versuchsdurchführung übertragen wollen und bestimmen Sie für diese Zeichenfolge den 7 Bit ASCII-Code (Tabelle 4.1). Bestimmen Sie für jeden Teilversuch ausgehend von den Bitfolgen die entsprechenden Sicherungsbits.

Lösen Sie die im folgenden gestellten Aufgaben mit Hilfe der Erläuterungen in dieser Versuchsbeschreibung. Die Lösungen sind vor Versuchsbeginn von allen Versuchsteilnehmern mit dem Betreuer zu besprechen.

Aufgabe 1

Welches Signal wird übertragen, wenn die Übertragungsstrecke aktiviert ist und keine Nutzdaten übertragen werden?

Aufgabe 2

Wie wird bei dem gegebenen Versuchsaufbau dem Empfänger der Beginn einer Nachricht mitgeteilt? Stellen Sie das entsprechende Signal graphisch dar.

Aufgabe 3

Eine Nachricht bestehe aus 5 alphanumerischen Zeichen, die nach dem 7 Bit ASCII-Code (Tabelle 4.1) codiert sind. Als Sicherungsmaßnahme werde die Spalten- und Zeilenparität eingesetzt. Bestimmen Sie die Anzahl der Bits, die beim Senden des Datenpakets insgesamt übertragen werden müssen. Entwerfen Sie für ein beliebiges Beispiel eine Fehlersituation und skizzieren diese in einer Tabellenschreibweise, bei der weder bei Überprüfung der Spaltenparität noch bei Überprüfung der Zeilenparität der Fehler erkannt werden kann.

Aufgabe 4

Was versteht man unter dem *Abstand zwischen zwei Codewörtern*, und welcher Zusammenhang ergibt sich mit dem Begriff *Hamming-Distanz* eines Codes?

Aufgabe 5

Bestimmen Sie für die nachfolgend aufgeführten Codewörter die Hamming-Distanz und machen Sie eine Aussage, wieviele Fehler stets erkannt werden können.

4

A	00000
B	11010
C	01101
D	10110

Aufgabe 6

Geben Sie die Generatormatrix G und die Prüfmatrix H eines systematischen Codes in der kanonischen Form an, und zeigen Sie den Zusammenhang zwischen ihnen auf.

Aufgabe 7

Stellen Sie an einem Beispiel dar, wie sich ein Codewort aus einem gegebenen Informationswort und der Generatormatrix des Codes ergibt.

A Lineare Algebra

A.1 Körper, Ringe, Gruppen

Es sei K eine Menge mit mindestens zwei Elementen, und $+$ und $\cdot$ zwei Abbildungen $(K \times K \to K)$, die wir Addition und Multiplikation nennen.

1. Ein **Körper** ist ein Tripel $(K, +, \cdot)$, für das folgende sieben Axiome gelten: Körper

   Für die Addition:

   A1 Das Assoziativgesetz: $\forall a, b, c \in K$ gilt

   $$a + (b + c) = (a + b) + c.$$

   A2 Das Kommutativgesetz: $\forall a, b \in K$ gilt

   $$a + b = b + a.$$

   A3 Existenz von Null und Inversen: Es gibt ein $n \in K$ mit

   a. n ist ein neutrales Element (Null), d. h. $\forall a \in K$ gilt

   $$a + n = n + a = a,$$

   und

   b. $\forall a \in K$ existiert ein inverses Element, d. h. $\forall a \in K \; \exists \; -a \in K$ mit

   $$a + (-a) = (-a) + a = n.$$

   Für die Multiplikation:

   M1 Das Assoziativgesetz: $\forall a, b, c \in K$ gilt

   $$a \cdot (b \cdot c) = (a \cdot b) \cdot c.$$

   M2 Das Kommutativgesetz: $\forall a, b \in K$ gilt

   $$a \cdot b = b \cdot a.$$

   M3 Existenz von Eins und Inversen: Es gibt ein $e \in K$ mit

   a. e ist ein neutrales Element (Eins), d. h. $\forall a \in K$ gilt

   $$a \cdot e = e \cdot a = a,$$

   und

   b. $\forall a \in K, a \neq n, \exists a^{-1} \in K$ mit

$$a \cdot a^{-1} = a^{-1}a = e.$$

Für die Addition und Multiplikation:

D Das Distributivgesetz: $\forall a, b, c \in K$gilt

a. $(a+b)c = ac + bc$,

b. $a \cdot (b+c) = ab + ac$[1].

Ring

2. Ein **Ring** ist ein Tripel $(R, +, \cdot)$, für das die fünf Axiome $A1 - A3$ und $M1$ und D gelten. Gilt zusätzlich $M2$, so spricht man von einem kommutativen Ring.

Gruppe

3. Eine **Gruppe** ist ein Paar $(M, +)$(additive Gruppe) oder $(M, \cdot)$(multiplikative Gruppe), für das die beiden Axiome $A1, A3$ (additive Gruppe) bzw. $M1, M3$ (multiplikative Gruppe) gelten. Gilt zusätzlich $A2$ bzw. $M2$, so spricht man von einer kommutativen Gruppe.

4. K sei ein Körper mit q Elementen.

   K ist bis auf Isomorphie durch q bestimmt.

Galoisfeld

Man nennt K**Galoisfeld**$q - ter$ Ordnung und schreibt auch $GF(q)$.

Es gibt eine Primzahl p und ein $m \in \mathbb{N}$, mit $q = p^m$. p heißt

Charakteristik

Charakteristik von K.

Unterkörper

5. Ein Körper $(K, \oplus, \odot)$ heißt **Unterkörper** des Körpers $(L, +, \cdot)$, wenn $\forall a, b \in K$ gilt

   $a.$ $K \subset L$ 1

   $b.$ $a \oplus b = a + b$

   $a \odot b = a \cdot b$

6. $GF(p)$ ist ein Unterkörper von $GF(q)$ genau dann, wenn

   $q = p^m$ für ein $m \in N$.

Ordnung eines Elementes

7. Die **Ordnung eines Elementes**α des Körpers K ist definiert als

   $$Ordnung(\alpha) = Min\{\gamma \in IN | \alpha^\gamma = 1\}.$$

   Für einen endlichen Körper mit q Elementen gilt $\alpha^{q-1} = 1$.

primitives Element

Ein Element der Ordnung $(q-1)$ nennt man ein **primitives Element**.

---

1 $D(b)$ folgt aus $M2$ und $D(a)$, bräuchte also nicht getrennt gefordert zu werden.

A.2 Vektorräume

K sei ein beliebiger Körper.

1. Ein **Vektorraum** über K ist ein Tripel $(V, +, \cdot)$, wobei V eine nichtleere Menge und $+$ und $\cdot$ zwei Abbildungen sind, Vektorraum

$$\begin{aligned} + &: \quad V \times V \to V \quad \text{Addition von Vektoren ,} \\ \cdot &: \quad K \times V \to V \quad \text{Skalare Multiplikation ,} \end{aligned}$$

für die folgende vier Axiome gelten:

V1 Das Paar $(V, +)$ ist eine kommutative Gruppe, d. h. es gelten $A1, A2, A3$ für $(V, +)$

V2 $\forall a, b \in V$ und $\alpha \in K$ gelten die Distributiv-Gesetze:

$$\begin{aligned} (\alpha + \beta)a &= \alpha a + \beta a \\ \alpha(a + b) &= \alpha a + \alpha b \end{aligned}$$

2

V3 $\forall \alpha, \beta \in K, a \in V$ gilt das Assoziativgesetz

$$\alpha(\beta a) \; = \; (\alpha\beta)a.$$

V4 Für das Einselement $e_K \in K$ und jedes $a \in V$ ist

$$e_K \cdot a \; = \; a.$$

2. $(V, +, \cdot)$ und $(U, \oplus, \odot)$ seien Vektorräume über einem beliebigen Körper K. U heißt **Untervektorraum** von V, wenn gilt: Untervektorraum

$$\begin{aligned} U1: &\quad U \subset V \\ U2: &\quad a \oplus b = a + b \qquad \forall a, b \in U \\ &\quad \alpha \odot a = \alpha \cdot a \quad \forall a \in U, \alpha \in K \end{aligned}$$

3

3. Sei V ein Vektorraum, I eine Indexmenge, $(v_i | i \in I)$ eine Familie von Vektoren aus V. Es sei

$$< v_i | i \in I >:= \left\{ \begin{array}{l} x | x \in V \text{ und es gibt eine endliche Teilmenge} \\ i \in I', \text{ sowie } \alpha_i \in K \text{ für } i \in I' \text{ mit} \\ x = \sum_{i \in I'} \alpha_i v_i \end{array} \right\}.$$

Es kann gezeigt werden, daß $< v_i | i \in I >$ ein Untervektorraum von V ist. $<>$ wird deshalb der von $(v_i | i \in I)$ erzeugte Untervektorraum genannt und $(v_i | i \in I)$ das **erzeugende System** von U. erzeugendes System

Linearkombination

4. V sei ein Vektorraum, I eine Indexmenge, $(v_i|i \in I)$ eine Familie von Vektoren aus V. Unter einer **Linearkombination** versteht man jede endliche Summe der Form

$$\sum_{i \in I'} \alpha_i v_i$$

mit $I' \subset I$ endlich und $\alpha_i \in K$ für $i \in I'$.

linear unabhängig

5. Eine Familie $(v_i|i \in I)$ von Vektoren aus V heißt **linear unabhängig** genau dann, wenn sich kein $v_i (i \in I)$ als Linearkombination der Familie $(v_j|j \in I \backslash \{i\})$ darstellen läßt.

Basis

6. Eine Familie $(v_i|i \in I)$ von Vektoren aus V heißt eine **Basis** des Vektorraumes V genau dann, wenn $(v_i|i \in I)$ ein erzeugendes System von V ist und $(v_i|i \in I)$ linear unabhängig ist.

7. Folgende Aussagen sind äquivalent:

   a. $(v_i|i \in I)$ ist eine Basis.

   b. Jedes $a \in V$ läßt sich eindeutig als Linearkombination der Familie $(v_i|i \in I)$ darstellen.

8. Jeder endlich erzeugte Vektorraum besitzt eine Basis.

9. Ist $V \neq \{0_V\}$ (Nullelement von V) ein endlich erzeugter Vektorraum über K, und ist $(v_1, \ldots v_n)$ eine Basis von V, so definiert man

$$dim_K V := n$$

Dimension

als die **Dimension** von V bezüglich K.

10. Sei V ein Vektorraum und $n \in \mathbb{N}$. Es ist $dimV = n$ genau dann, wenn es n linear unabhängige Vektoren in V gibt, $n+1$ Vektoren aus V aber immer abhängig sind.

Elementaroperationen

11. Ist eine Basis eines Vektorraumes gegeben, so erhält man durch **Elementaroperationen** an einer Basis eine neue Basis des Vektorraumes. Elementaroperationen sind: Vertauschen von Basisvektoren, Multiplikation eines Basisvektors mit einem Körperelement $\alpha \in K, \alpha \neq 0_K$, Addition eines mit einem Körperelement multiplizierten Basisvektors zu einem anderen Basisvektor.

12. Sei V ein endlich dimensionaler Vektorraum, in dem zusätzlich ein **Skalarprodukt** $u \cdot v \in K$ für beliebige Elemente $u, v \in V$ definiert ist, U ein Untervektorraum von U. Den Vektorraum

$$U^d = \{v \in V | v \cdot u = 0 \text{ für alle } u \in U\}$$

orthogonaler Vektorraum

nennt man den zu U (bezüglich V) **orthogonalen Vektorraum** ortho-

gonaler VektorraumEs gilt

$$dimU + dimU^d = dimV.$$

Index