

## 2 Grundzüge der Informations- und Codierungstheorie

### 2.1 Einleitung

### 2.2 Redundanz natürlicher Sprache

### 2.3 ISBN-Code

### 2.4 Bezeichnungen und Krafts Ungleichung

### 2.5 Huffman-Codierung

### 2.6 Entropie nach Shannon

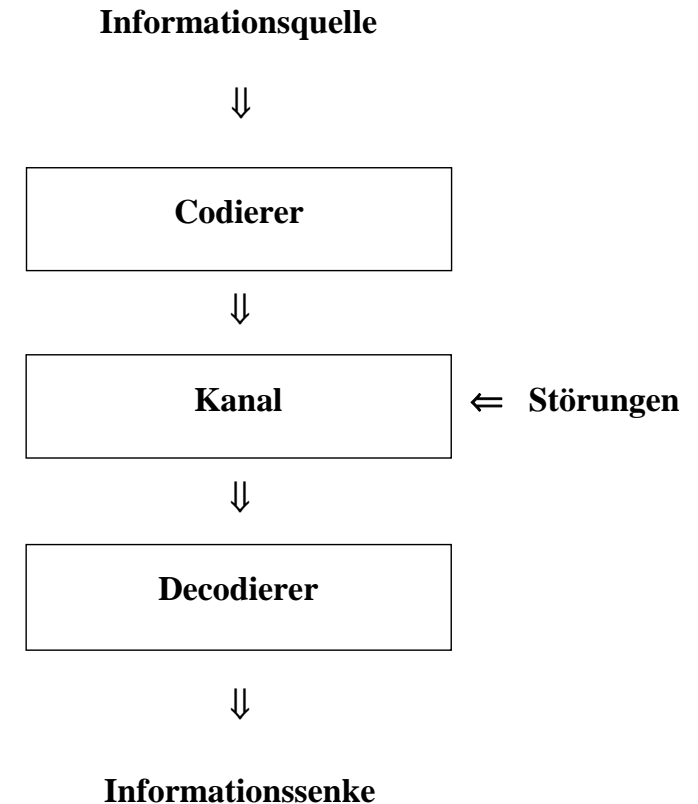
### 2.7 Fehlererkennende Codes

### 2.8 Fehlerkorrigierende Codes

### 2.9 Zyklische Codes

### 2.1 Einleitung

Grundbild der Informationsübertragung:



Bemerkungen:

- (i) Obiges Bild gilt für räumliche und zeitliche Informationsübertragung.
- (ii) Die Information, die man an der Senke erhält, ist eventuell verschieden von der Information, die die Quelle liefert.

## Ein Beispiel zur Kanalkapazität:

Betrachten wir einen unsicheren Übertragungskanal, der pro Zeiteinheit ein Zeichen übertragen kann. Der Zeichenvorrat bestehe aus den zwei Zeichen A und Z. Mit der Wahrscheinlichkeit 5% wird ein Zeichen verfälscht, die Zeichenfehler sind unabhängig voneinander. Diese Fehlerrate erscheint zu hoch. Man kann nun die Fehlerrate beim Empfänger dadurch senken, daß man jedes Zeichen dreimal hintereinander überträgt, z. B. sendet man statt der Zeichenfolge AZZAA die Zeichenfolge AAZZZZZZZAAAAAA. Der Empfänger führt eine Mehrheitsentscheidung für jedes Zeichentripel durch. Hiermit drückt man die Fehlerrate auf  $3 \cdot 25/10000 + 125/1000000 = 7625/1000000$ , eine Verbesserung etwa um den Faktor 6,5. Diese Verbesserung erkauft man durch die Senkung der Übertragungskapazität um den Faktor 3. Genügt einem diese Verbesserung nicht, dann kann man jedes Zeichen 5-mal, 7-mal, ... übertragen, dabei wird aber die nutzbare Kanalkapazität entsprechend gedrückt.

In 1948 wies nun Shannon nach, daß bei einer Senkung der Kanalkapazität auf etwa 0,7 Zeichen pro Zeiteinheit eine beliebig kleine Fehlerrate erreicht werden kann.

**Bemerkung:** Für die Berechnung der Verbesserung der Fehlerrate wurde die Formel  
$$\text{alt/neu} = \text{Verbesserungsfaktor}$$
 verwendet.

Die Informations- und Codierungstheorie sucht Fragen zu beantworten wie:

Was ist Information?

Wie unterscheidet man Information von zufälligen Zeichenfolgen?

Wie verbirgt man Information?

Wie beurteilt man die Stärke einer Chiffre?

Wie mißt man den Informationsgehalt einer Zeichenfolge?

Wie codiert man Information?

Wie entwirft man fehlererkennende Codes?

Wie entwirft man fehlerkorrigierende Codes?

Wie komprimiert man Information verlustfrei?

Wie komprimiert man Information unter Inkaufnahme tolerabler Verluste?

**Bemerkung:** Das Wort Information leitet sich vom lateinischen Wort informatio ab, dies läßt sich etwa mit Vorstellung, Begriff, Erläuterung, Darlegung, Unterricht, Belehrung übersetzen.

## 2.2 Zur Redundanz natürlicher Sprache:

**Im folgenden Text, der von Thomas Mann stammt, wurden absichtlich Buchstabenfehler eingefügt, der Text bleibt leserlich.**

Mehrefe Geschäfte weltlicheraund literarascher Natur hiehten den Reiselustizen nach etwa zwyi Wochen nach jenem tpaziergang in München zurück. Er gabwendlich Auftrag, dein Landhaus bicnen vier Wbchln zum Einruge instand usetzen und reuste an einem Tage zwlschen Mitte und Ende des Mai mit dem Nachtzuge nach Triest, goter nur viersndzwanzig Stumden verweilte und sich am nächstfolgenden Morgen nach Pola einschiffte. Was er suchtw, war das Freudartige und Beluglose, welcher jedoch rapch zu erreitten wäre? und so nahm er Aufenthalt auf einer seit einigen Jahren gerührten Insel der Adria, ungern der istriscsen Küstergelegen, mit farbig zerlumptem, in wildfremden Lauten redendem Landvolk und schön zerjissenen Klippenpprtien dort, wo das Meerhoffe war.

### Statistik:

**insgesamt: 755 Zeichen inklusive Zeilenendezeichen**  
**fehlerhaft: 38 Zeichen**  
**Fehlerquote: 5 %**

**Dies ist der gleiche Text wie auf der vorhergehenden Seite, zur besseren Lesbarkeit wurden die Fehler durch Unterstreichen markiert.**

Mehrefe Geschäfte weltlicheraund literarascher Natur hiehten den Reiselustizen nach etwa zwyi Wochen nach jenem tpaziergang in München zurück. Er gabwendlich Auftrag, dein Landhaus bicnen vier Wbchln zum Einruge instand usetzen und reuste an einem Tage zwlschen Mitte und Ende des Mai mit dem Nachtzuge nach Triest, goter nur viersndzwanzig Stumden verweilte und sich am nächstfolgenden Morgen nach Pola einschiffte. Was er suchtw, war das Freudartige und Beluglose, welcher jedoch rapch zu erreitten wäre? und so nahm er Aufenthalt auf einer seit einigen Jahren gerührten Insel der Adria, ungern der istriscsen Küstergelegen, mit farbig zerlumptem, in wildfremden Lauten redendem Landvolk und schön zerjissenen Klippenpprtien dort, wo das Meerhoffe war.

### Statistik:

**insgesamt: 755 Zeichen inklusive Zeilenendezeichen**  
**fehlerhaft: 38 Zeichen**  
**Fehlerquote: 5 %**

## Zum Vergleich das Original:

### Thomas Mann: Der Tod in Venedig, Beginn des dritten Kapitels

Mehrere Geschäfte weltlicher und literarischer Natur hielten den Reiselustigen noch etwa zwei Wochen nach jenem Spaziergang in München zurück. Er gab endlich Auftrag, sein Landhaus binnen vier Wochen zum Einzuge instandzusetzen und reiste an einem Tage zwischen Mitte und Ende des Mai mit dem Nachtzuge nach Triest, wo er nur vierundzwanzig Stunden verweilte und sich am nächstfolgenden Morgen nach Pola einschiffte. Was er suchte, war das Fremdartige und Bezuglose, welches jedoch rasch zu erreichen wäre, und so nahm er Aufenthalt auf einer seit einigen Jahren gerühmten Insel der Adria, unfern der istrischen Küste gelegen, mit farbig zerlumptem, in wildfremden Lauten redendem Landvolk und schön zerrissenen Klippenpartien dort, wo das Meer offen war.

Im folgenden Text, der aus Kafkas „Das Schloß“ stammt, wurde jeder dritte Buchstabe durch ein Leerzeichen ersetzt. Bleibt der Text leserlich?

Al si - K. e ka nt es n e ne  
We bi gu g - f st ei Wi ts au wa en,  
ar s z se ne Er ta ne sc on öl ig  
in te . Wa er o l ng fo t g we en?  
oc nu ei , zw i S un en tw na h  
s in r B re hn ng, nd m M rg n w r e  
fo tg ga ge , un ke n E se be ür ni  
ha te r g ha t, u d b s v r k rz m w r  
g ei hm ß i e T ge he le ew se , er t  
j tz di Fi st rn s. »K rz Ta e, k rz  
Ta e!« s gt er u s ch, li t v m  
S hl tt n u d g ng em ir sh us u.

Hier nun der vollständige Text.

Als sie - K. erkannte es an einer Wegbiegung - fast beim Wirtshaus waren, war es zu seinem Erstaunen schon völlig finster. War er so lange fort gewesen? Doch nur ein, zwei Stunden etwa nach seiner Berechnung, und am Morgen war er fortgegangen, und kein Essenbedürfnis hatte er gehabt, und bis vor kurzem war gleichmäßige Tageshelle gewesen, erst jetzt die Finsternis. »Kurze Tage, kurze Tage!« sagte er zu sich, glitt vom Schlitten und ging dem Wirtshaus zu.

### Bemerkung von Jean Cocteau:

"Das größte literarische Werk ist im Grunde nichts anderes als ein Alphabet in Unordnung."

Man kann nun verschiedene Auslassungen vornehmen, ohne daß obiger Text unleserlich wird.

### Streichen der Zwischenräume:

"DasgrößteliterarischeWerkistimGrundenichts anderesalseinAlphabetinUnordnung."

### Nur Kleinbuchstaben verwenden:

"dasgrößteliterarischewerkistimgrundenichts anderesalseinalphabetinunordnung."

### Streichen der Vokale:

"dsgrßltrrschwkrkstmgrndnchts ndrslsnlphbttnrdng."

Auch folgende komprimierte Aussage ist leicht lesbar.

"vrlsngnsndnchtlngwlg"

### Ziele der technischen Informationstheorie:

1. Nachrichten möglichst kompakt darstellen.
2. Nachrichten gegen Verfälschungen sichern.
3. Nachrichten für Unbefugte unleserlich machen.

## 2.3 Der ISBN-Code

ISBN = International Standard Book Number  
= Internationale Standard Buch-Nummer

Es gibt zwei Formen des ISBN-Codes, ISBN-10 und ISBN-13. Die Form des ISBN-10 wurde 1970 im Standard 2108 von der International Organization for Standardization festgeschrieben. Die ISBN-13 ist eine Anpassung an das europäische Artikel-Numerierungssystem. Zunächst wird nur die ursprüngliche Form behandelt.

Ein Beispiel: Das Buch "Pepper: Grundlagen der Informatik" trägt die ISB-Nummer 3 - 486 - 21153 - 6

### Aufbau des ISBN-Codes:

Sprachraum  
Verlag  
verlagsinterne Buchnummer  
Prüfziffer (Undezimalziffer, Zahlbasis 11)

Die Codierung eines Sprachraums oder einer Ländergruppe umfaßt eine oder mehrere Nummern, die aus einer bis fünf Ziffern nach folgendem Schema gebildet werden:

0 – 7  
80 – 94  
950 – 995  
9960 – 9989  
99900 – 99999

**Beispiele zur Vergabe von Sprachraum- und Landeskennungen für ISB-Nummern:**

<b>Kennung</b>	<b>Sprachraum oder Land</b>
<b>0, 1</b>	<b>Australien, Großbritannien, Irland, Kanada, Neuseeland, Südafrika, Swaziland, USA, Zimbabwe</b>
<b>2</b>	<b>Belgien (französisch), Kanada (französisch), Frankreich, Schweiz (französisch)</b>
<b>3</b>	<b>Österreich, Deutschland, Schweiz (deutsch)</b>
<b>7</b>	<b>China</b>
<b>82</b>	<b>Norwegen</b>
<b>88</b>	<b>Italien, Schweiz (italienisch)</b>
<b>90</b>	<b>Belgien (flämisch), Niederlande</b>
<b>92</b>	<b>Internationale Organisationen, Unesco</b>
<b>950, 987</b>	<b>Argentinien</b>
<b>977</b>	<b>Ägypten</b>
<b>978</b>	<b>Nigeria</b>
<b>9961</b>	<b>Algerien</b>
<b>9963</b>	<b>Zypern</b>
<b>99913, 99920</b>	<b>Andorra</b>

**Die Vergabe der Verlagsnummern folgt einem ähnlichen Schema wie die Vergabe der Ländernummern.**

**Beispiel für den Sprachraum 0:**

<b>00-19</b>	<b>entspricht</b>	<b>1.000.000 Titel</b>
<b>200-699</b>	<b>"</b>	<b>100.000 Titel</b>
<b>7000-8499</b>	<b>"</b>	<b>10.000 Titel</b>
<b>85000-89999</b>	<b>"</b>	<b>1.000 Titel</b>
<b>900000-949999</b>	<b>"</b>	<b>100 Titel</b>
<b>9500000-9999999</b>	<b>"</b>	<b>10 Titel</b>

**Bemerkungen:**

- (i) Die Bindestriche dienen nur der besseren Lesbarkeit, werden aber von der ISO empfohlen.**
- (ii) Eine ISB-Nummer besteht immer aus neun Dezimalziffern und einer Undezimalziffer.**
- (iii) X steht für die Undezimalziffer 10.**

**Prüfverfahren für ISB-Nummern:**

Seien die Ziffern eines ISBN-Codes  $z_{10}, z_9, \dots, z_2, z_1$ , dann nennt man einen ISBN-Code zulässig genau dann, wenn gilt

$$\left( \sum_{i=1}^{10} i \cdot z_i \right) \bmod 11 = 0$$

**Beispiel:**

Für die Buchnummer 3 - 486 - 21153 - 6 ergibt sich:

$$3*10 + 4*9 + 8*8 + 6*7 + 2*6 + 1*5 + 1*4 \\ + 5*3 + 3*2 + 6*1 = 220$$

$$\text{Prüfung: } 220 \bmod 11 = 0$$

**Berechnung einer Prüfziffer:**

Aus der Prüfformel folgt:

$$z_1 = \left( - \sum_{i=2}^{10} i*z_i \right) \bmod 11$$

Für die Anfangsnummer 3 - 499 - 14378 ergibt

sich:  $(-276) \bmod 11 = 10 = X$ ,

damit Vollnummer: 3 - 499 - 14378 - X.

**Bemerkungen:**

- (i) Die Prüfziffer schützt gegen Verfälschung einer Ziffer.
- (ii) Die Prüfziffer schützt gegen Vertauschung zweier Ziffern, dies schließt auch Zahlendreher, z.B. ..34.. nach ..43.., ein.
- (iii) Die Prüfziffer schützt gegen Falschdopplung, z.B. ..377.. nach ..337 .., dies ist ein Spezialfall von (i).

**Beweis zu (ii):** Die i-te und j-te Ziffer ( $i \neq j$ ) mögen vertauscht worden sein, dann gilt:

Prüfsumme (korrekte N.) – Prüfsumme (falsche N.)

$$= i*z_i + j*z_j - j*z_i - i*z_j = (i - j) * (z_i - z_j), \text{ wobei } z_i \neq z_j \text{ sei. Da beide Faktoren betraglich kleiner als 11 und ungleich 0 sind, ist das Produkt nicht ohne Rest durch 11 teilbar, damit ist auch die Prüfsumme der falschen Nummer nicht ohne Rest durch 11 teilbar.}$$

Ab ersten Januar 2007 werden nur noch ISBN-13 vergeben. Die ISBN-13 ist integriert in das EAN-System (EAN = European Article Number). Die EAN-13 besteht aus 12 Dezimalziffern zur Kennzeichnung einer Ware und einer Prüfziffer. Man bildet die ISBN-13 für Bücher, indem man das Präfix 978 oder 979 wählt, daran die neun Informationsziffern der ISBN-10 hängt und dann eine neue Prüfziffer berechnet.

**Algorithmus zur Berechnung der Prüfziffer einer EAN-13:**

Sei  $x_{12}x_{11}x_{10}x_9x_8x_7x_6x_5x_4x_3x_2x_1x_0$  die Ziffernfolge einer EAN-13. Dann gilt:

$$x_0 = (10 - (1*(x_{12}+x_{10}+x_8+x_6+x_4+x_2) \\ + 3*(x_{11}+x_9+x_7+x_5+x_3+x_1)) \bmod 10) \bmod 10$$

Nehmen wir als Beispiel: 978-3-528-25399-8

$$x_0 = (10 - ((9+8+5+8+5+9) + 3*(7+3+2+2+3+9)) \bmod 10) \\ \bmod 10 \\ = (10 - 122 \bmod 10) \bmod 10 = 8$$

**Beispiel: Wandlung einer ISBN-10 in eine ISBN-13:**

ISBN-10: 3-499-61210-0

ISBN-13: 978-3-499-61210-7

## 2.4 Bezeichnungen und Krafts Ungleichung

Zunächst einige Festlegungen:

Ein Zeichen ist ein Element einer endlichen Menge von wohlunterscheidbaren Dingen, dem Zeichenvorrat.

Einen Zeichenvorrat nennt man dann ein Alphabet, falls auf den Zeichen eine lineare Ordnung definiert ist.

**Bemerkung:** Die Unterscheidung von Zeichenvorrat und Alphabet wird häufig verwischt, so spricht man vom Alphabet der ASCII-Zeichen und vom Zeichenvorrat der Braille-Schrift.





Beispiele:

Alphabet der sieben römischen Zahlzeichen  
= {I, V, X, L, C, D, M}

Alphabet der 10 Dezimalziffern  
= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Alphabet der großen lateinischen Buchstaben  
= {A, B, C, D, E, F, G, H, I, J, K, L, M, N,  
O, P, Q, R, S, T, U, V, W, X, Y, Z}

Zeichenvorrat der Mondphasen

= {  ,  ,  ,  }

Zeichenvorrat der Wortsymbole von Pascal

= {and, array, begin, case, const, div, do, downto, else, end, file, for, function, goto, if, in, label, mod, nil, not, of, or, packed, procedure, program, record, repeat, set, then, to, type, until, var, while, with}

Zeichenvorrat der Binärzeichen = {0, 1}

Jedes Zeichen aus {0, 1} nennt man ein Bit (binary digit).

**Anmerkung:** Eine Gruppe von Bit nennt man ein Byte. Der Begriff wurde etwa im Juni 1956 von Werner Buchholz eingeführt und bezeichnet eine kleine Anzahl paralleler Bit, von 1 bis 6. Ein Byte diente damals zur Kennzeichnung eines Zeichens auf einem Externgerät. In 1956 wurden Externzeichen häufig durch 6 Bit codiert. Heute ist das 8-Bit-Byte weit verbreitet. Die Definition von C++ erinnert daran, daß die Zahl der Bit in einem Byte für die Zwecke der Sprache C++ vom Implementator festgelegt wird. Ursprünglich schrieb man "bite". Zitat aus der ISO-Definition von C++: "The fundamental storage unit in the C++ memory model is the byte. A byte is at least large enough to contain any member of the basic execution character set and is composed of a contiguous sequence of bits, the number of which is implementation-defined."

Zeichen aus einem Zeichenvorrat  $F$  kann man hintereinander schreiben, man erhält dann Zeichenketten oder Wörter über  $F$ . So sind  $ABABA$  und  $XYZFCD$  zwei Wörter über dem Alphabet der Großbuchstaben. Die Länge eines Wortes ist die Anzahl seiner Zeichen. Die Menge aller endlichen Wörter über einem Zeichenvorrat enthält auch ein Wort der Länge 0, das leere Wort, bezeichnet mit  $\epsilon$ .

Beispiel:

Alle Wörter der Länge drei über  $F = \{0, 1\}$  sind:  
000, 001, 010, 011, 100, 101, 110, 111.

Formal definiert man:

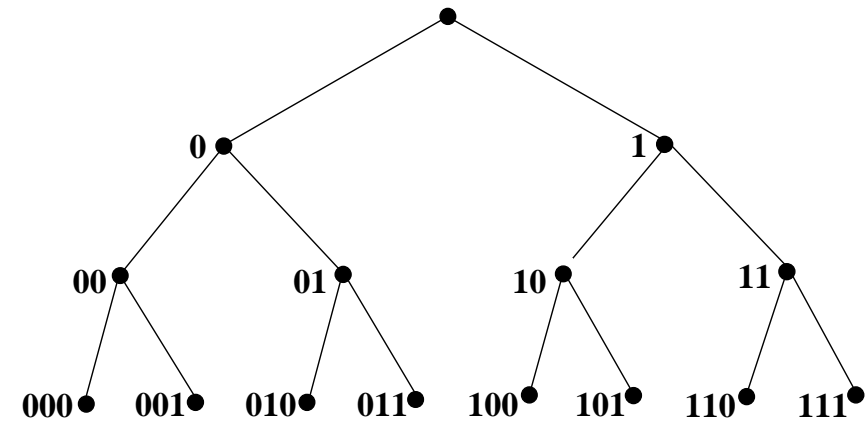
Sei  $F$  ein  $q$ -närer Zeichenvorrat mit  $q \geq 2$  und  $q$  ganzzahlig.

$F^n := \prod_{i=1}^n F$  ist dann die Menge aller Wörter über  $F$  der Länge  $n$ , wobei  $n$  eine natürliche Zahl ist.

$F^0$  sei die Menge, die nur das leere Wort enthält.

$F^* := \bigcup_{i=0}^{\infty} F^i$  ist dann die Menge aller Wörter über  $F$ .

Ein Beispiel: Sei  $F = \{0, 1\}$ ,  
Beginn des Wörterbaums für  $F^*$ :



**Definition:** Ein Code ist eine nichtleere Menge nichtleerer Wörter über einem Zeichenvorrat.

Beispiel:

$D = \{000, 100, 200, 001, 002, 101, 102, 201, 202\}$  ist ein Code über  $F = \{0, 1, 2\}$ .

Die Elemente eines Codes nennt man Codewörter.

Sei nun  $G$  ein  $u$ -närer Zeichenvorrat mit  $u \geq 1$ . Eine Codierung von  $G$  über  $F$  ist jede injektive Abbildung

$$C: G \longrightarrow F^* - \{\epsilon\}.$$

Es gilt:  $|C(G)| = |\{C(x) \mid x \in G\}| = u$ .

**Bemerkung:** Oft wird auch eine Codierung als Code bezeichnet.

**Beispiel:**

Sei  $G = \{I, V, X, L, C, D, M\}$ , sei  $F = \{0, 1, 2\}$ , dann ist

I  $\longrightarrow$  101,  
V  $\longrightarrow$  202,  
X  $\longrightarrow$  102,  
L  $\longrightarrow$  201,  
C  $\longrightarrow$  100,  
D  $\longrightarrow$  200,  
M  $\longrightarrow$  002

eine Codierung CR der römischen Zahlzeichen. Die Zahl DXXIX wird dargestellt durch 200102102101102.

Ein Code heißt Blockcode der Länge  $n$ , falls alle Codewörter die einheitliche Länge  $n$  besitzen, sonst spricht man von einem Code variabler Länge.

Blockcodes sind besonders einfach decodierbar. So kann die Zeichenfolge 002200001201 nicht die Codierung einer römischen Zahl sein, denn das Codewort 001 existiert nicht.

Man betrachte den Code  $C1 = \{0, 01, 010\}$ .

Die Zeichenfolge

$Z1 = 0010010$

kann auf mehrere Arten in Codewörter des Codes C1 zerlegt werden, z. B.:

0 01 0 01 0,  
0 010 01 0,  
0 01 0 010,  
0 010 010.

Welche Zerlegungen als korrekte anzusehen sind, muß mühsam aus dem Kontext erschlossen werden.

Hätte man statt des Codes C1 den Code  $C2 = \{0, 10, 110\}$  gewählt, dann ist jedes Wort über C2 eindeutig decodierbar; dies ist am Decodierungsbaum für C2 sichtbar.

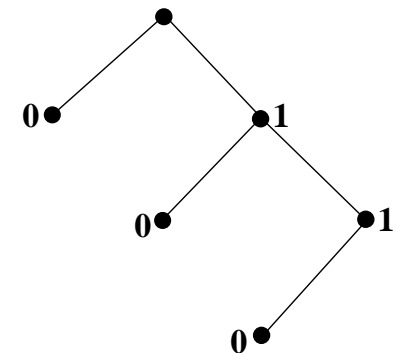
Die Zeichenkette

$Z1 = 0010010$

besitzt die eindeutige Zerlegung

$Z1 = 0 0 10 0 10.$

Decodierbaum zu C2:



**Definition:** Ein Code C heißt eindeutig decodierbar, falls

$$\begin{aligned} \text{aus } w &= c_1 c_2 c_3 \dots c_r, \\ w' &= c_1' c_2' c_3' \dots c_s', \\ w &= w' \end{aligned}$$

stets

$$r = s$$

und  $\forall i \in 1 \dots r \quad c_i = c_i'$  folgt.

**Bemerkungen:**

- (i) Jeder Blockcode ist eindeutig decodierbar.
- (ii) Der Code C2 ist eindeutig decodierbar.
- (iii) Die eindeutige Decodierbarkeit muß nicht immer so einfach zu realisieren sein wie beim Code C2. Auch der folgende Code C3 ist eindeutig decodierbar,  $C3 = \{0, 01\}$ .

**Satz von McMillan (1956):**

Sei C ein eindeutig decodierbarer s-elementiger Code mit Codelängen  $l_1 \leq l_2 \leq l_3 \dots \leq l_s$  über einem q-nären F, dann gilt:

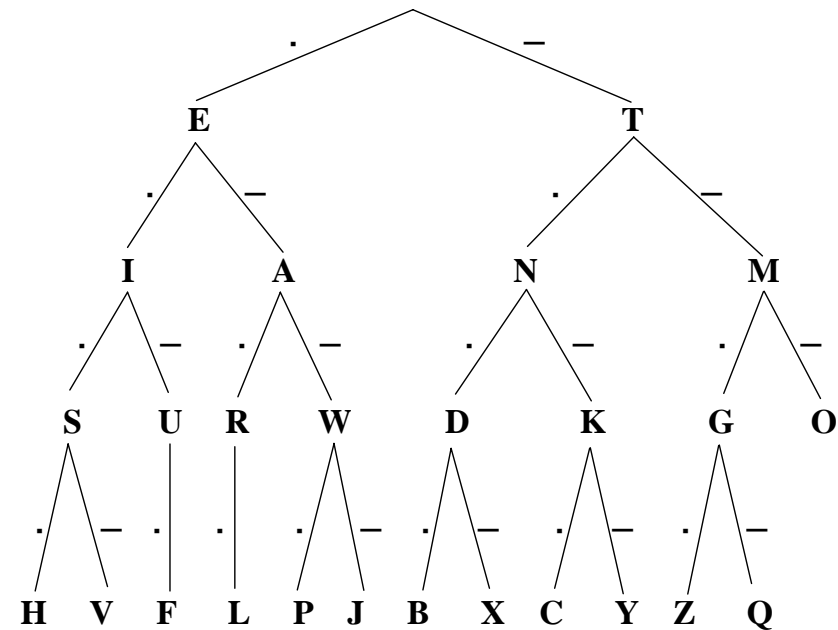
$$\sum_{i=1}^s \frac{1}{q^{l_i}} \leq 1$$

**Beispiel:**

Der Code  $\{1, 00, 01, 11\}$  ist nicht eindeutig decodierbar,

denn  $\frac{1}{2} + 3 \cdot \frac{1}{4} = 1,25$

**Teil des Morse-Codebaums:**



**Erste Nachricht über einen Telegraphen am 24. 05. 1844**  
**"What hath God wrought?"**

**Nachricht als Binärkode, kaum entzifferbar:**

.....  
 .....

**Nachricht als Ternärkode (Quaternärkode?):**

--- ..... - - ..... - - .....  
 --- ..... - - ..... - - .....  
 --- ..... - - ..... - - .....

### Krafts Ungleichung:

Für diesen Abschnitt seien gegeben:

$F$  ein  $q$ -närer Zeichenvorrat ( $q > 1$ ),

$C \subset F^*$  ein Code über  $F$ .

**Definition:** Man nennt  $C$  einen unmittelbaren Code, auch unmittelbar decodierbaren Code, falls er folgende Eigenschaft besitzt:

sei  $w = f_1 f_2 f_3 \dots f_m \in F^*$  eine Folge von Codewörtern; die einzelnen Codewörter  $c_i$ ,  $1 \leq i \leq n$ , lassen sich mittels eines sequentiellen Durchgangs durch  $w$  bestimmen, ohne ein dem Codewort  $c_i$  folgendes Zeichen von  $w$  zu betrachten.

Ein Beispiel für einen unmittelbaren Code ist

$C_4 = \{0, 10, 110, 1110, 1111\}$

**Definition:** Man nennt  $C$  einen Präfixcode, falls für alle

$a, b \in C$  mit  $a \neq b$  gilt:

$a$  ist nicht Anfangsstück von  $b$  und

$b$  ist nicht Anfangsstück von  $a$ .

**Bemerkungen:**

- (i) Die Begriffe Präfixcode und unmittelbarer Code sind Synonyme (Beweis trivial).
- (ii) Statt der Ausdrucksweise,  $C$  ist ein Präfixcode, sagt man auch,  $C$  besitzt die Präfixeigenschaft.

Unmittelbare Codes werden durch den folgenden Satz von Kraft (1949) charakterisiert.

**Satz:** Eine notwendige und hinreichende Bedingung für die Existenz eines unmittelbaren  $s$ -elementigen Codes  $C$  mit Codelängen  $l_1 \leq l_2 \leq l_3 \dots \leq l_s$  über einem  $q$ -nären  $F$  ist:

$$\sum_{i=1}^s \frac{1}{q^{l_i}} \leq 1.$$

**Beweis:** Sei  $l_s = m$ , seien  $u_i$  für  $i = 1, 2, \dots, m$  die Zahl der Codewörter der Länge  $i$ . Die Kraftsche Ungleichung läßt sich folgendermaßen umschreiben:

$$\sum_{i=1}^s \frac{1}{q^{l_i}} = \sum_{j=1}^m \frac{u_j}{q^j} = \frac{1}{q^m} * \sum_{j=1}^m u_j * q^{m-j} \leq 1$$

$$u_m + \sum_{j=1}^{m-1} u_j * q^{m-j} \leq q^m \quad (*)$$

Sei nun  $C$  ein unmittelbarer Code, sei  $a$  ein Codewort der Länge  $i$  ( $1 \leq i \leq m$ ), dann folgt aus der Präfixeigenschaft, daß wegen  $a$   $q^{m-i}$  Wörter der Länge  $m$  aus  $F^*$  nicht Codewörter sein können. Mit anderen Worten,  $a$  verbraucht  $q^{m-i}$  Wörter aus  $F^m$ . Die Summe auf der linken Seite von (\*) ist die Zahl der durch den Code  $C$  verbrauchten Wörter von  $F^m$ . Somit folgt aus der Existenz von  $C$  die Gültigkeit von (\*). Die Umkehrung dieser Argumentation zeigt, daß (\*) hinreichend ist.



Code mit verkürzten Codewörtern:

A	00
B	010
C	011
D	100
E	101
F	110
G	111

Für den letzten Code liefert Krafts Ungleichung:

$$\frac{1}{4} + 6 * \frac{1}{8} = 1$$

Krafts Ungleichung zeigt, daß kein Codewort des letzten Codes kürzbar ist.

Tabelle des American Standard Code for Information Interchange (ASCII):

$b_3b_2b_1b_0$	$b_6b_5b_4$	000	001	010	011	100	101	110	111
0 0 0 0		NUL	DLE	SP	0	@	P	'	p
0 0 0 1		SOH	DC1	!	1	A	Q	a	q
0 0 1 0		STX	DC2	"	2	B	R	b	r
0 0 1 1		ETX	DC3	#	3	C	S	c	s
0 1 0 0		EOT	DC4	\$	4	D	T	d	t
0 1 0 1		ENQ	NAK	%	5	E	U	e	u
0 1 1 0		ACK	SYN	&	6	F	V	f	v
0 1 1 1		BEL	ETB	'	7	G	W	g	w
1 0 0 0		BS	CAN	(	8	H	X	h	x
1 0 0 1		HT	EM	)	9	I	Y	i	y
1 0 1 0		LF	SUB	*	:	J	Z	j	z
1 0 1 1		VT	ESC	+	;	K	[	k	{
1 1 0 0		FF	FS	,	<	L	\	l	
1 1 0 1		CR	GS	-	=	M	]	m	}
1 1 1 0		SO	RS	.	>	N	^	n	~
1 1 1 1		SI	US	/	?	O	_	o	DEL

**Zeichentabelle nach DIN 66003:**

$b_3b_2b_1b_0$	$b_6b_5b_4$	000	001	010	011	100	101	110	111
0 0 0 0		NUL	(TC <sub>7</sub> ) DLE	SP	0	S	P	'	p
0 0 0 1		(TC <sub>1</sub> ) SOH	DC1	!	1	A	Q	a	q
0 0 1 0		(TC <sub>2</sub> ) STX	DC2	"	2	B	R	b	r
0 0 1 1		(TC <sub>3</sub> ) ETX	DC3	#	3	C	S	c	s
0 1 0 0		(TC <sub>4</sub> ) EOT	DC4	\$	4	D	T	d	t
0 1 0 1		(TC <sub>5</sub> ) ENQ	(TC <sub>8</sub> ) NAK	%	5	E	U	e	u
0 1 1 0		(TC <sub>6</sub> ) ACK	(TC <sub>9</sub> ) SYN	&	6	F	V	f	v
0 1 1 1		BEL	(TC <sub>10</sub> ) ETB	'	7	G	W	g	w
1 0 0 0		FE <sub>0</sub> (BS)	CAN	(	8	H	X	h	x
1 0 0 1		FE <sub>1</sub> (HT)	EM	)	9	I	Y	i	y
1 0 1 0		FE <sub>2</sub> (LF)	SUB	*	:	J	Z	j	z
1 0 1 1		FE <sub>3</sub> (VT)	ESC	+	;	K	Ä	k	ä
1 1 0 0		FE <sub>4</sub> (FF)	IS <sub>4</sub> (FS)	,	<	L	Ö	l	ö
1 1 0 1		FE <sub>5</sub> (CR)	IS <sub>3</sub> (GS)	-	=	M	Ü	m	ü
1 1 1 0		SO	IS <sub>2</sub> (RS)	.	>	N	ß	n	ß
1 1 1 1		SI	IS <sub>1</sub> (US)	/	?	O	DEL	o	DEL

**Abkürzungen:**

<b>SOH</b>	<b>Start Of Heading</b>
<b>STX</b>	<b>Start Of Text</b>
<b>ETX</b>	<b>End Of Text</b>
<b>EOT</b>	<b>End Of Transmission</b>
<b>ENQ</b>	<b>Enquiry</b>
<b>ACK</b>	<b>Acknowledge</b>
<b>BS</b>	<b>Backspace</b>
<b>HT</b>	<b>Horizontal Tab</b>
<b>LF</b>	<b>Line Feed</b>
<b>VT</b>	<b>Vertical Tab</b>
<b>FF</b>	<b>Form Feed</b>
<b>CR</b>	<b>Carriage Return</b>
<b>SO</b>	<b>Shift Out</b>
<b>SI</b>	<b>Shift In</b>
<b>DLE</b>	<b>Data Link Escape</b>
<b>DCi</b>	<b>Device Control i (1 ≤ i ≤ 4)</b>
<b>NAK</b>	<b>Negative Acknowledge</b>
<b>SYN</b>	<b>Synchronous Idle (SYNC)</b>
<b>ETB</b>	<b>End Of Transmission Block</b>
<b>CAN</b>	<b>Cancel</b>
<b>EM</b>	<b>End Of Medium</b>
<b>SUB</b>	<b>Substitute</b>
<b>ESC</b>	<b>Escape</b>
<b>FS</b>	<b>File separator</b>
<b>GS</b>	<b>Group separator</b>
<b>RS</b>	<b>Record separator</b>
<b>US</b>	<b>Unit separator</b>
<b>DEL</b>	<b>Delete</b>
<b>TC</b>	<b>Transmission Control</b>
<b>IS</b>	<b>Information Separator</b>
<b>FE</b>	<b>Format Effector</b>
<b>SP</b>	<b>Space</b>

## 2.5 Huffman-Codierung

**Aufgabe:** Codierung von Nachrichten über einem gegebenen Alphabet als Bitketten, wobei die relativen Häufigkeiten der Alphabetsymbole bekannt sind.

**Illustration an einem Beispiel:**

Alphabet = { E, I, N, S, D, L, R }

relative Häufigkeiten:

E:	18
I:	10
D:	2
L:	5
N:	6
R:	4
S:	7

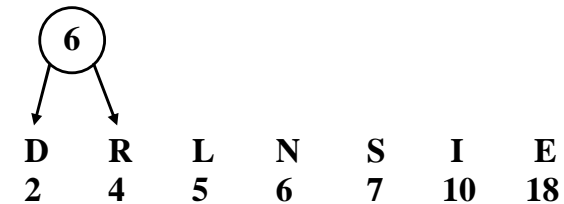
**Bemerkung:** Die Huffman-Codierung ist optimal unter der Randbedingung, daß jedes Alphabetsymbol durch eine eindeutige Bitkette codiert wird.

**Bildung eines Huffman-Baumes:**

**Anordnen der Symbole nach aufsteigenden Häufigkeiten:**

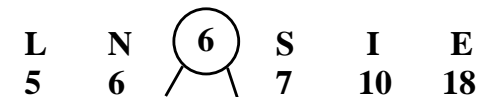
D	R	L	N	S	I	E
2	4	5	6	7	10	18

**Zusammenfassen zweier Symbole kleinster Häufigkeit zu einem neuen Symbol:**



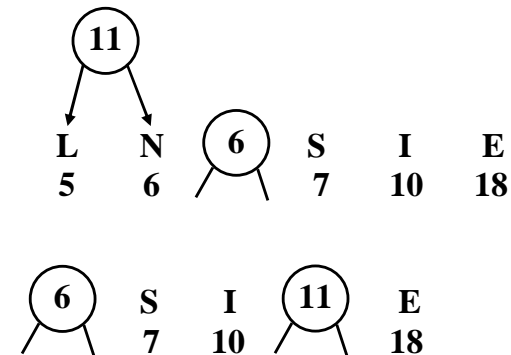
D	R	L	N	S	I	E
2	4	5	6	7	10	18

**Einordnen des neuen Symbols in die Symbolfolge:**

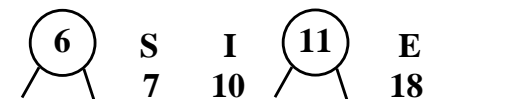


L	N	S	I	E
5	6	7	10	18

**Wiederholen der Schritte Auswählen und Einordnen führt zu:**

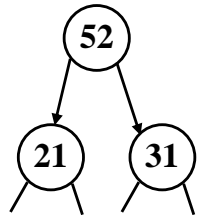
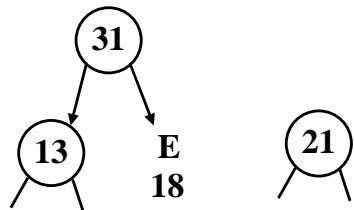
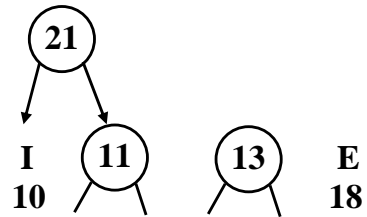
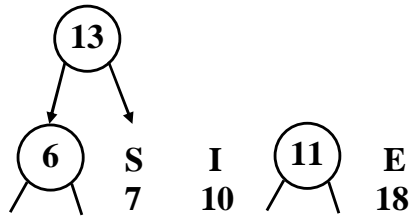


L	N	S	I	E
5	6	7	10	18

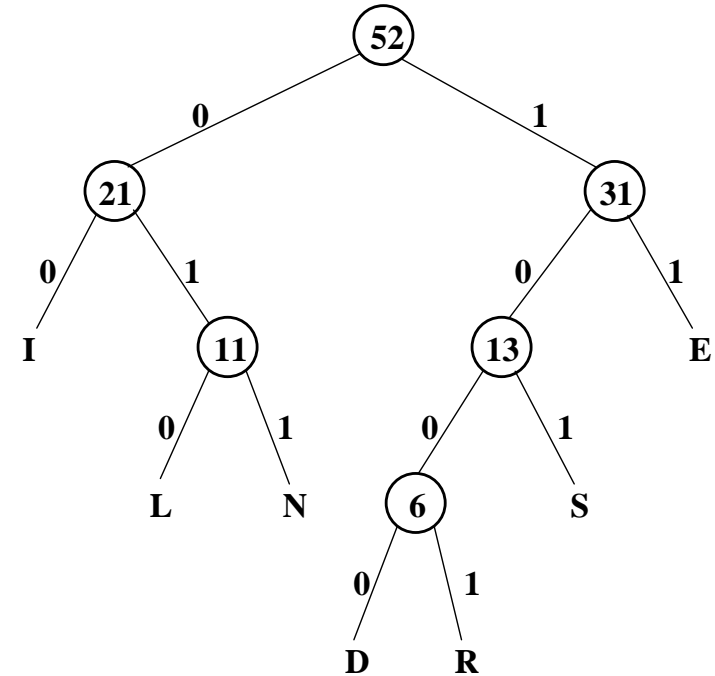
S	I	E
7	10	18

**Bildung eines Huffman-Baumes, Fortsetzung:**



**Reduktion beendet.**

**Ein vollständiger Codierbaum:**



**Beispiel zur Codierung:**

**E I N S I E D L E R**  
**1 1 0 0 1 1 1 0 1 0 0 1 1 1 0 0 0 0 1 0 1 1 1 0 0 1**

**Beispiel zur Decodierung:**

**1 0 0 1 0 0 1 1 1 0 1 1 1**  
**R I E S E**

## Huffman-Codierung: Allgemeine Formulierung

Den  $s$  Codesymbolen  $c_1, c_2, \dots, c_s$  seien die Wahrscheinlichkeiten  $p_1 \geq p_2 \geq p_3 \geq \dots \geq p_s$  mit  $\sum_{i=1}^s p_i = 1$  zugeordnet.

Für die folgenden Betrachtungen beschränken wir uns auf das Binäralphabet  $\{0, 1\}$ ; eine Erweiterung der Aussagen auf einen  $q$ -nären Zeichenvorrat läßt sich einfach bewerkstelligen.

**Aufgabe:** Bildung eines unmittelbaren Codes minimaler

$$\text{durchschnittlicher Länge } L = \sum_{i=1}^s p_i l_i,$$

wobei  $l_i$  die Länge der Codierung von  $c_i$  ist.

**Bemerkung (i):**

Für einen Code minimaler durchschnittlicher Codelänge gilt:  $l_1 \leq l_2 \leq l_3 \dots \leq l_s$ .

**Beweis indirekt:** Für einen Code minimaler durchschnittlicher Codelänge mögen  $i$  und  $j$  existieren mit

$0 < i < j \leq s$ ,  $l_i > l_j$  und  $p_i > p_j$ , dann gilt:

$$p_i * l_j + p_j * l_i - (p_i * l_i + p_j * l_j)$$

$$= p_i * (l_j - l_i) - p_j * (l_j - l_i)$$

$$= (p_i - p_j) * (l_j - l_i) < 0, \text{ Widerspruch!}$$

**Bemerkung (ii):**

Für einen unmittelbaren Code minimaler durchschnittlicher Codelänge gilt:  $l_s = l_{s-1}$ .

**Beweis:**

Sei  $x = a_1 a_2 a_3 \dots a_m$  das wegen Bem. (i) einzige Codewort mit Länge  $l_s = m$ ; aufgrund der Präfixeigenschaft ist  $y = a_1 a_2 a_3 \dots a_{m-1}$  kein Codewort und auch nicht das Anfangsstück eines Codeworts. Das Codewort  $x$  ist ersetzbar durch  $y$ ; dies steht im Widerspruch zur Minimalität der mittleren Codewortlänge.

**Verfahren zur Bildung des Huffman-Codes:**

Es sei angenommen, daß die Zahl der Codesymbole mindestens drei beträgt. Man faßt zwei Symbole mit den geringsten Wahrscheinlichkeiten  $p_s$  und  $p_{s-1}$  zu einem neuen Symbol mit der Wahrscheinlichkeit  $p_s + p_{s-1}$  zusammen. Hierdurch wird das ursprüngliche Problem auf ein Problem mit  $s-1$  Codesymbolen reduziert. Man wiederholt diese Reduktion um jeweils ein Symbol bis nur noch zwei Symbole vorhanden sind. Diesen ordnet man die Codewörter 0 und 1 zu. Durch anschließende Rückgängigmachung der Reduktionen unter gleichzeitiger Verlängerung der Codewörter um 0 und 1 gewinnt man den vollen Code.

Ein weiteres Beispiel:

Symbol:  $c_1$   $c_2$   $c_3$   $c_4$   $c_5$   
 $p_i$  ( $1 \leq i \leq 5$ ): 0,4 0,2 0,2 0,1 0,1

Bildung neuer Symbole:

neua =  $c_4$  vereinigt  $c_5$  mit  $p_{\text{neua}} = 0,2$   
neub =  $c_3$  vereinigt neua mit  $p_{\text{neub}} = 0,4$   
neuc =  $c_2$  vereinigt neub mit  $p_{\text{neuc}} = 0,6$

Es sind nur noch die beiden Symbole  $c_1$  und neuc vorhanden. Man ordnet 0  $c_1$  und 1 neuc zu. Expansion von neuc ergibt die Codierungen  $c_2 = 10$  und neub = 11, Fortführung der Expansion führt zu  $c_3 = 110$  und neua = 111 und schließlich zu  $c_4 = 1110$  und  $c_5 = 1111$ .

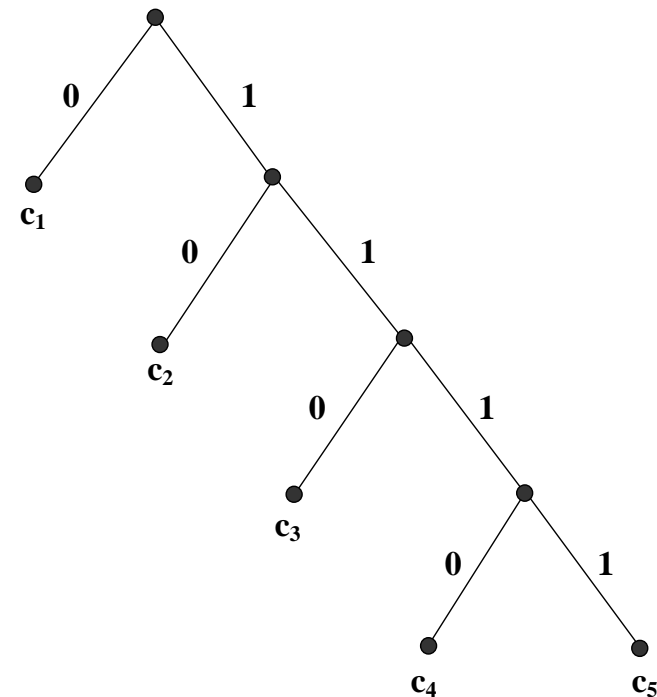
Code H1:  $c_1 = 0$   
 $c_2 = 10$   
 $c_3 = 110$   
 $c_4 = 1110$   
 $c_5 = 1111$

Bemerkung: Die durch Huffmans Verfahren erhaltene Codierung ist nicht eindeutig. Hätte man im zweiten Reduktionsschritt die Symbole  $c_2$  und  $c_3$  zusammengefaßt, dann wären die Codes H2 oder H3 entstanden. Sie haben die gleiche mittlere Codelänge wie H1.

Code H2:  $c_1 = 00$   
 $c_2 = 10$   
 $c_3 = 11$   
 $c_4 = 010$   
 $c_5 = 011$

Code H3:  $c_1 = 0$   
 $c_2 = 100$   
 $c_3 = 101$   
 $c_4 = 110$   
 $c_5 = 111$

Codebaum zu H1:



mittlere Codelänge  
 $= 0,4 + 0,2 \cdot 2 + 0,2 \cdot 3 + 0,1 \cdot 4 + 0,1 \cdot 4 = 2,2$

Es bleibt zu zeigen: Der durch Huffmans Algorithmus entstandene Code hat eine minimale durchschnittliche Codelänge.

Man führt den Beweis indirekt.

Sei  $C$  ein Huffman-Code mit durchschnittlicher Codelänge  $L$ ; sei  $D$  ein weiterer unmittelbarer Code zur gleichen Symbolmenge mit durchschnittlicher Codelänge  $M$ ,  $M < L$  und  $M$  minimal.

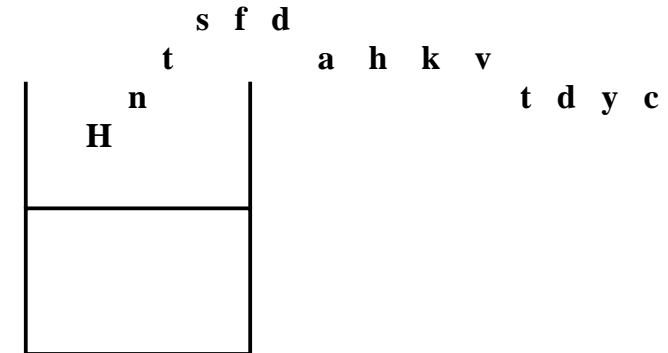
Man betrachte die  $C$  und  $D$  zugeordneten Decodierbäume  $A$  und  $B$ . Beide Bäume haben keine "toten Zweige". Man betrachte jeweils die Endknoten, die zwei Symbolen geringster Wahrscheinlichkeit zugeordnet sind. Durch längeninvariante Umbenennung erreicht man, daß diese beiden Endknoten einen jeweils gleichen unmittelbaren Vorgängerknoten haben. Dem jeweiligen Vorgängerknoten gibt man das Gewicht  $p_{s-1} + p_s$ , die Endknoten streicht man. Damit hat man die Codelänge jeweils um den festen Betrag  $p_{s-1} + p_s$  reduziert.

Eine Fortsetzung des Verfahrens führt dazu, daß sich der Baum  $C$  auf einen Baum mit durchschnittlicher Codelänge 1 reduziert und der Baum  $D$  auf einen mit durchschnittlicher Codelänge  $< 1$ . Dies ist nicht möglich. q.e.d.

**Bemerkung:** Einen Algorithmus zur Erzeugung von Huffman-Codes programmiert man direkt auf Baumstrukturen. Bei der Programmierung sollte man darauf achten, einen Code minimaler Varianz zu erhalten. Für die Codes  $H_1$ ,  $H_2$  und  $H_3$  sind die Varianzen 1,36, 0,1 und 0,96.

## 2.6 Entropie nach Shannon

Gegeben sei eine Quelle für einen Zeichenstrom:



Eigenschaften der Quelle:

- (i) Die Zahl der verschiedenen Zeichen ist endlich.
- (ii) Jedes Zeichen erscheint im Zeichenstrom mit einer festen Wahrscheinlichkeit.
- (iii) Das Erscheinen eines Zeichens ist ein unabhängiges Ereignis.

Frage nach der Unbestimmtheit des nächsten Zeichens:

Parameter:

$n$  = Zahl der verschiedenen Zeichen,

$p_i$  ( $1 \leq i \leq n$ ) = Wahrscheinlichkeit des  $i$ -ten Zeichens

mit  $\sum_{i=1}^n p_i = 1$ .

**Man setzt: Eigenschaften einer Unbestimmtheit  
= Eigenschaften eines Informationsgewinns.**

**Bezeichnung für Unbestimmtheit:  $H(p_1, p_2, \dots, p_n)$**

**Bemerkung: Die Unbestimmtheit ist nur durch die Wahrscheinlichkeitsverteilung charakterisiert.**

**Als Maß für die Unbestimmtheit formulierte Claude Shannon 1948:**

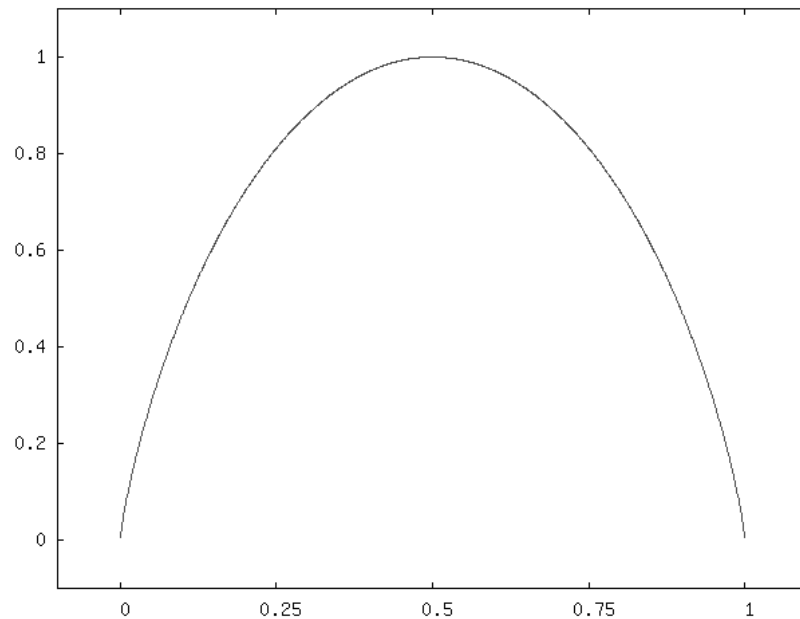
$$(*) \quad H(p_1, \dots, p_n) = - \sum_{p_k \neq 0} p_k * \log p_k$$

**Bemerkung: Claude Shannon nannte die Größe  $H(\dots)$  Entropie. In obiger Formel ist die Basis des Logarithmus nicht festgelegt, benutzt man den Zweierlogarithmus, dann mißt man die Entropie in bits als Abkürzung für binary units, benutzt man den natürlichen Logarithmus, dann mißt man die Entropie in nats von natural units, benutzt man den dekadischen Logarithmus, dann mißt man die Entropie in dets von decimal units. Im folgenden werden wir meistens den Zweierlogarithmus verwenden.**

**Die Entropie besitzt eine Reihe schöner Eigenschaften.**

- (E1)  $H(p_1, p_2, \dots, p_n)$  ist maximal,  
falls  $p_i = 1/n$  ( $1 \leq i \leq n$ )
- (E2)  $H$  ist symmetrisch:  
Für jede Permutation  $\pi$  von  $1, 2, \dots, n$  gilt:  
 $H(p_1, p_2, \dots, p_n) = H(p_{\pi(1)}, p_{\pi(2)}, \dots, p_{\pi(n)})$
- (E3)  $H(p_1, p_2, \dots, p_n) \geq 0$   
mit  $H(0, \dots, 0, 1, \dots, 0) = 0$
- (E4)  $H(p_1, p_2, \dots, p_n, 0) = H(p_1, p_2, \dots, p_n)$
- (E5)  $H(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}) \leq H(\frac{1}{n+1}, \frac{1}{n+1}, \dots, \frac{1}{n+1})$
- (E6)  $H$  ist stetig in seinen Argumenten.
- (E7) Additivität: seien  $n, m \in \mathbb{N}^+$   
 $H(\frac{1}{n \cdot m}, \frac{1}{n \cdot m}, \dots, \frac{1}{n \cdot m})$   
 $= H(\frac{1}{n}, \dots, \frac{1}{n}) + H(\frac{1}{m}, \dots, \frac{1}{m})$
- (E8) Sei  $p = p_1 + p_2 + \dots + p_n$ ,  
sei  $q = q_1 + q_2 + \dots + q_m$   
mit  $p + q = 1$ ,  $p > 0$ ,  $q > 0$   
 $H(p_1, p_2, \dots, p_n, q_1, q_2, \dots, q_m) = H(p, q) +$   
 $p * H(\frac{p_1}{p}, \dots, \frac{p_n}{p}) + q * H(\frac{q_1}{q}, \dots, \frac{q_m}{q})$

**Bemerkung:** Claude Shannon faßte die Eigenschaften E1 bis E8 als Axiome zur Bestimmung eines Informationsmaßes auf. Er leitete daraus die Formel (\*) ab. Man kann es auch so formulieren, die Entropie beschreibt den mittleren Informationsgehalt eines Zeichens einer stationären Nachrichtenquelle.



**Veranschaulichung der Entropie-Funktion für einen Zweizeichenvorrat 0 und 1 mit den Wahrscheinlichkeiten  $p$  und  $1-p$ . Man sieht, daß das Maximum der Entropie bei  $p = 0,5$  liegt. Bei  $p = 0$  oder  $p = 1$  ist die Entropie null.**

**Betrachten wir eine stationäre Nachrichtenquelle mit den Zeichen  $c_i$  ( $1 \leq i \leq 5$ ) und den zugeordneten Wahrscheinlichkeiten  $p_i$  ( $1 \leq i \leq 5$ ):**

Symbol:	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
$p_i$ ( $1 \leq i \leq 5$ ):	0,4	0,2	0,2	0,1	0,1

**Als Entropie  $H$  der Nachrichtenquelle berechnet man  $H = 2,12193$ ; dieser Wert ist etwas geringer als die mittlere Codelänge von 2,2 eines Huffman-Codes für diese Quelle.**

**Satz:** Sei  $S(p_1, p_2, \dots, p_n)$  eine stationäre, gedächtnislose Nachrichtenquelle, sei  $H$  ihre Entropie, sei  $L$  die durchschnittliche Länge einer unmittelbaren binären Codierung des Zeichenvorrats von  $S$ , dann gilt:

$$H \leq L.$$

**Beweis:**

Seien  $l_1, l_2, \dots, l_n$  die Längen der Codewörter. Dann gilt:

$$\begin{aligned} H - L &= - \sum_{j=1}^n p_j \cdot \log_2(p_j) - \sum_{j=1}^n p_j \cdot l_j \\ &= \frac{1}{\ln 2} * \sum_{j=1}^n p_j * \ln \frac{2^{-l_j}}{p_j} \end{aligned}$$

Da  $\ln(x) \leq x-1$  für  $x > 0$ , gilt

$$\begin{aligned} H - L &\leq \frac{1}{\ln 2} * \sum_{j=1}^n p_j * \left( \frac{2^{-l_j}}{p_j} - 1 \right) \\ &= \frac{1}{\ln 2} * \sum_{j=1}^n (2^{-l_j} - p_j) \end{aligned}$$

Der Wert der letzten Summe ist wegen Krafts Ungleichung kleiner oder gleich Null. Damit

$$H \leq L.$$

**Der Codierungssatz von Shannon gestattet es, die Güte einer Codierung abzuschätzen.**

**Beispiel:** Sei eine Viersymbolquelle  $Q$  gegeben. Die Wahrscheinlichkeiten seien: 0,4; 0,3; 0,2; 0,1.

Symbol	Wahrscheinlichkeit $p$	$p \cdot \log_2(1/p)$
s1	0,4	0,52877
s2	0,3	0,52109
s3	0,2	0,46439
s4	0,1	0,33219
Summe		1,84644

**Die Normcodierung von 4 Elementen mittels Bitketten hat eine durchschnittliche Länge von 2.**

**Bemerkung:** Die Größen  $\log_2(1/p_i)$  für  $0 < i \leq n$  für eine stationäre Quelle geben einen ersten Hinweis auf die Länge eines entsprechenden Codeworts.

**Beispiel:**

$$\begin{aligned} \log_2(1/0,4) &= 1,32193 \\ \log_2(1/0,3) &= 1,73697 \\ \log_2(1/0,2) &= 2,32193 \\ \log_2(1/0,1) &= 3,32193 \end{aligned}$$

In erster Näherung lassen sich auch deutschsprachige Texte so modellieren, als ob sie von einer stationären Nachrichtenquelle stammen. Für ein deutsches Alphabet von 30 Zeichen erhält man eine Entropie von 4,11 und für ein Alphabet von 26 Zeichen eine Entropie von 4,07.

Schätzwerte für Häufigkeiten der Buchstaben in der deutschen Schriftsprache für 30-Zeichen-Alphabet und 26-Zeichen-Alphabet:

Buchstabe	Wahrscheinlichkeit Zeichenzahl		Buchstabe	Wahrscheinlichkeit Zeichenzahl	
	30	26		30	26
ZI*)	0,1515		o	0,0177	0,0298
e	0,1470	0,1748	b	0,0160	0,0193
n	0,0884	0,0984	z	0,0142	0,0114
r	0,0686	0,0754	w	0,0142	0,0148
i	0,0638	0,0773	f	0,0136	0,0165
s	0,0539	0,0683	k	0,0096	0,0146
t	0,0473	0,0613	v	0,0074	0,0094
d	0,0439	0,0483	ü	0,0058	
h	0,0436	0,0423	p	0,0050	0,0096
a	0,0433	0,0647	ä	0,0049	
u	0,0319	0,0417	ö	0,0025	
l	0,0293	0,0349	j	0,0016	0,0027
c	0,0267	0,0268	y	0,0002	0,0008
g	0,0267	0,0306	q	0,0001	0,0002
m	0,0213	0,0258	x	0,0001	0,0004

\*) ZI = Zwischenraum und Interpunktionszeichen

Informationsgehalt der deutschen Buchstaben:

Buchstabe	Wahrscheinlichkeit w	$\log_2(1/w)$
ZI	0,1515	2,72261
e	0,147	2,76611
n	0,0884	3,49981
r	0,0686	3,86565
i	0,0638	3,9703
s	0,0539	4,21357
t	0,0473	4,40202
d	0,0439	4,50964
h	0,0436	4,51953
a	0,0433	4,52949
u	0,0319	4,9703
l	0,0293	5,09296
c	0,0267	5,22702
g	0,0267	5,22702
m	0,0213	5,553
o	0,0177	5,82011
b	0,016	5,96578
z	0,0142	6,13797
w	0,0142	6,13797
f	0,0136	6,20025
k	0,0096	6,70275
v	0,0074	7,07826
ü	0,0058	7,42973
p	0,0050	7,64386
ä	0,0049	7,673
ö	0,0025	8,64386
j	0,0016	9,28771
y	0,0002	12,2877
q	0,0001	13,2877
x	0,0001	13,2877

**Beispiel eines Experimentes von Shannon zur Bestimmung der Entropie der englischen Sprache:**

**Raten des Folgebuchstabens:**

T H E R E I S N O R E V E R S E O N  
 1 1 1 5 1 1 2 1 1 2 1 1 15 1 17 1 1 1 2 1 3 2 1

A M O T O R C Y C L E A F R I E N D  
 2 2 7 1 1 1 1 4 1 1 1 1 1 3 1 8 6 1 3 1 1 1

O F M I N E F O U N D T H I S O U T  
 1 1 1 1 1 1 1 1 6 2 1 1 1 1 1 1 2 1 1 1 1 1 1

R A T H E R D R A M A T I C A L L Y  
 4 1 1 1 1 1 1 11 5 1 1 1 1 1 1 1 1 1 1 1 1

T H E O T H E R D A Y  
 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

**Es treten folgende Häufigkeiten der einzelnen Zahlen auf:**

1 – 79,      2 – 8,      4 – 3,      4 – 2,  
 5 – 2,      6 – 3,      7 – 1,      8 – 1,  
 11 – 1,      15 – 1,      17 – 1.

Der Versuchstext umfaßt 102 Zeichen, approximierte Wahrscheinlichkeiten sind daher 79/102, 8/102, ...  
 Man berechnet eine obere Schranke für die Entropie des Englischen zu

$$H(\text{Englisch}) = 1,42$$

**Erweiterung einer Informationsquelle:**

Betrachten wir eine Quelle  $Q = \{z_1, z_2\}$  mit  $w_1 = 0,25$  und  $w_2 = 0,75$ .

Statt nur Einzelsymbole zu betrachten, kann man auch Symbolpaare, Symboltripel, u.s.w. betrachten. Man schreibt

$$Q^2 = \{z_1z_1, z_1z_2, z_2z_1, z_2z_2\}$$

$$Q^3 = \{z_1z_1z_1, z_1z_1z_2, z_1z_2z_1, z_1z_2z_2, z_2z_1z_1, z_2z_1z_2, z_2z_2z_1, z_2z_2z_2\}$$

Für die entsprechenden Wahrscheinlichkeiten gilt:

$$W(z_1z_1) = 0,0625; \quad W(z_1z_2) = 0,1875;$$

$$W(z_2z_1) = 0,1875; \quad W(z_2z_2) = 0,5625.$$

$$W(z_1z_1z_1) = 0,015625; \quad W(z_1z_1z_2) = 0,046875;$$

$$W(z_1z_2z_1) = 0,046875; \quad W(z_1z_2z_2) = 0,140625;$$

$$W(z_2z_1z_1) = 0,046875; \quad W(z_2z_1z_2) = 0,140625;$$

$$W(z_2z_2z_1) = 0,140625; \quad W(z_2z_2z_2) = 0,421875.$$

Die Entropien berechnen sich zu

$$H(Q) = 0,811278;$$

$$H(Q^2) = 1,62256;$$

$$H(Q^3) = 2,43383.$$

Es gilt:  $H(Q) = H(Q^2) / 2 = H(Q^3) / 3.$

Die letzte Aussage gilt allgemein.

Sei  $S = (Q, W)$  eine Informationsquelle mit Zeichenvorrat  $Q$  und Wahrscheinlichkeitsverteilung  $W$ . Für  $n \geq 1$  nennt man  $S^n = (Q^n, W^n)$  die  $n$ -te Erweiterung der Quelle  $S$ , wobei  $Q^n$  die Menge aller Wörter der Länge  $n$  über  $Q$  sei und  $W^n$  durch  $W(w) = W(q_1 q_2 \dots q_n) = W(q_1) \cdot \dots \cdot W(q_n)$  definiert sei.

Es gilt:  $H(S^n) = n \cdot H(S)$ .

**Beweis:**

Man rechnet aus:

$$\begin{aligned}
 H(S^n) &= - \sum_{\substack{i_1, i_2, \dots, i_n \\ 0 \leq i_k \leq q}} w_{i_1} \cdot w_{i_2} \cdot \dots \cdot w_{i_n} \cdot \log_2(w_{i_1} \cdot \dots \cdot w_{i_n}) \\
 &= - \sum_{\substack{i_1, i_2, \dots, i_n \\ 0 \leq i_k \leq q}} w_{i_1} \cdot \dots \cdot w_{i_n} \cdot \log_2(w_{i_1}) - \sum_{\dots} w_{i_1} \cdot \dots \cdot \log_2(w_{i_2}) \\
 &\quad - \sum_{\dots} \dots - \sum_{\substack{i_1, i_2, \dots, i_n \\ 0 \leq i_k \leq q}} w_{i_1} \cdot w_{i_2} \cdot \dots \cdot w_{i_n} \log_2(w_{i_n}) \\
 &= - \sum_{i_1=1}^q w_{i_1} \cdot \log_2(w_{i_1}) \cdot \sum_{i_2=1}^q w_{i_2} \cdot \dots \cdot \sum_{i_n=1}^q w_{i_n} \cdot \\
 &\quad - \sum_{i_2=1}^q w_{i_2} \cdot \log_2(w_{i_2}) \cdot \dots - \sum_{i_n=1}^q w_{i_n} \log_2(w_{i_n}) \\
 &= n \cdot H(S)
 \end{aligned}$$

Die Quellenerweiterung ermöglicht es, bessere Quellencodierungen zu finden.

Betrachten wir noch einmal unser Beispiel,

Code für  $Q$ :

$$z_1 \longrightarrow 0; \quad z_2 \longrightarrow 1.$$

Code für  $Q^2$ :

$$\begin{aligned}
 z_1 z_1 &\longrightarrow 010; & z_1 z_2 &\longrightarrow 011; \\
 z_2 z_1 &\longrightarrow 00; & z_2 z_2 &\longrightarrow 0.
 \end{aligned}$$

Code für  $Q^3$ :

$$\begin{aligned}
 z_1 z_1 z_1 &\longrightarrow 11100; & z_1 z_1 z_2 &\longrightarrow 11101; \\
 z_1 z_2 z_1 &\longrightarrow 11110; & z_1 z_2 z_2 &\longrightarrow 100; \\
 z_2 z_1 z_1 &\longrightarrow 11111; & z_2 z_1 z_2 &\longrightarrow 101; \\
 z_2 z_2 z_1 &\longrightarrow 110; & z_2 z_2 z_2 &\longrightarrow 0.
 \end{aligned}$$

Der mittlere Bitaufwand für die Codierung eines Einzelzeichens reduziert sich von 1 über 0,84375 auf 0,82292.

**Verrauschter Kanal:**



**Bemerkungen:**

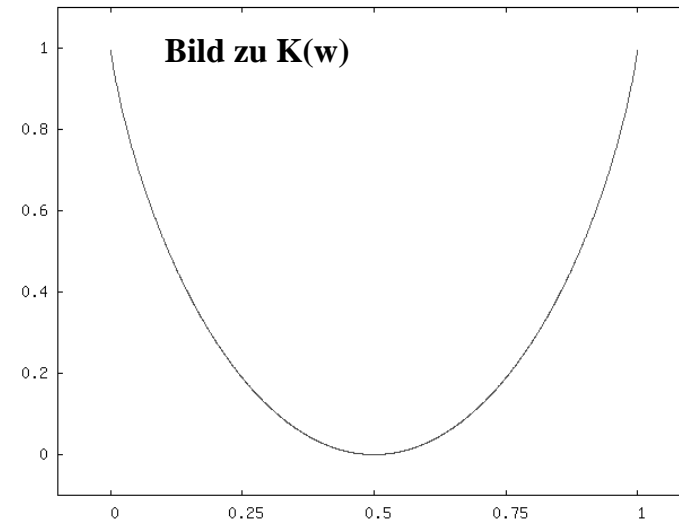
- (i) **Eingabe- und Ausgabezeichenvorrat müssen nicht identisch sein.**
- (ii) **Die Rauschquelle könnte man wie eine normale Informationsquelle charakterisieren. Ihre Wirkung beschreibt man jedoch relativ zum Eingabealphabet durch eine Matrix bedingter Wahrscheinlichkeiten.**

$$\left[ W(a_j | e_i) \right] \quad (i = 1, \dots, n; j = 1, \dots, m)$$

$W(a_j | e_i)$  ist die Wahrscheinlichkeit, daß die Eingabe des Zeichens  $e_i$  zur Beobachtung des Ausgabezeichens  $a_j$  führt.

- (iii) **Man nennt für einen binären Blockcode  $C$  der Länge  $n$  die Größe  $R = \log_2(M)/n$  die Übertragungsrate des Codes  $C$  vom Umfang  $M$ .**

- (iv) **Es ist die Aufgabe der Codierungstheorie zu klären, unter welchen Bedingungen man durch Einführung von Redundanz von den beobachteten Ausgaben auf die erfolgten Eingaben mit hoher Wahrscheinlichkeit rückschließen kann.**
- (v) **Für einen gestörten binären symmetrischen Kanal mit Störwahrscheinlichkeit  $w$  bezeichnet man mit  $K(w)$  seine Kapazität. Es gilt:**  
 $K(w) = 1 + w * \log_2(w) + (1-w)*\log_2(1-w)$



- (vi) **Satz von Shannon: Man habe einen gestörten binären symmetrischen Kanal mit Störwahrscheinlichkeit  $w$  und Kapazität  $K(w)$ . Falls die Übertragungsrate  $R$  kleiner als  $K(w)$  ist, findet man zu jedem  $\epsilon > 0$  einen Code  $C$  mit Übertragungsrate  $R(C)$  und  $K(w) \geq R(C) \geq R$  und Fehlerdecodierwahrscheinlichkeit  $< \epsilon$ .**

## 2.7 Fehlererkennende Codes

Beispiel eines Codes mit Prüfinformation:

Ein 2-aus-5-Code:

Code	Wert	Berechnung	
1 1 0 0 0	1	0 + 1	
1 0 1 0 0	2	0 + 2	
0 1 1 0 0	3	1 + 2	
1 0 0 1 0	4	0 + 4	
0 1 0 1 0	5	1 + 4	
0 0 1 1 0	6	2 + 4	
1 0 0 0 1	7	0 + 7	
0 1 0 0 1	8	1 + 7	
0 0 1 0 1	9	2 + 7	
0 0 0 1 1	0	4 + 7	!!Überlauf!!
<hr/>			
0 1 2 4 7	Stellenwertigkeit		

Die folgenden Bitketten sind sicherlich keine Elemente des 2-aus-5-Codes:

11001, 1001, 100001, 10101, 00001, 10111.

Zur Fehlerbehandlung:

Den Daten fügt man Prüfinformation hinzu, oft Prüfsumme genannt.

Daten	Prüfsumme
-------	-----------

Man kennt drei Nutzungsarten für die Prüfsumme:

- Fehlerentdeckung,
- Fehlerkorrektur,
- Korrektur einfacher Fehler, Entdeckung schwerer Fehler.

Beispiele zur Berechnung der Prüfinformation:

- Prüfziffer,
- Parität,
- Summenbildung,
- CRC-Verfahren (Cyclic Redundancy Check),
- Hamming-Codes,
- BCH-Codes (Bose, Ray-Chauduri, Hocquengham),
- Reed-Solomon-Codes.

**Bemerkung:** Die Schutzinformation muß nicht getrennt von den zu schützenden Daten verwaltet werden, sie kann mit den eigentlichen Daten eng verwoben sein. Ein Beispiel ist der 2-aus-5-Code.

### Parität:

Einer Bitkette wird ein Bit hinzugefügt, so daß die Anzahl der Einerbit gerade ist.

### Beispiel:

1010100 1  
0110011 0

Die Berechnung des Paritätsbits erfolgt mittels des Exklusiven Oders.

Bitkette:  $b_0 b_1 b_2 \dots b_n$

Paritätsbit =  $b_0 \text{ xor } b_1 \text{ xor } b_2 \text{ xor } \dots \text{ xor } b_n$

Beispiel: 101010

$pb = 1 \text{ xor } 0 \text{ xor } 1 \text{ xor } 0 \text{ xor } 1 \text{ xor } 0 = 1$

Prüfung einer geschützten Bitkette:

$$\text{XOR}_{i=0}^n b_i \text{ xor } pb = 0$$

### Bemerkungen:

- (i) Die obige Parität nennt man die gerade Parität; neben dieser existiert auch eine ungerade Parität, man gewinnt sie durch Invertierung der geraden Parität.
- (ii) Die Prüfmächtigkeit der einfachen Parität ist gering, gerade Anzahlen von Bitfehlern entdeckt sie nicht.

### Zweidimensionale Parität:

### Beispiel:

H	100 1000	0
A	100 0001	0
M	100 1101	0
M	100 1101	0
I	100 1001	1
N	100 1110	0
G	100 0111	0
I	100 1001	1

### Bemerkungen:

- (i) Die Fehlerentdeckungskapazität ist gut; nur wenige Fehlermuster (welche?) werden nicht entdeckt.
- (ii) Die zweidimensionale Parität kann auch zur Fehlerkorrektur eingesetzt werden.

### Beispiel zur Fehlerentdeckungskapazität:

Das obige Code-Beispiel umfaßt 64 Bit; es existieren daher 635.376 Vier-Bit-Fehler, von denen 784 nicht entdeckt werden; damit ist der Anteil der nichtentdeckbaren Vier-Bit-Fehler kleiner als 0,0013.

## Zweidimensionale Addition:

Eingabe:        3   7   4  
                   5   4   8  
                   1   3   5

Bildung von Spalten- und Zeilensummen modulo 10:

3	7	4	4
5	4	8	7
1	3	5	9
9	4	7	

Verfälschung eines Eintrags bei der Übertragung:

3	7	4	4
5	4	3	7
1	3	5	9
9	4	7	

Neuberechnung der Spalten- und Zeilensummen:

3	7	4	4	4
5	4	3	7	2 ← Fehler
1	3	5	9	9
9	4	7		
9	4	2		

↑ Fehler

Nun läßt sich der Fehler einfach korrigieren!

## 2.8 Fehlerkorrigierende Codes

Mit einem Paritätsbit kann man eine ungerade Zahl von Bitfehlern entdecken, um Bitfehler auch berichtigen zu können, benötigt man mehrere Paritätsgleichungen. Für die Korrektur von Bitfehlern muß bekannt sein, daß bei einer Übertragung von Bitketten die Anzahl der Bitfehler eine vorgegebene Schranke nicht überschreitet.

Formel:

$$\text{Parität } P = b_0 + b_1 + b_2 + \dots + b_{n-1} \pmod{2},$$

falls Bitkette  $B = b_0b_1b_2 \dots b_{n-1}$ .

Beispiel: Wir betrachten Bitketten der Länge 8.

Seien:

$$P_0 = b_0 + b_1 + b_2 + \dots + b_6 \pmod{2}$$

$$P_1 = b_0 + \dots + b_3 + b_4 + \dots + b_6 + b_7 \pmod{2}$$

$$P_2 = \dots + b_1 + \dots + b_3 + \dots + b_5 + b_6 + b_7 \pmod{2}$$

$$P_3 = \dots + b_2 + \dots + b_4 + b_5 + \dots + b_7 \pmod{2}$$

Weiß man, daß nur Ein-Bit-Fehler auftreten können, dann bestimmt man aus dem Vergleich von berechneten und übertragenen Paritäten, ob ein Bit verfälscht wurde. Wurde ein Bit verfälscht, dann liefert der Vergleich auch das fehlerhafte Bit.

### Numerisches Beispiel:

$$B = 10011101$$

$$P_0 = 1 + 0 + 0 + 0 = 1 \pmod{2}$$

$$P_1 = 1 + 1 + 1 + 0 + 1 = 0 \pmod{2}$$

$$P_2 = 0 + 1 + 1 + 0 + 1 = 1 \pmod{2}$$

$$P_3 = 0 + 1 + 1 + 1 = 1 \pmod{2}$$

gesendete Bitkette: GK = 10011101 1011

empfangene Bitkette: EK = 10011101 1111

Neuberechnung der Paritätsbits ergibt:

$$P_0 = 1, P_1 = 0, P_2 = 1, P_3 = 1.$$

Man stellt nur eine Abweichung bezüglich des Paritätsbits  $P_1$  fest, die Datenbits sind unverfälscht.

Eine weitere fehlerhafte Übertragung:

gesendete Bitkette: GK = 10011101 1011

empfangene Bitkette: EK = 10010101 1011

Neuberechnung der Paritätsbits ergibt:

$$P_0 = 1, P_1 = 1, P_2 = 1, P_3 = 0.$$

Man stellt nur eine Abweichung bezüglich der Paritätsbits  $P_1$  und  $P_3$  fest, das Datenbit  $b_4$  ist falsch. Die korrigierte empfangene Bitkette ist:  
10011101.

### Ein Code von Hamming zur Korrektur von Ein-Bit-Fehlern:

Algorithmus:

Annahme: Sei  $n$  die Bitbreite des Ausgangscodes.

1. Bestimme kleinstes  $k$  mit  $n \leq 2^k - k - 1$ .
2. Die Bitpositionen  $2^0, 2^1, \dots, 2^{k-1}$  nehmen die Prüfbits auf.
3. Die übrigen Positionen nehmen die Originalbits ein.
4. Berechne Prüfbit  $i$  als Modulo-2-Summe aller Bits, deren Positionsnummer ein gesetztes  $i$ -Bit enthält.
5. Bei der Überprüfung werden die Kontrollbits in die Summenbildung einbezogen.

Beispiel:

Originalvektor:  $(a_1, a_2, a_3, a_4)$

$n = 4$ , damit  $k = 3$ , denn  $4 \leq 2^3 - 3 - 1$

Positionsnummern:

$$001 \quad c_1 = a_1 + a_2 + a_4$$

$$010 \quad c_2 = a_1 + a_3 + a_4$$

$$011 \quad a_1$$

$$100 \quad c_3 = a_2 + a_3 + a_4$$

$$101 \quad a_2$$

$$110 \quad a_3$$

$$111 \quad a_4$$

Sei  $(b_{001}, b_{010}, b_{011}, b_{100}, b_{101}, b_{110}, b_{111})$  der empfangene Vektor.

Berechnung der Fehlerposition durch:

$$f_0 = b_{001} + b_{011} + b_{101} + b_{111}$$

$$f_1 = b_{010} + b_{011} + b_{110} + b_{111}$$

$$f_2 = b_{100} + b_{101} + b_{110} + b_{111}$$

$$\text{Fehlerposition} = f_0 + 2 \cdot f_1 + 2^2 \cdot f_2$$

**Bemerkung:** Ist die Fehlerposition gleich 0, dann liegt kein Fehler vor.

Ein numerisches Beispiel:

Originalvektor:  $(1, 0, 0, 1)$

$$c_1 = 1 + 0 + 1 = 0$$

$$c_2 = 1 + 0 + 1 = 0$$

$$c_3 = 0 + 0 + 1 = 1$$

Geschützter Vektor:  $(0, 0, 1, 1, 0, 0, 1)$

Empfangener Vektor:  $(0, 0, 1, 1, 1, 0, 1)$

Berechnung der Fehlerposition:

$$f_0 = 0 + 1 + 1 + 1 = 1$$

$$f_1 = 0 + 1 + 0 + 1 = 0$$

$$f_2 = 1 + 1 + 0 + 1 = 1$$

Fehlerposition = 5

Korrigierter Vektor:  $(0, 0, 1, 1, 0, 0, 1)$

**Hamming-Abstand:**

Um Bitketten gegen Übertragungsfehler zu schützen, fügt man ihnen weitere Schutzbit hinzu. Als einfaches Beispiel betrachten wir die Codierung:

$$C: \{0, 1\}^2 \rightarrow \{0, 1\}^5$$

gegeben durch:

$$00 \rightarrow 00111$$

$$01 \rightarrow 01100$$

$$10 \rightarrow 10010$$

$$11 \rightarrow 11001$$

Es läßt sich nun die Wahrscheinlichkeit  $P$  berechnen, daß eine zufällige Bitkette eine korrekte ist:

$$P = \frac{\text{korrekte Fälle}}{\text{alle Fälle}} = \frac{4}{32} = \frac{1}{8} = 0,125$$

Im obigen Fall läßt sich die Schwere eines Fehlers bestimmen. Zwischen je zwei Bitketten führt man einen Abstand ein.

**Definition:** Zwei Bitketten  $a$  und  $b$  gleicher Länge haben den Hamming-Abstand  $d(a, b) = n$ , falls sie sich in genau  $n$  Positionen unterscheiden.

**Beispiel:**  $x = 10111\ 0110$   
 $y = 00101\ 0101$   
 $d(x, y) = 4$

**Bemerkungen:**

- (i) Der Hamming-Abstand ist eine Metrik auf  $\{0, 1\}^n$ .
- (ii) Bei der Decodierung einer Nicht-Code-Bitkette  $k$  wählt man als wahrscheinliches Codeelement eine Bitkette kleinsten Hamming-Abstandes zu  $k$ .
- (iii) Als Hamming-Abstand eines Codes  $C$  bezeichnet man:  $\min \{d(x, y) \mid x, y \in C, x \neq y\}$ . Man sollte Codes möglichst großen Hamming-Abstandes verwenden.
- (iv) Als Hamming-Gewicht  $w(k)$  einer Bitkette  $k$  bezeichnet die Anzahl der Positionen ungleich 0 in  $k$ ; es gilt:  
 $w(b_1, b_2, \dots, b_n) = d(b_1, b_2, \dots, b_n, (0, \dots, 0))$ .
- (v) Die Begriffe Hamming-Abstand und Hamming-Gewicht lassen sich auf Vektoren über beliebigen Körpern übertragen.
- (vi) Beträgt der Hamming-Abstand zweier Codewörter mindestens  $h$ , dann lassen sich  $h-1$  Bitfehler erkennen oder  $\lfloor (h-1) / 2 \rfloor$  Bitfehler korrigieren.

In einer Decodiertabelle ordnet man jede Bitkette der Länge 5 einem Element der Codierung  $C$  zu.

Codeelement	00111	01100	10010	11001
A1 Abstand 1	00110	01101	10011	11000
	00101	01110	10000	11011
	00011	01000	10110	11101
	01111	00100	11010	10001
	10111	11100	00010	01001
A2 Abstand 2	00001	10100	01010	11111
	10101	00000	11110	01011

**Bemerkungen:**

- (i) Bei Bitkettenübertragungen ordnet man jeder empfangenen Bitkette die Bitkette in der ersten Zeile der gleichen Spalte zu.
- (ii) Die Bitketten der Gruppe A1 unterscheiden sich nur in einem Bit von der gleichspaltigen Bitkette in der ersten Zeile, somit können Ein-Bit-Fehler korrigiert werden.
- (iii) Die Bitketten der Gruppe A2 unterscheiden sich in zwei Bit vom Spaltenführer. Benutzt man die Tabelle zur Korrektur, dann führt man auch Fehlkorrekturen durch, denn die Bitkette 00000 unterscheidet sich in zwei Bit von den Codeelementen 01100 und 10010. Ändert sich bei der Übertragung die Bitkette 01100 in die Bitkette 01111, dann erfolgt eine Fehlzuordnung.
- (iv) Sei  $p$  die Wahrscheinlichkeit für die unabhängige Änderung eines Bit, dann berechnet sich die Wahrscheinlichkeit für eine korrekte Decodierung gemäß obiger Tabelle zu:

$$(1 - p)^5 + 5*(1 - p)^4*p + 2*(1 - p)^3*p^2.$$

## 2.9 Zyklische Codes

Zur Vorbereitung: Rechnen mit Binärpolynomen:

Jeder Bitkette läßt sich ein Polynom über dem Körper  $\{0, 1\}$  zuordnen.

Beispiel: Bitkette = 1001101

Einige Möglichkeiten der Zuordnung:

- (a)  $1 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0 =$   
 $x^6 + x^3 + x^2 + x^0$
- (b)  $x^0 + x^{-3} + x^{-4} + x^{-6}$
- (c)  $x^0 + x^3 + x^4 + x^6$
- (d)  $x^0 + x^{-2} + x^{-3} + x^{-6}$

Ab jetzt betrachten wir nur die Zuordnung (a).

Addition von Polynomen:

$$(x^6 + x^3 + x^2 + x^0) + (x^6 + x^5 + x^2 + x^1) = x^5 + x^3 + x^1 + 1$$

Subtraktion von Polynomen:

$$(x^6 + x^3 + x^2 + x^0) - (x^6 + x^5 + x^2 + x^1) = x^5 + x^3 + x^1 + 1$$

Multiplikation von Polynomen:

$$(x^6 + x^3 + x^2 + x^0) * (x^6 + x^5 + x^2 + x^1) =$$

$$x^{12} + x^9 + x^8 + x^6 + x^{11} + x^8 + x^7 + x^5 + x^8 + x^5 + x^4 + x^2 + x^7 + x^4 + x^3 + x^1 =$$

$$x^{12} + x^{11} + x^9 + x^8 + x^6 + x^3 + x^2 + x^1$$

Zweite Art der Multiplikation von Polynomen:

$$(x^6 + x^3 + x^2 + x^0) * (x^6 + x^5 + x^2 + x^1)$$


---


$$\begin{array}{r} 1001101 \\ 1001101 \\ 1001101 \\ 1001101 \\ \hline 110110100111 \end{array}$$

$$= x^{12} + x^{11} + x^9 + x^8 + x^6 + x^3 + x^2 + x^1$$

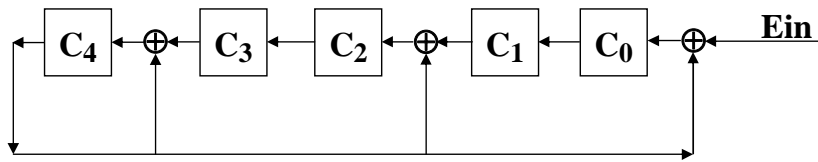
Division von Polynomen:

$$\frac{x^9 + x^8 + x^6 + x^4 + x^3 + x^1 + x^0}{x^9 + x^6 + x^5} : \frac{x^4 + x^1 + x^0}{x^4 + x^1 + x^0} = \frac{x^5 + x^4}{x^4 + x^1 + x^0}$$

$$\begin{array}{r} 8 \ 5 \ 4 \\ x + x + x \\ 8 \ 5 \ 4 \\ x + x + x \\ \hline x^3 + x^1 + x^0 \end{array}$$

$$\text{Rest} = x^3 + x^1 + x^0$$

**Division mittels Schieberegister:**



**Divisor:**  $x^5 + x^4 + x^2 + 1$

**Rechnung zum Dividenden:**  $x^{14} + x^{12} + x^8 + x^7 + x^5$

	C4	C3	C2	C1	C0	Eingabe
<b>Start:</b>	0	0	0	0	0	1
<b>Schritt 1:</b>	0	0	0	0	1	0
<b>Schritt 2:</b>	0	0	0	1	0	1
<b>Schritt 3:</b>	0	0	1	0	1	0
<b>Schritt 4:</b>	0	1	0	1	0	0
<b>Schritt 5:</b>	1	0	1	0	0	0
<b>Schritt 6:</b>	1	1	1	0	1	1
<b>Schritt 7:</b>	0	1	1	1	0	1
<b>Schritt 8:</b>	1	1	1	0	1	0
<b>Schritt 9:</b>	0	1	1	1	1	1
<b>Schritt 10:</b>	1	1	1	1	1	0
<b>Schritt 11:</b>	0	1	0	1	1	0
<b>Schritt 12:</b>	1	0	1	1	0	0
<b>Schritt 13:</b>	1	1	0	0	1	0
<b>Schritt 14:</b>	0	0	1	1	1	0
<b>Schritt 15:</b>	0	1	1	1	0	

**Divisionsrest:**  $x^3 + x^2 + x^1$

**Definition:** Sei  $z = z_1z_2 \dots z_n$  eine Zeichenkette der Länge  $n$ . Man sagt die Zeichenkette  $y = z_nz_1 \dots z_{n-1}$  geht aus der Zeichenkette  $z$  durch eine zyklische Rechtsverschiebung hervor,  $y$  nennt man die Rechtsverschobene von  $z$ , in Zeichen  $y = RV(z)$ .

**Definition:** Einen Code  $C$  nennt man zyklisch, falls mit  $e \in C$  auch  $RV(e) \in C$ .

**Bemerkung:** Im folgenden betrachten wir nur Blockcodes über  $F = \{0, 1\}$ . Einen Blockcode  $C$  über  $\{0, 1\}$  nennt man linear, falls mit  $a \in C$  und  $b \in C$  auch  $a+b \in C$ . Wir verlangen von einem zyklischen Code zusätzlich, daß er auch linear ist.

**Beispiel:**

Sei  $101$  ein Element eines zyklischen Codes  $C$ . Dann enthält  $C$  mindestens  $000, 110 = RV(101), 011 = RV(110)$  und  $101 = RV(011)$ .

$C = \{000, 101, 110, 011\}$  ist ein vollständiger (linearer) zyklischer Code.

**Nachweis:**  $101 + 101 = 000, 101 + 110 = 011,$   
 $101 + 011 = 110, 110 + 011 = 101.$

**Durch Addition erhält man keine weiteren Elemente.**

**Weiteres Beispiel:**

Der Code  $CA = \{0000, 1001, 0110, 1111\}$  ist sicher kein zyklischer Code. Durch Vertauschen der Zeichen auf der dritten und vierten Position erhält man einen zyklischen Code CB, der weitgehend gleichwertig zu CA ist.  $CB = \{0000, 1010, 0101, 1111\}$  ist zyklisch.

**Bildung eines zyklischen Codes durch Division mit einem Polynom:** hier Divisor =  $x^3 + x + 1$

Zahl	Bitkette	Code
0	0000	0000 000
1	0001	0001 011
2	0010	0010 110
3	0011	0011 101
4	0100	0100 111
5	0101	0101 100
6	0110	0110 001
7	0111	0111 010
8	1000	1000 101
9	1001	1001 110
10	1010	1010 011
11	1011	1011 000
12	1100	1100 010
13	1101	1101 001
14	1110	1110 100
15	1111	1111 111

**Bemerkungen:**

(i) **Rechnung zum Fall 9:**  
 $1001000 : 1011 = 1010$   

$$\begin{array}{r} 1011 \\ 1000 \\ \hline 1011 \\ 110 \end{array}$$
 damit Code = 1001110

(ii) Um einen zyklischen Code in  $\{0, 1\}^n$  zu erhalten, muß gelten: das erzeugende Divisionspolynom muß ein Faktor von  $x^n - 1$  sein.

(iii) **Beispiele einiger Faktorisierungen:**  
 $x^7 + 1 = (x+1) * (x^3 + x^2 + 1) * (x^3 + x + 1)$   
 $x^8 + 1 = (x+1)^8$   
 $x^{10} + 1 = (x+1)^2 * (x^4 + x^3 + x^2 + x + 1)^2$   
 $x^{15} + 1 = (x+1) * (x^2 + x + 1) * (x^4 + x + 1) * (x^4 + x^3 + 1) * (x^4 + x^3 + x^2 + x + 1)$

(iv) Mehrere Bitfehler lassen sich mit BCH-Codes korrigieren, zwei Beispiele seien:  
 $BCH(15, 5, 3) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$   
 $BCH(15, 7, 2) = x^8 + x^7 + x^6 + x^4 + 1$   
 Hierbei tragen die Zahlen die folgende Bedeutung: die erste Zahl beschreibt die Codelänge, die zweite Zahl die Zahl der Informationsbit und die dritte Zahl die Korrekturkapazität, die Polynome sind die Generatorpolynome.

**Einsatz der Polynomdivision zum Nachrichtenschutz:**

**Bemerkung:** Losgelöst von der Theorie der zyklischen Codes setzt man die Polynomdivision zur Entdeckung von Übertragungsfehlern ein. Man spricht dann von einem CRC-Verfahren. (CRC = Cyclic Redundancy Check)

Sei  $P(x)$  ein Polynom mit Term 1 vom Grad  $k > 0$ ,  
sei  $Q(x)$  eine zu übermittelnde Nachricht.

Man berechnet

$$r(x) = \text{Rest}\left(\frac{x^k * Q(x)}{P(x)}\right)$$

und übermittelt  $N(x) = x^k * Q(x) + r(x)$ .

Am Empfangsort prüft man:

$$\text{Rest}\left(\frac{N(x)}{P(x)}\right) = \text{Rest}\left(\frac{x^k * Q(x) + r(x)}{P(x)}\right) = 0$$

**Bemerkungen:**

- (i) Die geschützte Nachricht ist ein Vielfaches des Schutzpolynoms.
- (ii) Der Polynomschutz stellt einen Schutz gegen Bündelfehler bis zur Länge  $k$  dar.
- (iii) Um mindestens den Paritätsschutz zu gewährleisten, wählt man  $x+1$  als Faktor des Divisionspolynoms.

**Beispiel zum CRC-Schutz:**

Nachricht: 1010001101  
Divisionspolynom: 110101

**Durchführung der Division:**

$$\begin{array}{r}
 x^{14} + x^{12} + x^8 + x^7 + x^5 : x^5 + x^4 + x^2 + x^0 = x^9 + x^8 + x^6 + x^4 + x^2 + x^1 \\
 \underline{x^{14} + x^{13} + x^{11} + x^9} \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} x^{13} + x^{12} + x^{11} + x^9 + x^8 \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \underline{x^{13} + x^{12} + x^{10} + x^8} \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \phantom{x^{13} + x^{12} + x^{10} + x^8} x^{11} + x^{10} + x^9 + x^7 \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \phantom{x^{13} + x^{12} + x^{10} + x^8} \phantom{x^{11} + x^{10} + x^9 + x^7} \underline{x^{11} + x^{10} + x^8 + x^6} \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \phantom{x^{13} + x^{12} + x^{10} + x^8} \phantom{x^{11} + x^{10} + x^9 + x^7} \phantom{x^{11} + x^{10} + x^8 + x^6} x^9 + x^8 + x^7 + x^6 + x^5 \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \phantom{x^{13} + x^{12} + x^{10} + x^8} \phantom{x^{11} + x^{10} + x^9 + x^7} \phantom{x^{11} + x^{10} + x^8 + x^6} \underline{x^9 + x^8 + x^6 + x^4} \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \phantom{x^{13} + x^{12} + x^{10} + x^8} \phantom{x^{11} + x^{10} + x^9 + x^7} \phantom{x^{11} + x^{10} + x^8 + x^6} \phantom{x^9 + x^8 + x^7 + x^6 + x^5} x^7 + x^5 + x^4 \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \phantom{x^{13} + x^{12} + x^{10} + x^8} \phantom{x^{11} + x^{10} + x^9 + x^7} \phantom{x^{11} + x^{10} + x^8 + x^6} \phantom{x^9 + x^8 + x^7 + x^6 + x^5} \underline{x^7 + x^6 + x^4 + x^2} \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \phantom{x^{13} + x^{12} + x^{10} + x^8} \phantom{x^{11} + x^{10} + x^9 + x^7} \phantom{x^{11} + x^{10} + x^8 + x^6} \phantom{x^9 + x^8 + x^7 + x^6 + x^5} \phantom{x^7 + x^6 + x^5 + x^4} x^6 + x^5 + x^2 \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \phantom{x^{13} + x^{12} + x^{10} + x^8} \phantom{x^{11} + x^{10} + x^9 + x^7} \phantom{x^{11} + x^{10} + x^8 + x^6} \phantom{x^9 + x^8 + x^7 + x^6 + x^5} \phantom{x^7 + x^6 + x^5 + x^4} \underline{x^6 + x^5 + x^3 + x^1} \\
 \phantom{x^{14} + x^{13} + x^{11} + x^9} \phantom{x^{13} + x^{12} + x^{10} + x^8} \phantom{x^{11} + x^{10} + x^9 + x^7} \phantom{x^{11} + x^{10} + x^8 + x^6} \phantom{x^9 + x^8 + x^7 + x^6 + x^5} \phantom{x^7 + x^6 + x^5 + x^4} \phantom{x^6 + x^5 + x^4 + x^2} x^3 + x^2 + x^1
 \end{array}$$

Rest =  $x^3 + x^2 + x^1$

damit CRC-Schutzcode = 01110

damit geschützte Nachricht = 101000110101110

## Schutzkapazität von Polynomen:

Sei  $g(x)$  das Schutzpolynom, sei  $v(x) = a(x) * g(x)$  die geschützte Nachricht, sei  $f(x)$  die verfälschte Nachricht, sei  $e(x) = f(x) - v(x)$  das Fehlerpolynom;  $v(x)$  sei vom Grade  $n$ .

**Ein-Bit-Fehler:**  $e(x) = x^i$  :

Diese Fehler werden entdeckt, falls  $g(x)$  mindestens zwei Terme besitzt. Die Division des eintermigen  $e(x)$  durch ein mehrtermiges Polynom liefert immer einen Rest.

Sei nun  $x+1$  ein Faktor von  $g(x)$ , dann gilt:  $x+1$  ist auch ein Faktor von  $v(x)$ .  $v(1) = b(1) * (1+1) = 0$ , dies bedeutet,  $v$  hat gerade Parität und jede ungerade Zahl von Fehlern wird entdeckt.

**Zwei-Bit-Fehler:**

$$e(x) = x^i + x^j = x^i * (1 + x^{j-i}) \text{ für } 0 \leq i < j \leq n$$

Alle Zwei-Bit-Fehler werden entdeckt, falls  $x$  nicht Faktor von  $g(x)$  und  $g(x)$  nicht  $1 + x^h$  für  $1 \leq h \leq n$  teilt.

**Bündelfehler:**

$$e(x) = x^i * (1 + e_1 * x + \dots + e_{k-1} * x^{k-1})$$

Nur falls  $e(x) * x^{-i}$  ein Vielfaches von  $g(x)$  ist, wird der Fehler nicht entdeckt.

$$\text{CRC-16: } x^{16} + x^{15} + x^2 + 1 = (x+1) * (x^{15} + x + 1)$$

Die kleinste Zahl  $m$ , für die CRC-16  $x^m + 1$  ohne Rest teilt, ist 32767.

## Schlußbemerkung:

Es gibt viele Bücher über Informations- und Codierungstheorie. Ein Buch, das mir gefällt, ist dieses:

Werner Heise und Pasquale Quattrocchi,:  
Informations- und Codierungstheorie, 3. Aufl.,  
Springer, 1995  
ISBN-10: 3-540-57477-8