Vorlesung WS 2014/2015
Ruhr-Universität Bochum

Algebraische Codierung für die sichere Datenübertragung

Dr.-Ing. Klaus Huber
Berlin

coding@klaus-huber.net

Die wichtigsten Bücher

E.Berlekamp: Algebraic Coding Theory,
Aegean Park Press, 1968/1984

Peterson, Weldon: Error-Correcting Codes,
MIT Press, second Edition 1972

MacWilliams, Sloane: The Theory of Error Correcting Codes, North Holland 1977

Blahut: Theory and Practice of Error Control Codes,
Addison Wesley 1984

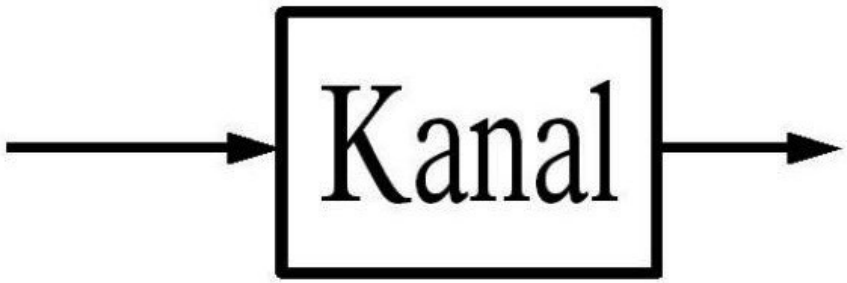
Pless, Huffman: Handbook of Coding Theory I, II,
North Holland 1998

McEliece: The Theory of Information and Coding,
Cambridge, 1977

Gliederung

1. Übersicht und Einführung
2. Grundlagen
   - Lineare, Nichtlineare Codes
   - Fehlererkennung und Korrektur
   - Generator- und Prüfmatrizen
   - Codeschranken
   - Hammingcodes
3. Die wichtigsten Codeklassen
   - BCH-, RS-, Goppacodes
4. Decodierverfahren für die Hammingmetrik
5. Codes für andere Metriken
   - Berlekamps Lee-Metrik Codes, Mannheim Metrik Codes
6. Das Kryptosystem von McEliece
7. Die MacWilliamstransformation

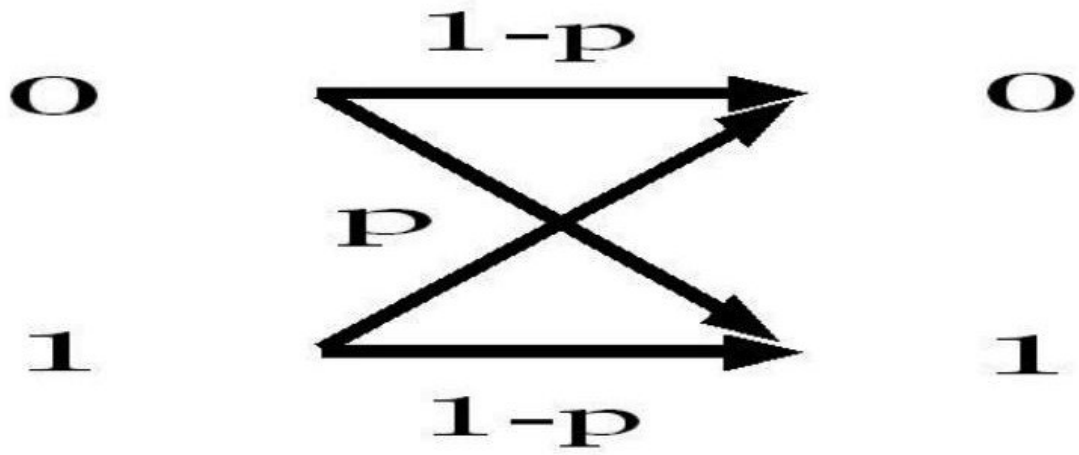
Übersicht und Einführung



Kanalkapazität

Charakteristischer Parameter eines Kanals. Die **Kanalkapazität** gibt an mit welcher Datenrate man prinzipiell fehlerfrei (d.h. mit Fehlerwahrscheinlichkeit Null) Daten übertragen kann.
Die Kanalkapazität ist eine Obergrenze für das prinzipiell erreichbare. Sie geht auf Claude Shannon zurück. Bei Datenraten oberhalb der Kanalkapazität ist eine fehlerfreie Datenübertragung nicht möglich.

Beispiel: Binärer symmetrischer Kanal (BSC)



$$C = 1 + p \operatorname{ld} p + (1-p) \operatorname{ld} (1-p)$$

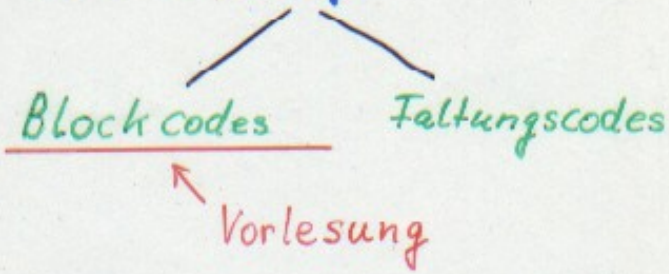
Beispiel: Bandbegrenzter Kanal mit Gaußschem weißen Rauschen

$$C = W\, ld\left(1+\frac{S}{N}\right)$$

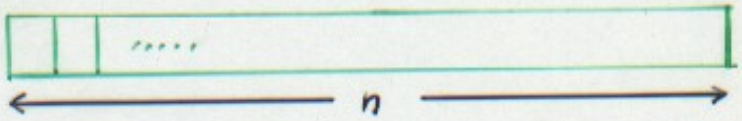
W: Bandbreite
S : Signalleistung
N: Rauschleistung

(AWGN-Kanal: Additive white Gaussian noise - channel)

Um in die Nähe der Kanalkapazität zu kommen benötigt man Codierung



Blockcodes



Blockcode: Menge von Codewörtern der Länge n über einem Alphabet.

Beispiel: Code über binärem Alphabet

$$\mathcal{C} = \{ (0,0,0), (1,1,1) \}$$

Konvention in Codierung:

Vektoren sind Zeilenvektoren

$\underline{c}_1 = (0,0,0)$ $\qquad \underline{c}_1^T = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

Bequemlichkeit: $\underline{c}_1 = 000$ (insbesondere bei Binärcodes)

d.h. $\mathcal{C} = \{000, 111\}$

charakteristische Werte eines Blockcodes:

Länge n im Beispiel $n = 3$

Anzahl der Codewörter M " " $M = 2$

Mindest (hamming) distanz d " " $d = 3$

Hamming Distanz zweier Vektoren:

Anzahl der Stellen an denen sich die Vektoren unterscheiden

z.B.

000	010	110	001
111	101	101	011
$d = 3$	$d = 3$	$d = 2$	$d = 1$

Mindestdistanz eines Codes: Kleinste Distanz zwischen zwei beliebigen verschiedenen Codewörtern des Codes

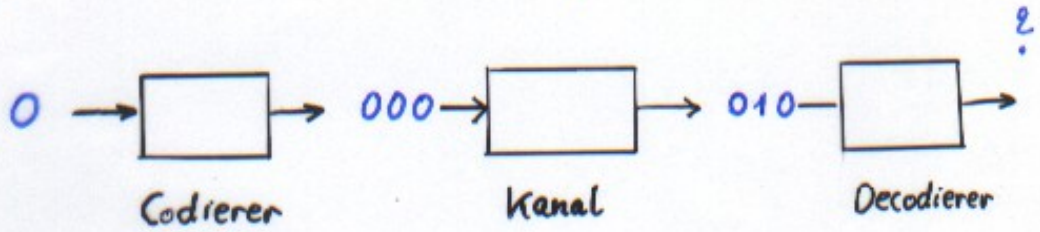
1. Wertetripel zur Charakterisierung

eines Blockcodes : (n, M, d)

Beispiel zur Codierung:

Um 0 zu senden: Übertrage 000

Um 1 zu senden: Übertrage 111



Der Decodierer findet zu einem empfangenen Vektor das "nächste" Codewort.

$\mathcal{C} = \{000, 111\}$

Distanz: 010 → 1, 010 → 2 ⇒ 000 → 0

000: nächstes Codewort

0: Informationsbit

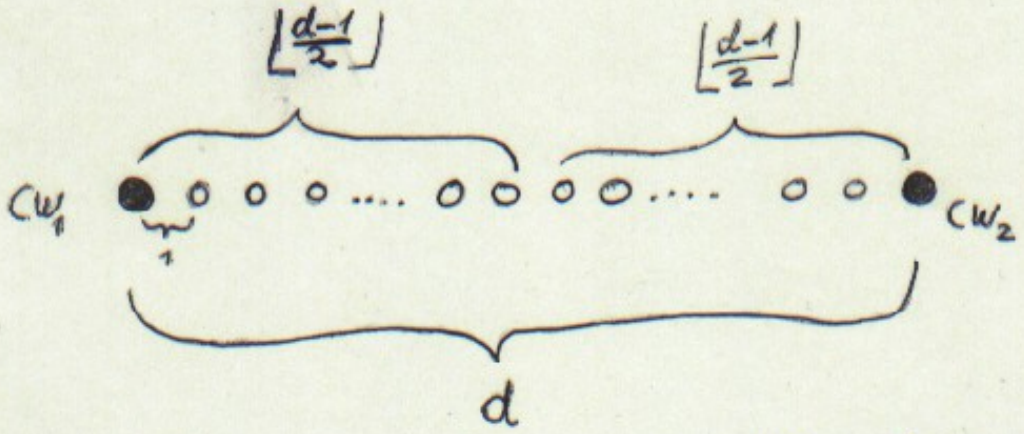
Ein Code mit Mindestdistanz d

kann bis zu $d-1$ Fehler erkennen.

Ein Code mit Mindestdistanz d kann

bis zu $\left\lfloor \frac{d-1}{2} \right\rfloor$ Fehler korrigieren

Hierzu: 2-dim. Veranschaulichung



hier: d ungerade

In Praxis besonders wichtig:

Lineare Codes

- Summe zweier Codewörter ergibt wieder ein Codewort (→ Alphabet muß Struktur haben) → Gruppe, Körper
- Komplexität von Codierung und Decodierung ist geringer als bei nichtlinearen Codes
- Bei einem linearen Code ist die

  Mindestdistanz = Mindestgewicht

  ↑ kleinste Distanz eines CW ≠ 0 zum Nullvektor
- Beschreibung des Codes mit linear unabh. Basisvektoren. Diese, in einer Matrix geeignet zusammengefaßt, bilden die Generatormatrix.

Beispiel:

$$\underline{G} = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right)$$

$\underline{c} = \underline{m} \cdot \underline{G}$ etwa $\underline{m} = 0010$

$\searrow \quad \underline{c} = 0010110$

Länge des Codes: $n = 7$

Anzahl der Codewörter: $M = 2^4 = 16$

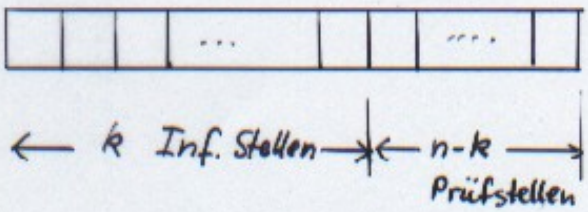
Mindestdistanz: $d = ?$ → H.A.

allgemeiner Block Code: (n, M, d)

linearer Code: $[n, k, d]$ ← 2. Wertetripel

in Beispiel

$k = ld\, M = 4$



→ systematische Codierung. Die Information kann direkt an k Stellen abgelesen werden.

Hauptproblem der Codierung:

Bei gegebenem n und k → Code finden, für den d so groß wie möglich ist sowie effiziente Codier- und Decodierverfahren

(bzw. n, d → maximales k

k, d → kleinstes n)

Fakt: Die Beschränkung auf lineare Codes läßt genügend sehr gute Codes "übrig".

(Es gibt Beispiele von nichtlinearen Binärcodes (z. B. Preparata Codes), die besser als alle linearen sind.)

Generatormatrix

$$\underline{G} = (\underline{I}_k \mid \underline{P})$$

$\underline{I}_k$: $k \times k$ Einheits-Matrix

$\underline{P}$: $k \times (n-k)$ Matrix

weitere sehr wichtige Matrix: **Prüfmatrix $\underline{H}$**

$$\underline{H} = (-\underline{P}^T \mid \underline{I}_{n-k})$$

$$\underline{G} \cdot \underline{H}^T = -\underline{P} + \underline{P} = \underline{O}$$

$$\Rightarrow \quad \underline{m} \cdot \underline{G} \cdot \underline{H}^T = \underline{c} \cdot \underline{H}^T = 0$$

$$\underline{H} \cdot \underline{c}^T = 0$$

Mit der $\underline{H}$-Matrix läßt sich leicht testen ob ein Vektor dem Code angehört.

Beispiel:

$$G = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right) = \left(\underline{\underline{I}}_4 \mid \underline{\underline{P}}\right)$$

$$H = \left(\begin{array}{ccccc|cc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right) = \left(-\underline{\underline{P}}^T \mid \underline{\underline{I}}_3\right)$$

$\underline{r} = \underline{c} + \underline{e}$ ← Fehler

$$\underline{\underline{H}} \cdot \underline{r}^T = \underline{\underline{H}} \cdot (\underline{c}^T + \underline{e}^T) = \underbrace{\underline{\underline{H}} \cdot \underline{c}^T}_{=0} + \underline{\underline{H}} \underline{e}^T$$

$$\underline{s}^T = \underline{\underline{H}} \cdot \underline{r}^T = \underline{\underline{H}} \cdot \underline{e}^T$$

$\underline{s}$ wird Syndrom genannt.

Bewußte Analogie zum medizinischen Ausdruck.

H.A.: Geben Sie Generator- und Prüfmatrix des binären $[n,1,n]$ Wiederholungscodes an (z.B. $[3,1,3] \rightarrow \{000, 111\}$).

H.A.: Geben Sie Generator- und Prüfmatrix des binären $[n,n-1,2]$ Codes an, der gebildet wird, wenn man jeden Nachrichtenvektor ($\underline{m}$) der Länge $n-1$ um ein Prüfbit ergänzt, sodaß das Gewicht des Codeworts gerade ist.

Modifikationen an $\underline{G}$

$$\underline{G} = (\underline{I}_k \mid \underline{P}) \longrightarrow [n, k, d]\text{-Code}$$

- eine Zeile von $\underline{G}$ streichen.

  die dann erhaltene 0-Spalte ebenfalls streichen.

$$\Rightarrow \quad \underline{\tilde{G}} = (\underline{I}_{k-1} \mid \underline{\tilde{P}}) \longrightarrow [n-1, k-1, d]\text{-Code}$$

- eine Spalte von $\underline{P}$ streichen

$$\underline{\hat{G}} = (\underline{I}_k \mid \underline{\hat{P}}) \longrightarrow [n-1, k, d-1]\text{-Code}$$

2 Codeschranken

Singleton Schranke:

$$d \leq n - k + 1$$

Beweis: Gewicht 1 im Informationsteil
+ Gewicht von maximal $n-k$ im Prüfteil eines Codewortes ($\neq \underline{0}$).

Codes, die diese Schranke mit Gleichheit erfüllen nennt man MDS-Codes
(MDS: Maximal Distance Separable, zu ihnen gehören die demnächst behandelten Reed-Solomon Codes.)

Hinweis: bei (nichttrivialen) Binärcodes ist Gleichheit in der Schranke nicht möglich.

Kugelpackungsschranke (auch Hammingschranke):

(Sphere packing)

$$M \cdot V_t \leq q^n$$

M: Anzahl der Codewörter des Codes

q: Alphabetgröße (binäre Codes: $q=2$)

n: Codelänge

t: $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ Korrekturfähigkeit

d: Mindestdistanz des Codes

V_t: "Kugel" mit Hammingradius t um Codewort, die das Codewort und alle Vektoren im Abstand $\leq t$ um das Codewort enthält.

Hausaufgabe: Bestimmen Sie V_t für Binärcodes

Codes, bei denen in obiger Gleichung **das** Gleichheitszeichen gilt, **nennt man perfekte** Codes.

binäre Hamming-Codes

Alle von Null verschiedenen binären m-Tupel werden als Spalten der Prüfmatrix geschrieben

z.B. $m = 3$

$$\underline{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

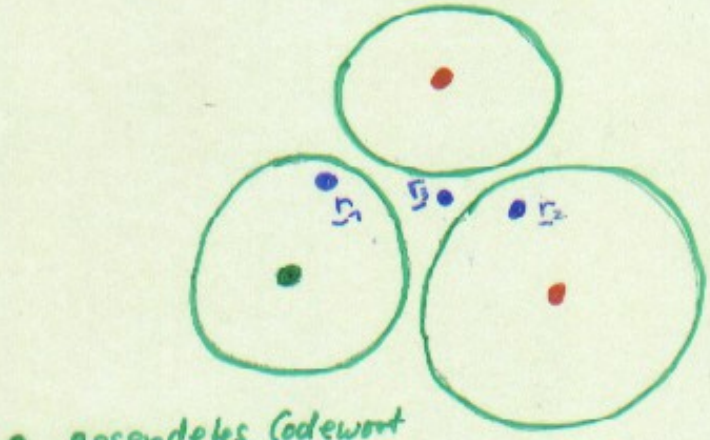
$$\Rightarrow \quad n = 2^m - 1$$

$$k = n - m = 2^m - 1 - m$$

$$d = 3$$

Die Mindestdistanz ist gleich 3, da jeder Fehler mit Gewicht 1 ein verschiedenes Syndrom erzeugt mit dem der Fehler lokalisiert werden kann.

Möglichkeiten beim Decodieren



1. Möglichkeit: empfangener Vektor liegt in Korrekturkugel des gesendeten Codeworts (z.B. r_1)

$\Rightarrow$ richtige Decodierung

2. Möglichkeit: empfangener Vektor liegt in Korrekturkugel eines anderen Codeworts

$\Rightarrow$ falsche Decodierung

3. Möglichkeit: empfangener Vektor liegt zwischen Korrekturkugeln

$\Rightarrow$ Decodier versagen.

Decodierverfahren, die immer das (oder die) nächsten Codewörter finden nennt man vollständige Decodierverfahren.

Der Euklidsche Algorithmus

Der EA ist ein effizientes Verfahren zur Bestimmung des grössten gemeinsamen Teilers zweier Zahlen m und n.

Sei $r_{-2} = m$ und $r_{-1} = n$, dann gilt:

$$r_{-2} = a_0 \cdot r_{-1} + r_0 \qquad a_0 = \left\lfloor \frac{r_{-2}}{r_{-1}} \right\rfloor \;,\; r_0 < r_{-1}$$

$$r_{-1} = a_1 \cdot r_0 + r_1$$

$\lfloor \cdot \rfloor$ Rundungsklammer

$$\vdots$$

$$r_{k-2} = a_k \cdot r_{k-1} + r_k \qquad a_k = \left\lfloor \frac{r_{k-2}}{r_{k-1}} \right\rfloor \;,\; r_k < r_{k-1}$$

$$\vdots$$

$$r_{t-2} = a_t \cdot r_{t-1} + r_t$$

$$r_{t-1} = a_{t+1} \cdot r_t$$

r_t ↑ $ggT(m, n)$

Beispiel: $m = 77$ $n = 49$

$77 = 1 \cdot 49 + 28$

$49 = 1 \cdot 28 + 21$

$28 = 1 \cdot 21 + 7$

$21 = 3 \cdot 7$

↑ $ggT(77, 49)$

Warum liefert dieser Algorithmus den ggT?

- Die r_i bilden eine monoton fallende Reihe, die mit einem letzten $r_t \neq 0$ endet.
- r_t teilt r_{t-1} $(r_t \mid r_{t-1})$

  $\Rightarrow r_t \mid r_{t-2}$ etc. $\Rightarrow r_t \mid n$ und $r_t \mid m$
- Jeder gemeinsame Teiler von r_{-2} und r_{-1}, also auch der grösste teilt $r_{-2}, r_{-1}, r_0, r_1 \ldots$ also auch r_t $\Rightarrow$ r_t ist der grösste gem. Teiler

Der erweiterte Euklidsche Algorithmus

$$r_{-2} = 1 \cdot m + 0 \cdot n$$

$$r_{-1} = 0 \cdot m + 1 \cdot n$$

$$r_{k-2} = q_{k-2} \cdot m + p_{k-2} \cdot n$$

$$r_{k-1} = q_{k-1} \cdot m + p_{k-1} \cdot n$$

$$r_k = q_k \cdot m + p_k \cdot n$$

$r_{k-2} = a_k \cdot r_{k-1} + r_k$

$a_k = \left\lfloor \frac{r_{k-2}}{r_{k-1}} \right\rfloor$

$$\downarrow \quad r_k = r_{k-2} - a_k \cdot r_{k-1} = q_k \cdot m + p_k \cdot n$$

$$\Rightarrow \quad q_{k-2} \cdot m + p_{k-2} \cdot n - a_k (q_{k-1} \cdot m + p_{k-1} \cdot n) = q_k \cdot m + p_k \cdot n$$

$$\Rightarrow \quad \boxed{\begin{aligned} q_k &= q_{k-2} - a_k \cdot q_{k-1} \\ p_k &= p_{k-2} - a_k \cdot p_{k-1} \\ r_k &= r_{k-2} - a_k \cdot r_{k-1} \end{aligned}}$$

3x die gleiche Rekursionsgleichung

mit $a_k = \left\lfloor \frac{r_{k-2}}{r_{k-1}} \right\rfloor$

$$\downarrow \quad \boxed{ggT(m,n) = q_t \cdot m + p_t \cdot n}$$

$R = \frac{k}{n}$

Beispiel: $ggT(35, 21)$

$\left\lfloor \frac{35}{21} \right\rfloor = 1$

$35 - 1 \cdot 21$

$35 = 1 \cdot 35 + 0 \cdot 21$

$21 = 0 \cdot 35 + 1 \cdot 21$

$1 - 1 \cdot 0$

$0 - 1 \cdot 1$

$14 = 1 \cdot 35 - 1 \cdot 21$

$7 = -1 \cdot 35 + 2 \cdot 21$

Binäre 1-Fehlerkorrigierende Hamming Codes

$$H = \begin{pmatrix} 0 & 0 & 0 & & 1 \\ 0 & 0 & 0 & & 1 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & 0 & & 1 \\ 0 & 1 & 1 & & 1 \\ 1 & 0 & 1 & \cdots & 1 \end{pmatrix}$$

enthält alle binären m-Tupel $\neq 0$

$\Rightarrow [n, k, 3]$ Code mit $n = 2^m - 1$, $k = n - m$

Jeder Einzelbitfehler führt zu einem eindeutigen Syndrom. Rate $\frac{k}{n} \to 1$ für wachsendes m

Idee um mehr als einen Bitfehler korrigieren zu können: mit m Zeilen in H → 1 Fehler

weitere m Zeilen → 2 Fehler

etc.

Aber wie?

Fast 10 Jahre Forschung um gute und "elegante" Lösung zu finden.

→ Theorie der endlichen Körper wird benötigt.

Endliche Körper

Ein endlicher Körper ist eine Menge M mit nur endlich vielen Elementen, wobei zwischen diesen Elementen eine Addition und eine Multiplikation erklärt ist und die sogenannten Körperaxiome erfüllt sind.

Körperaxiome: M ist abelsche Gruppe bzgl. Addition
$M \setminus \{0\}$ ist " " " Multiplikation
Es gilt das Distributivgesetz

$$a \cdot (b + c) = ab + ac$$

Eine Gruppe ist eine Menge mit einer Verknüpfung bei der gilt:

Abgeschlossenheit $a \odot b = c$ (Verknüpfung; a, b in Menge; c auch in Menge)

neutrales Element $e \odot a = a$

inverses Element $a \odot \bar{a} = e$

Assoziativgesetz gilt $a \odot (b \odot c) = (a \odot b) \odot c$

abelsche Gruppe: Kommutativgesetz gilt

$$a \odot b = b \odot a$$

Einfachstes Beispiel

$\{0,1\}$ mit

+	0	1
0	0	1
1	1	0

↑ XOR

und

·	0	1
0	0	0
1	0	1

↑ UND

⇒ Addition und Multiplikation modulo 2.

Weiteres Beispiel: $\{0,1,2\}$

Addition und Multiplikation modulo 3.

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

·	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Weiter mit: $\{0,1,2,3\}$

Add. u. Mult. modulo 4

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

·	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

$2 \cdot 2 = 0 \bmod 4$ (Nullteiler)

2 hat keine mult. Inverse

Addition und Multiplikation modulo einer ganzen Zahl führen zu endlichen Körpern wenn diese Zahl eine Primzahl ist.

Der endliche Körper mit p Elementen wird oft mit $GF(p)$ bezeichnet.

GF: Galoisfeld, p: Primzahl

Die besonders interessanten Fälle mit 2^m Elementen erhält man nicht durch "einfache" modulo-Rechnung.

+	0	1	α	β
0	0	1	α	β
1	1	0	β	α
α	α	β	0	1
β	β	α	1	0

·	0	1	α	β
0	0	0	0	0
1	0	1	α	β
α	0	α	β	1
β	0	β	1	α

$GF(2^2)$ ↑ Körperaxiome erfüllt

$\alpha + \beta = 1$ $\qquad$ $\alpha \cdot \alpha = \beta$

$\alpha + \alpha^2 = 1 \quad \Rightarrow \quad x^2 + x + 1 = 0$

Analogie: komplexe Zahlen

$p(x) = x^2 + 1$ NS von $x^2 + 1 = 0 \rightarrow$ Def.: i

$z_1 = u_1 + i v_1$
$z_2 = u_2 + i v_2$
} i^2 wird ersetzt durch -1.

$p(x)$ ist über $\mathbb{R}$ irreduzibel (keine NS in $\mathbb{R}$).

Bei $GF(2^2)$: α ist Wurzel von $p(x) = x^2 + x + 1$

Das Polynom x^2+1 kann nicht genommen werden da gilt $(x+1)\cdot(x+1) = x^2 + \underbrace{2x}_{=0 \bmod 2} + 1 = x^2 + 1$ ↓ Nullteiler

$x^2 + x + 1$ ist irreduzibel über $GF(2)$.

Darstellung des Körpers $GF(2^2)$ mit α:

i	α^i	α	1
0	1	0	1
1	α	1	0
2	α^2	1	1
3	α^3	0	1

$\alpha^3 = (\alpha+1)\alpha = \alpha^2 + \alpha = 1$

Weiteres Beispiel: $GF(2^3)$

Irreduzibles Polynom vom Grad 3:

z. B. durch Probieren finden: $p(x) = x^3 + x + 1$

Nullstelle von $p(x)$: α

i	α^i	α^2 α 1
0	1	0 0 1
1	α	0 1 0
2	α^2	1 0 0
3	α^3	0 1 1
4	α^4	1 1 0
5	α^5	1 1 1
6	α^6	1 0 1
7	α^7	0 0 1

Darstellung mit Standard basis

$\alpha^3 = \alpha + 1$

$\alpha^7 = 1$

Exponentendarstellung

Addition: komponentenweise modulo 2

Multiplikation: Je nach Darstellung: Exponenten addieren

oder Polynommultiplikation modulo $p(\alpha)$.

Vorgehensweise funktioniert für alle Körper $GF(p^m)$:

Finde irreduzibles Polynom vom Grad m über $GF(p)$. Rechnung mod p und mod $p(x)$.

Besonders "beliebt": Irreduzible Polynome, deren Wurzel ein primitives Element ist.

primitives Element: α ist ein primitives Element, wenn die Potenzen von α alle von Null verschiedenen Elemente erzeugt, d.h. $\alpha^0, \alpha^1, \alpha^2, \ldots \alpha^{q-2}$ sind alle Elemente von $GF(q) \setminus \{0\}$.

$\Rightarrow \alpha^{q-1} = 1$ bzw. $\gamma^q = \gamma$ in $GF(q)$

$\forall \gamma$ aus $GF(q)$

Für $q = p \rightarrow$ Fermatscher Satz: $x^p = x \bmod p$

irreduzible Polynome, die ein primitives Element als Wurzel haben nennt man primitive Polynome. Diese werden in Codierungstheorie bevorzugt benutzt.

Liste von primitiven binären Polynomen zur Konstruktion von $GF(2^m)$ $m = 2, 3, \ldots, 8$

m	$p(x)$
2	$x^2 + x + 1$
3	$x^3 + x + 1$
4	$x^4 + x + 1$
5	$x^5 + x^2 + 1$
6	$x^6 + x + 1$
7	$x^7 + x^3 + 1$
8	$x^8 + x^4 + x^3 + x^2 + 1$

Anmerkung: histor. hat der Begriff primitives Polynom eine andere Bedeutung!

Zusammenfassung der Übungsaufgaben

1.) Skizzieren Sie die Kanalkapazität C des binären Symmetrischen Kanals in Abhängigkeit von der Bitfehlerwahrscheinlichkeit.

2.) Bestimmen Sie die Mindestdistanz des mit der Generatormatrix $\left(I_4 \mid \begin{matrix} 0&1&1 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{matrix}\right)$ generierten Codes (→ S.13)

3.) Geben Sie Generator- und Prüfmatrix des binären $[n, 1, n]$ Wiederholungscodes an.

4.) Geben Sie Generator- und Prüfmatrix des binären $[n, n-1, 2]$ Codes an, der gebildet wird indem man jeden Nachrichtenvektor der Länge $(n-1)$ um ein Prüfbit ergänzt, sodaß das Gewicht des Codeworts gerade ist.

5.) Bestimmen Sie V_t für Binärcodes.

6.) Erstellen Sie ein Computerprogramm zur Berechnung des ggT zweier ganzer Zahlen

7.) Konstruieren Sie den Körper $GF(2^4)$.

8.) Konstruieren Sie den Körper $GF(3^2)$.