Kodierungstheorie

Mathematik für die Datenkommunikation

Prof. Dr. Andreas Vogt

Fachhochschule
Nordwestschweiz

Kodierungstheorie

Wie werden üblicherweise Nachrichten übertragen?

Quelle

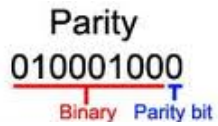
Texte
Bilder
Musik

Quellenkodierung

Umwandlung in eine möglichst kurze Folge gewisser Zeichen

Kanalkodierung

Hinzufügen von Redundanz



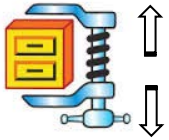
fehleranfällige Übertragung

Kanaldekodierung

An Hand der Redundanz wird festgestellt, ob Fehler bei der Übertragung passiert sind, und diese gegebenenfalls korrigiert.

Quellendekodierung

Umwandlung in das Ausgangssignal



Senke

Quellenkodierung

Definition

Es seien A_Q und A_K zwei endliche Mengen, das Quellenalphabet und das Kanalalphabet.

Eine injektive Abbildung $f : A_Q \to A_K^+$ heisst in diesem Zusammenhang Kodierung oder auch Code.

$f(x) \neq f(y)$ falls $x \neq y$

> Zur Erinnerung: A_K^+ besteht aus allen Wörtern, die man duch Hintereinderschreiben von Symbolen aus A_K erhalten kann.
>
> Meist ist $A_K = \{0, 1\}$. Dann ist $A_K^+ = \{0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$.

$f(a)$ heisst Codewort von a.

Ein Element aus A_Q^+ bezeichnen wir als Nachricht.

Die Kodierung wird vermöge $f^+(a_1 \ldots a_n) = f(a_1) \ldots f(a_n)$ auf Nachrichten erweitert.

Beispiel: Morse Code

A •-	J •---	S •••
B -•••	K -•-	T -
C -•-•	L •-••	U ••-
D -••	M --	V •••-
E •	N -•	W •--
F ••-•	O ---	X -••-
G --•	P •--•	Y -•--
H ••••	Q --•-	Z --••
I ••	R •-•	

Hier ist: $A_Q = \{A, B, \ldots, Z\}$

$A_K = \{\bullet, -\}$

f ist durch die Tabelle definiert: $f(A) = \bullet-$

$f(B) = -\bullet\bullet\bullet$

⋮

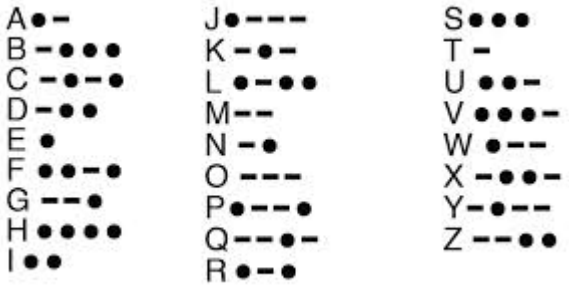
Es gilt z.B. $f^+(AB) = \bullet- -\bullet\bullet\bullet$

Quellenkodierung

Definition

Ein Code f heisst eindeutig dekodierbar, wenn auch die Abbildung f^+ injektiv ist.

Beispiel: Morse Code



Sie empfangen ••---•.

Was kann das alles bedeuten?

- E E T T T E
- I M N
- und noch vieles andere

Der Morse-Code ist also nicht eindeutig dekodierbar.

Man muss also hinreichend grosse Pausen zwischen den einzelnen Buchstaben einer Nachricht machen.

Dies ist sehr ineffizient.

Ein Code heisst präfixfrei, wenn kein Codewort das Präfix eines anderen Codewortes ist.

$x \in A_K^+$ heisst dabei Präfix von $y \in A_K^+$, wenn y mit x beginnt.

Beispiel: 0010 ist Präfix von 0010111.

0010 ist kein Präfix von 001111.

Der Morse Code ist nicht präfix-frei: $f(E) = \bullet$ ist etwa Präfix von $f(A) = \bullet\text{–}$.

Quellenkodierung

Beispiel:

Es sei $A_Q = \{A, B, C, D\}$, $A_K = \{0, 1\}$ und f_1, f_2, f_3, f_4 durch die folgende Tabelle definiert:

	f_1	f_2	f_3	f_4
A	0	0	0	0
B	0	1	01	10
C	1	00	011	110
D	10	11	0111	111

Es ist z.B. $f_1(A) = 0$ und $f_3(C) = 011$.

Welche Abbildung ist ein Code/präfix-frei/eindeutig dekodierbar?

f_1	nicht injektiv, also kein Code
f_2	injektiv, also Code, nicht präfix-frei, nicht eindeutig dekodierbar
f_3	injektiv, also Code, nicht präfix-frei, trotzdem eindeutig dekodierbar Die 0 zeigt neues Zeichen der Nachricht an. Beispiel: 01 011 01 = B C B
f_4	injektiv, also Code, präfix-frei, eindeutig dekodierbar

Im Folgenden beschränken wir uns auf den wichtigsten Fall $A_K = \{0, 1\}$.

Quellenkodierung

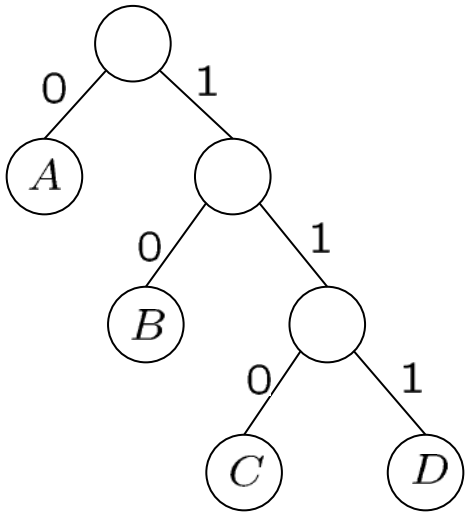
Jedem präfix-freien Code entspricht ein Baum, bei dem die Codewörter an den Blättern stehen:

Nach links ausgehende Kanten labeln wir mit 0, nach rechts ausgehende Kanten mit 1. Jedes Codewort kann dann als Codierung eines von der Wurzel ausgehenden Pfades in einem Baum aufgefasst werden. Den Knoten am Ende dieses Pfades labeln wir mit dem zu codierenden Zeichen.

Beispiel:

	f_4
A	0
B	10
C	110
D	111

entspricht:



Quellenkodierung

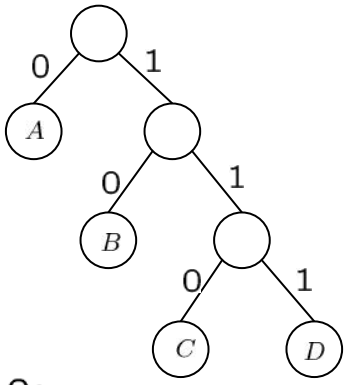
Präfix-freie Codes sind stets eindeutig dekodierbar, und die Dekodierung ist effizient möglich:

> Für jede Nachricht, die zu dekodieren ist, läuft man den entsprechenden Baum entlang, bis man ein Blatt erreicht, gibt das dazugehörige Zeichen aus und fängt wieder bei der Wurzel an.

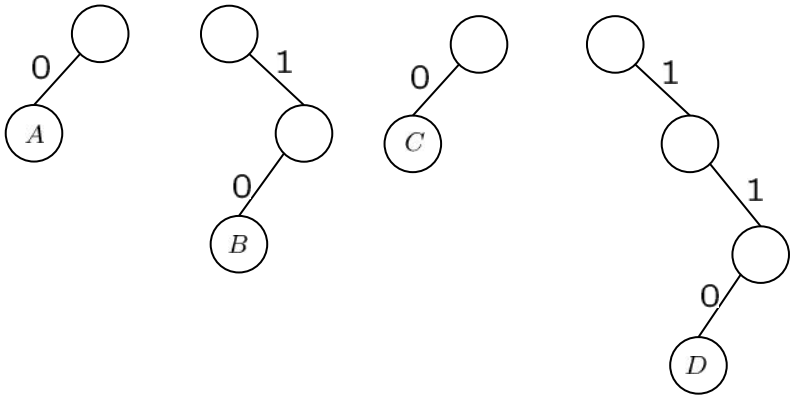
Beispiel:

	f_4
A	0
B	10
C	110
D	111

entspricht:



Dekodierung von z.B. 0100110:



Wir erhalten: $ABAC$

Quellenkodierung

Eine wichtige Klasse von präfix-freien Codes sind Blockcodes:

Definition:

Es sei $l \in \mathbb{N}$. Ein Code $f : A_Q \to A_K^+$ heisst Blockkcode der Länge l, falls $\forall x \in A_Q$: $|f(x)| = l$, wobei $|z|$ die Anzahl an Zeichen in z bedeutet.

Bei Blockcodes sind die Codewörter also alle gleich lang.

Beispiel: $A_Q = \{A, B, C, D\}$

	f
A	000
B	100
C	110
D	111

Dies ist ein Blockcode der Länge 3.

Satz: Blockcodes sind stets präfix-frei.

Begründung:

Da alle Codewörter gleich lang sind, kann keins so anfangen wie ein anderes.

Auf Grund der simplen Struktur haben Blockcodes in gewissen Situationen Vorteile, es gibt allerdings einen gravierenden Nachteil:

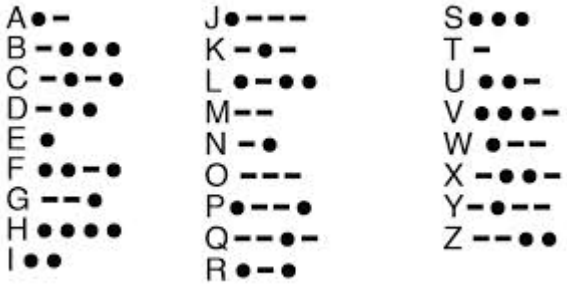
Die Codewörter sind manchmal länger als benötigt, was Ressourcen bei der Übertragung bzw. Speicherung verschwendet.

Ziel: Konstruktion von Codes mit möglichst kurzen Codewörtern.

Quellenkodierung

Wir hatten schon gesehen, dass der Morse Code nicht eindeutig dekodierbar ist.

Morse Code



Was kann man Positives über den Morse Code sagen?

Der Morse Code berücksichtigt, dass E in den meisten Sprachen im Vergleich z.B. zu Z öfter vorkommt:

Die Anzahl der zu übertragenden Zeichen für E ist deutlich geringer als die für Z.

Wir lernen nun ein Verfahren kennen, mit dem man zu einem Quellenalphabet, bei dem man weiss, wie oft im Schnitt die einzelnen Zeichen auftreteten, einen Code konstruieren kann, bei dem die Codewörter im Schnitt am kürzesten sind.

Der Code wird zudem präfix-frei sein.

Quellenkodierung

Gegeben: Alphabet $A_Q = \{a_1, a_2, a_3, \ldots, a_n\}$
mit Auftrittswahrscheinlichkeiten $p_1, p_2, \ldots, p_n$.

Prozentangaben, wie oft das entsprechende Zeichen vorkommt

Gesucht: Code $f : A_Q \to \{0,1\}^+$, welcher im Schnitt die kürzesten Codewörter hat.
Die mittlere Wortlänge $\sum_{i=1}^{n} p_i \cdot |f(a_i)|$ soll möglichst klein sein.

Dazu eignet sich die sogenannte Huffman-Kodierung, welche optimal in dem Sinne ist, dass die Codewörter im Schnitt am kürzesten sind, wenn zufällig Nachrichten geschickt werden und die Zeichen der Nachricht unabhängig voneinander gemäss den gegebenen Auftrittswahrscheinlichkeiten vorkommen.

Wir illustrieren das Vorgehen an folgendem Beispiel:

A_Q	Auftrittswahrscheinlichkeit
a	7%
b	8%
c	5%
d	5%
e	75%

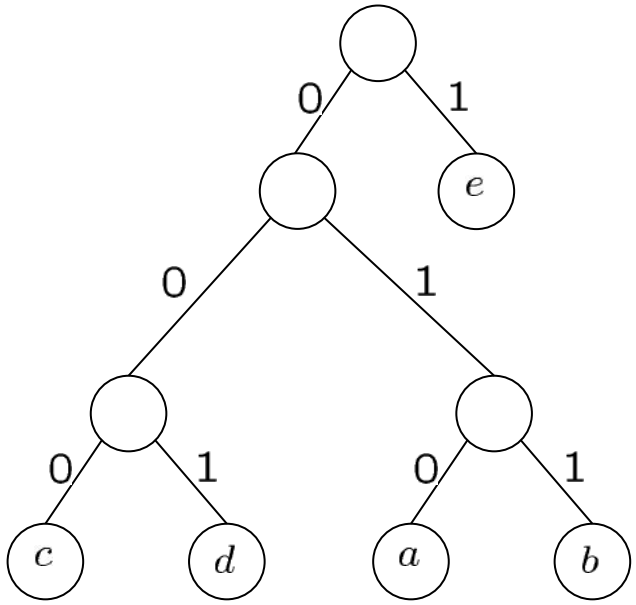
Die beiden unwahrscheinlichsten Zeichen sollten die längsten Codewörter erhalten.

Wir konstruieren den zum Code gehörenden Baum also von hinten ausgehend von diesen beiden Zeichen.

Quellenkodierung

Die beiden unwahrscheinlichsten Zeichen werden zu einem neuen Zeichen zusammengefasst. Es müssen dann nur noch die verbleibenden Zeichen und dieses neue Zeichen kodiert werden.

Dies wird mit dem selben Ansatz solange gemacht, bis der Baum fertig konstruiert ist.



A_Q nach erstem Schritt	Auftrittswahrscheinlichkeit
a	7%
b	8%
cd	5%+5%=10%
e	75%

A_Q nach zweitem Schritt	Auftrittswahrscheinlichkeit
ab	7%+8%=15%
cd	10%
e	75%

A_Q nach drittem Schritt	Auftrittswahrscheinlichkeit
abcd	15%+10%=25%
e	75%

Wir erhalten als Kodierung:

A_Q	f
a	010
b	011
c	000
d	001
e	1

Quellenkodierung

Die Huffman-Kodierung wird in der Praxis oft eingesetzt.

Wenn man damit eine Textdatei komprimieren möchte, kann man etwa wie folgt vorgehen:

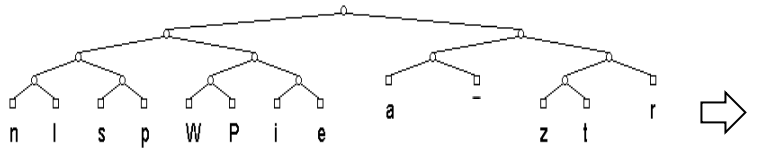
1) Man bestimmt für jeden Buchstaben die relative Häufigkeit.
2) Man konstruiert dafür einen optimalen Huffman-Code.
3) Man speichert den Huffman-Code (etwa den zugehörigen Baum) mit ab.

Beispiel: Textdatei: `Wir sparen Platz`

Entspricht im ASCII-Code:
01010111011010010111001000100000011100110111000001100001011100100110010101101110001000000101000001101100011000010111010001111010001011 10

Huffman-Kodierung:

A_Q	Auftrittswahrscheinlichkeit
W	1/16
i	1/16
r	1/8
_	1/8
s	1/16
p	1/16
a	1/8
e	1/16
n	1/16
P	1/16
l	1/16
t	1/16
z	1/16



A_Q	Codewort
W	0100
i	0110
r	111
_	101
s	0010
p	0011
a	100
e	0111
n	0000
P	0101
l	0001
t	1101
z	1100

Der Text ergibt sich also zu:
0100011011110100100011100111011100001010101000110011011100

Das ist deutlich kürzer als in der ASCII-Kodierung.

(Es muss allerdings auch noch der Baum bzw. die Kodierungstabelle abgespeichert werden.)

Quellenkodierung

Die Huffman-Kodierung ist in vielen Fällen nicht optimal.

Zur Erinnerung: Bei der Huffman-Kodierung sind die Codewörter im Schnitt am kürzesten, wenn zufällig Nachrichten geschickt werden, und die Zeichen der Nachricht unabhängig voneinander gemäss den gegebenen Auftrittswahrscheinlichkeiten vorkommen.

Die einzelnen Zeichen einer Nachricht sind meist nicht unabhängig voneinander:

Bei deutschen Texten kommt hinter `c` z.B. sehr oft ein `h`.

Eventuell wäre es also günstiger, ein eigenes Codewort für `ch` zu haben.

Oder auch für `der`, `die` oder `das`... ← Kommen in deutschen Texten sehr oft vor.

Vielleicht ist aber nur die Einleitung deutsch, und dann geht es mit Englisch weiter.

Für diese Anwendungsgebiete eignen sich daher eher "dynamische Verfahren".

Häufig, etwa in den Graphikformaten `gif` oder `tiff`, wird der Lempel-Ziv-Welch-Algorithmus verwendet.

Dieser Algorithmus erzeugt dynamisch ein Wörterbuch mit häufig auftretenden Zeichenketten, um häufig auftretenden Zeichenketten eine Abkürzung zuzuweisen.

Dabei kann aus der Kodierung das Wörterbuch rekonstruiert werden, es muss also nicht mit übermittelt werden.

Wir können auf den Algorithmus (leider) nicht genauer eingehen.

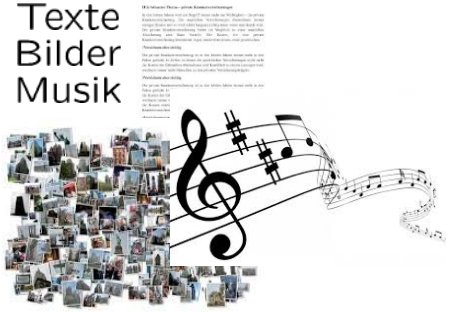
Serie 03, bis A.9

Kodierungstheorie

Wie werden üblicherweise Nachrichten übertragen?

Quelle

Texte
Bilder
Musik

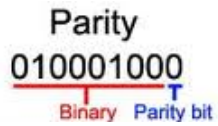


Quellenkodierung

Umwandlung in eine möglichst kurze Folge gewisser Zeichen

Kanalkodierung

Hinzufügen von Redundanz



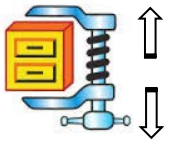
fehleranfällige Übertragung

Kanaldekodierung

An Hand der Redundanz wird festgestellt, ob Fehler bei der Übertragung passiert sind, und diese gegebenenfalls korrigiert.

Quellendekodierung

Umwandlung in das Ausgangssignal



Senke

Kanalkodierung

Bisher haben wir eine Nachricht möglichst kurz kodiert, aber immer noch so lang, dass man die ursprüngliche Nachricht wiederherstellen konnte.

Jetzt machen wir die Nachricht künstlich länger.

Ziel:

Man sollte erkennen können, ob während der Übertragung Fehler passiert sind – und man sollte diese Fehler gegebenenfalls korrigieren können.

Beispiel: ISBN 13

Die ISBN (International Standard Book Number) dient zur eindeutigen Kennzeichnung von Büchern oder anderen Veröffentlichungen.

Aufbau: 978-3-48657-785-3

- Prüfziffer
- Titelnummer: wird vom Verlag selber vergeben
- Verlagsnummer: hier: Oldenbourg-Verlag
- Ländernummer: etwa 3 für den deutschsprachigen Raum
- Präfix: 978 oder 979 (je nach Buch)

Wenn z_i die i-te Ziffer der ISBN bezeichnet, dann gilt für die Prüfziffer z_{13} die folgende Formel:

$$z_{13} = \left(10 - \underbrace{\left(\sum_{i=1}^{12} z_i \cdot 3^{(i+1) \bmod 2}\right)}_{= z_1 + 3z_2 + z_3 + 3z_4 + z_5 + 3z_6 + z_7 + 3z_8 + z_9 + 3z_{10} + z_{11} + 3z_{12}} \bmod 10\right) \bmod 10$$

Im Beispiel: $z_{13} = (10-(9+3\cdot 7+8+3\cdot 3+4+3\cdot 8+6+3\cdot 5+7+3\cdot 7+8+3\cdot 5) \bmod 10) \bmod 10 = (10 - 147 \bmod 10) \bmod 10 = (10-7) \bmod 10 = 3$

Beispiel: ISBN

Satz: Bei einer gültigen ISBN-Nummer gilt:

$$(z_1 + 3z_2 + z_3 + 3z_4 + z_5 + 3z_6 + z_7 + 3z_8 + z_9 + 3z_{10} + z_{11} + 3z_{12} + z_{13}) \bmod 10 = 0.$$

Beweis:

$$(z_1 + 3z_2 + z_3 + 3z_4 + z_5 + 3z_6 + z_7 + 3z_8 + z_9 + 3z_{10} + z_{11} + 3z_{12} + z_{13}) \bmod 10 =$$

$$= \left(\sum_{i=1}^{12} z_i \cdot 3^{(i+1) \bmod 2} + z_{13} \right) \bmod 10$$

$$= \left(\sum_{i=1}^{12} z_i \cdot 3^{(i+1) \bmod 2} + \left(10 - \left(\sum_{i=1}^{12} z_i \cdot 3^{(i+1) \bmod 2} \right) \bmod 10 \right) \bmod 10 \right) \bmod 10$$

$$= \left(\sum_{i=1}^{12} z_i \cdot 3^{(i+1) \bmod 2} + 10 - \sum_{i=1}^{12} z_i \cdot 3^{(i+1) \bmod 2} \right) \bmod 10$$

$$= 10 \bmod 10 = 0.$$

Beispiel: ISBN

Satz: Wenn beim Einlesen oder Übertragen einer ISBN eine einzelne Stelle fehlerhaft ist, wird dies stets festgestellt.

Beweis:

Wenn nun ein Fehler bei einer ungeraden Stelle passiert, etwa anstelle z_1 wird z_1' eingelesen, dann gilt:

$$\begin{aligned}(z_1' + 3z_2 + z_3 + 3z_4 + z_5 + 3z_6 + z_7 + 3z_8 + z_9 + 3z_{10} + z_{11} + 3z_{12} + z_{13}) \bmod 10 = \\ = (z_1' - z_1 + z_1 + 3z_2 + z_3 + 3z_4 + z_5 + 3z_6 + z_7 + 3z_8 + z_9 + 3z_{10} + z_{11} + 3z_{12} + z_{13}) \bmod 10 = \\ = (z_1' - z_1) \bmod 10 \neq 0\end{aligned}$$

Der Fehler würde also auffallen. (Analog bei $z_3, z_5, z_7, z_9, z_{11}, z_{13}$.)

Wenn nun ein Fehler bei einer geraden Stelle passiert, etwa anstelle z_2 wird z_2' eingelesen, dann gilt:

$$\begin{aligned}(z_1 + 3z_2' + z_3 + 3z_4 + z_5 + 3z_6 + z_7 + 3z_8 + z_9 + 3z_{10} + z_{11} + 3z_{12} + z_{13}) \bmod 10 = \\ = (3z_2' - 3z_2 + z_1 + 3z_2 + z_3 + 3z_4 + z_5 + 3z_6 + z_7 + 3z_8 + z_9 + 3z_{10} + z_{11} + 3z_{12} + z_{13}) \bmod 10 = \\ = (3z_2' - 3z_2) \bmod 10 \ = (3 \cdot (z_2' - z_2)) \bmod 10 \neq 0\end{aligned}$$

(Das Dreifache einer Zahl x mit $-9 \leq x \leq 9$ und $x \neq 0$ ist nie durch 10 teilbar.)

Der Fehler würde also auffallen. (Analog bei $z_4, z_6, z_8, z_{10}, z_{12}$.)

Beispiel: ISBN

Man erkennt also, dass z.B. 978-3-48657-789-3 nicht korrekt ist.

Es ist allerdings nicht möglich, zu sagen, an welcher Stelle der Fehler passiert ist.

Geschweige denn, dass man an Hand der Prüfziffer den Fehler direkt (ohne nochmaliges Einlesen/Übermitteln) beheben kann.

Man hat aber auch nur eine Prüfziffer drangehängt. In vielen Anwendungen ist es problemlos möglich, beim Feststellen eines Fehlers, ein erneutes Einlesen/Übertragen zu veranlassen.

Dann reichen Codes, die Fehler nur feststellen, diese aber nicht korrigieren können, aus. (Diese Codes kommen eben mit weniger zusätzlichen Prüfinformationen aus.)

In anderen Anwendungen reicht das Erkennen eines Fehlers nicht aus, da das erneute Anfordern einer dann (hoffentlich) korrekten Nachricht nicht möglich ist oder zu lange dauert (etwa bei der Raumfahrt).

bis 3.11

Kanalkodierung

Beispiel: Wiederholungscode der Länge 3

Wir möchten ein Bit (also 0 oder 1) über einen Kanal übertragen.

Dabei senden wir das Zeichen dreimal.

Nachricht	Gesendete Nachricht
0	000
1	111

Nachricht 0  000  001 

001 kann nicht sein, es muss also was schief gelaufen sein.

Bei dieser Kodierung erkennt man also, wenn ein oder zwei Zeichen falsch übertragen werden.

Bei drei Fehlern würde aus 000 eben 111 werden, und dann merkt man nicht, das was falsch gelaufen ist.

Man erhält 001. Wie sollte man dekodieren?

Am plausibelsten scheint es zu sein, dass die Ursprungsnachricht 0 war.

(Denn es ist sinnvoll anzunehmen, dass zwei Fehler bei der Übertragung seltener vorkommen, als ein Fehler.)

Wenn man diese Regel zum Dekodieren verwendet, also wir suchen das Codewort, welches an den meisten Stellen mit dem empfangenen Wort übereinstimmt, und geben die dazu passende Nachricht aus, dann gilt:

Wenn bei der Übertragung höchstens ein Zeichen falsch übertragen wird, dann erhält man so die korrekte Nachricht zurück.

Kanalkodierung

Im Folgenden betrachten wir Blockcodes der Länge n, wobei die Nachrichten Elemente von A^m sind, also injektive Abbildungen $f : A^m \to A^n$. Hierbei bezeichnet A das Kanalalphabet (was meistens, aber nicht immer, $\{0, 1\}$ ist).

Die Menge der Codewörter, also die Menge $f(A^m)$ bezeichnen wir mit $\mathcal{C}$.

Meist schreibt man die Elemente der Menge A^m in der Form $(a_1, a_2, \ldots, a_m)$. Im Bereich der Kodierung lässt man die Klammern und die Kommas meist weg und schreibt kurz $a_1 a_2 \ldots a_m$.

Beispiele:

(1) Beim Wiederholungscode der Länge 3 ist $A = \{0, 1\}$, $m = 1$ und $n = 3$.

Die Abbildung $f : A \to A^3$ ist so definiert: $f(z) = (z, z, z)$

Hier ist $\mathcal{C} = \{000, 111\}$.

(2) Bei ISBN-13 ist also $A = \{0, 1, \ldots, 9\}$, $m = 12$ und $n = 13$.

Die Abbildung $f : A^{12} \to A^{13}$ ist so definiert:

$$f(z_1, z_2, \ldots, z_{12}) = (z_1, \ldots, z_{12}, z_{13})$$

wobei z_{13} gemäss der Formel auf Folie 3.16 bestimmt wird.

Hier ist $\mathcal{C} = \{\text{alle gültigen ISBN-Nummern}\}$.

Kanalkodierung

Definition: Für zwei Elemente $x = (x_1, \ldots, x_n) \in A^n$ und $y = (y_1, \ldots, y_n) \in A^n$ ist

$$d(x, y) = \left| \left\{ i \in \{1, 2, \ldots, n\} \mid x_i \neq y_i \right\} \right|.$$

$d(x, y)$ heisst Hamming-Distanz oder Hamming-Abstand von x und y.

Beispiele: (1) $x = 10010$, $y = 10011 \Rightarrow d(x, y) = 1$

(2) $x = 111$, $y = 000 \Rightarrow d(x, y) = 3$

Definition:

Der Dekodiervorgang, das Codewort y zu suchen, welches zum empfangenen Wort x den kleinsten Hamming-Abstand hat und die dazu passende Nachricht auszugeben, heisst Minimaldistanz-Dekodierung.

Falls es zwei Codeworte y_1 und y_2 gibt, die den selben (minimalen) Abstand zu x haben, wird ein Fehler ausgegeben.

Beispiele: Wir betrachten folgenden Code:

Nachricht	Codewort
00	0000
01	0111
10	1101
11	1111

Was ist die Minimaldistanz-Dekodierung von:

(a) 0000	00
(b) 0001	00
(c) 1010?	Fehler, die Codes 0000 und 1111 haben die gleiche minimale Hamming-Distanz.

Kanalkodierung

Definition: Für einen Code f mit zugehöriger Menge $\mathcal{C}$ von Codewörtern heisst

$$d(\mathcal{C}) = \min\{d(x,y) \mid x,y \in \mathcal{C}, x \neq y\}$$

Minimaldistanz.

Beispiele:

(1) Beim Wiederholungscode der Länge 3 ist $\mathcal{C} = \{000, 111\}$.

Die Minimaldistanz ist also 3.

(2) Bei einem Code sei $\mathcal{C} = \{0000, 1010, 0101, 1111\}$.

Die Minimaldistanz ist dann 2.

Definition: Ein Code $f : A^m \to A^n$ mit Minimaldistanz d heisst (n, m, d)-Code.

Beispiel: Der Wiederholungscode der Länge 3 ist ein $(3, 1, 3)$-Code.

bis 3.12

Kanalkodierung

Es gilt folgender wichtiger Satz:

Satz: Für einen Code mit Minimaldistanz d gilt:

(1) Wenn bei der Übertragung $\leq d-1$ Zeichen falsch übertragen werden, stellt der Empfänger fest, dass etwas falsch gelaufen ist.

(2) Wenn bei der Übertragung $\leq \lfloor \frac{d-1}{2} \rfloor$ Zeichen falsch übertragen werden, liefert die Minimaldistanz-Dekodierung das korrekte Resultat.

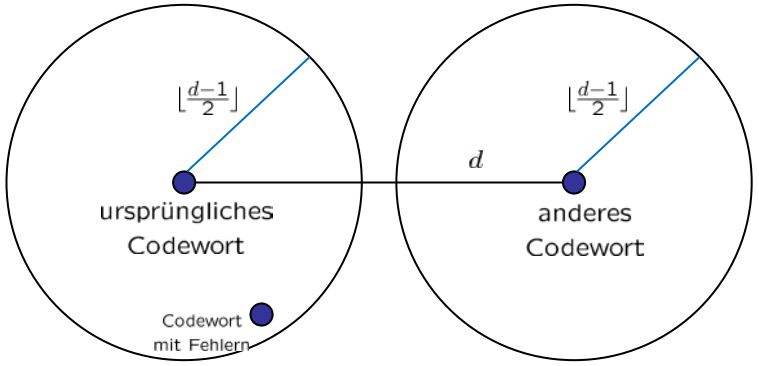
$\frac{d-1}{2}$ abgerundet

Beweis:

(1) Je zwei Codewörter unterscheiden sich an mindestens d Stellen. Es müssen also mindestens d Zeichen falsch übertragen werden, damit aus einem Codewort ein anderes gültiges Codewort wird.

(2) Wenn $\leq \lfloor \frac{d-1}{2} \rfloor$ Fehler passiert sind, dann ist der Hamming-Abstand zum ursprünglichen Codewort $\leq \lfloor \frac{d-1}{2} \rfloor$.

Und da zwei Codewörter mindestens den Abstand d voneinander haben, ist der Abstand des verfälschten Codewortes zu jedem anderen Codewort (ausser dem urprünglichen) mindestens $d - \lfloor \frac{d-1}{2} \rfloor > \lfloor \frac{d-1}{2} \rfloor$. Es wird also zum richtigen Codewort dekodiert.



Kanalkodierung

Wir möchten zwei Bits übertragen.

Wir wiederholen dazu die Nachricht dreimal:

Nachricht	Codewort
00	000000
01	010101
10	101010
11	111111

Die Minimaldistanz ist hier 3.

Es liegt ein $(6, 2, 3)$-Code vor.

Es kann also nur garantiert werden, dass ein einzelner Bitfehler korrigiert werden kann.

Wir betrachten nun folgenden Code:

Nachricht	Codewort
00	00000
01	01101
10	10110
11	11011

Die Minimaldistanz ist hier auch 3.

Es liegt ein $(5, 2, 3)$-Code vor.

Es kann also auch garantiert werden, dass ein einzelner Bitfehler korrigiert werden kann.

Der zweite Code ist also gleichwertig bezüglich der Feherlerekennung und der Fehlerkorrektur, er kommt aber mit einem Bit weniger aus!

Kanalkodierung

"Gute" Codes haben die Eigenschaft, trotz einer hohen Fehlererkennung/Fehlerkorrektur-Anzahl relativ wenige zusätzliche Bits zu benötigen.

Eine Abschätzung für die Minimaldistanz ist die Singleton-Schranke:

Satz: Bei einem (n, m, d)-Code gilt stets $d \leq n - m + 1$.

Beweis:

Es gilt $|\mathcal{C}| = q^m$, wobei q die Anzahl Zeichen des Alphabets und $\mathcal{C}$ wie üblich die Menge der Codewörter bezeichnet.

Da sich die Elemente von $\mathcal{C}$ paarweise an mindestens d Stellen unterscheiden, sind auch die Worte, die entstehen, wenn man jeweils die letzten $d-1$ Stellen abschneidet, noch alle verschieden.

Die Menge der abgeschnittenen Codewörter bezeichnen wir mit $\mathcal{C}'$.

Dann gilt einerseits $|\mathcal{C}| = |\mathcal{C}'|$ und andererseits

$|\mathcal{C}'| \leq q^{n-(d-1)}$, da jedes Element aus $\mathcal{C}$ aus $n - (d-1) = n - d + 1$ Elementen des Alphabets besteht.

Wir erhalten $q^m \leq q^{n-d+1}$, woraus direkt $m \leq n - d + 1$ bzw. $d \leq n - m + 1$ folgt.

Anstelle einer (auch interessanten) theoretischen Diskussion weiterer Kenngrössen von Codes (etwa der Informationsrate) und der Beziehungen dieser Kenngrössen untereinander, schauen wir uns den sogenannten Reed-Solomon-Code (RS-Code) im Detail an.

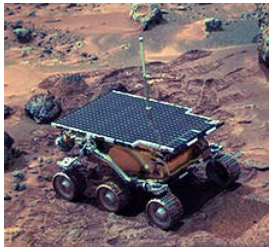
Kanalkodierung

RS-Codes sind in der Praxis sehr weit verbreitet:

- CD/DVD/Blue-ray

- Raumfahrt

  Voyager, Pathfinder...



- Bar-Codes

  QR-Code, PDF417...

- DSL

RS-Codes bauen auf Polynomen über endlichen Körpern auf.

bis 3.14

Kanalkodierung

Definition: Es sei $(K, +, \cdot)$ ein Körper.

Eine Abbildung $f : K \to K$ der Form $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$ heisst Polynom.

Die Zahlen $a_0, \ldots, a_n \in K$ heissen Koeffizienten des Polynoms.

Die grösste Zahl n mit $a_n \neq 0$ heisst Grad des Polynoms.

Der Grad des Nullpolynoms $f(x) = 0$ ist als $-\infty$ definiert.

Beispiele:

1) Es sei $(K, +, \cdot) = (\mathbb{R}, +, \cdot)$ und $f(x) = x^3 + 2x + 1$.
   Dann ist f ein Polynom vom Grad 3.
   Es gilt etwa $f(2) = 13$.

2) Es sei $(K, +, \cdot) = (\mathbb{Z}_5, +_5, \cdot_5)$ und $f(x) = x^2 + 3$.
   Dann ist f ein Polynom vom Grad 2.
   Es gilt etwa $f(2) = 2$.

Kanalkodierung

Polynome können einfach addiert und multipliziert werden:

Beispiele:

Es seien $(K, +, \cdot) = (\mathbb{Z}_5, +_5, \cdot_5)$ und $f(x) = 3x^3 + x + 1$ und $g(x) = x^2 + 2x + 4$.

(1) $f(x) + g(x) = 3x^3 + x^2 + 3x$

(2) $f(x) \cdot g(x) = 3x^5 + x^4 + 3x^3 + 3x^2 + x + 4.$

Satz:

Es sei $f(x)$ ein Polynom vom Grad n und $g(x)$ ein Polynom vom Grad m. Dann ist

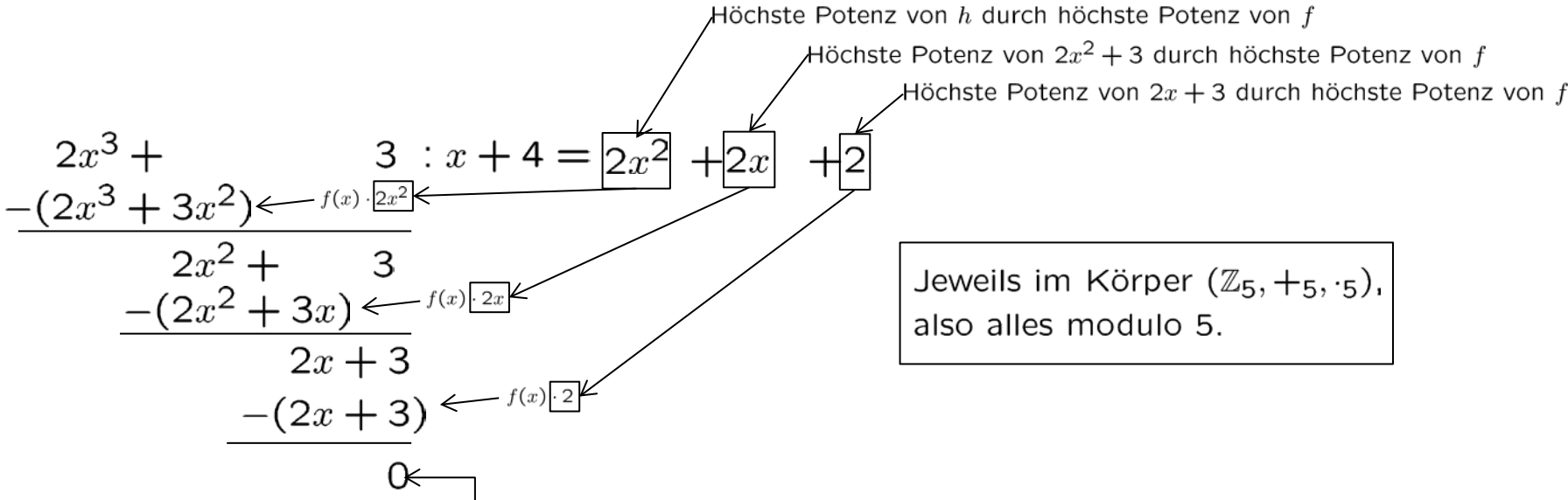
(1) $f(x) + g(x)$ ein Polynom vom Grad $\square$

(2) $f(x) \cdot g(x)$ ein Polynom vom Grad $\square$

Kanalkodierung

Wenn für drei Polynome $h(x), g(x)$ und $f(x)$ gilt, dass $h(x) = g(x) \cdot f(x)$, und $f(x)$ nicht das Nullpolynom ist, dann kann man aus $h(x)$ mit Polynomdivision durch $f(x)$ das Polynom $g(x)$ berechnen.
Wir schauen uns die Vorghehensweise an einem Beispiel an.

Dazu seien $(K, +, \cdot) = (\mathbb{Z}_5, +_5, \cdot_5)$, $h(x) = 2x^3 + 3$ und $f(x) = x + 4$.



Man mach solange weiter, bis hier ein Polynom $r(x)$ steht, bei dem sich die höchste Potenz nicht mehr durch die höchste Potenz von $f(x)$ teilen lässt.

Nur wenn $r(x)$ das Nullpolynom ist, ist $f(x)$ ein Faktor von $h(x)$.

$r(x)$ heisst $h(x)$ mod $f(x)$, das Polynom, was dann oben steht, heisst $h(x)$ div $f(x)$.

Es gilt (wie bei Zahlen auch) $f(x) = (f(x) \text{ div } g(x)) \cdot g(x) + f(x) \text{ mod } g(x)$.

Aufgabe: Berechnen Sie $f(x)$ mod $g(x)$ und $f(x)$ div $g(x)$ wobei

(1) $f(x) = x^4 + x + 1$, $g(x) = x^2 + x$ und $(K, +, \cdot) = (\mathbb{Z}_2, +_2, \cdot_2)$

(2) $f(x) = x^4 + 3x + 1$, $g(x) = 2x^2 + 3$ und $(K, +, \cdot) = (\mathbb{Z}_5, +_5, \cdot_5)$

(3) $f(x) = 2x^2 + 3x$, $g(x) = 4x + 1$ und $(K, +, \cdot) = (\mathbb{Z}_5, +_5, \cdot_5)$

Lösung:

(1) $f(x)$ mod $g(x) = 1$ und $f(x)$ div $g(x) = x^2 + x + 1$

(2) $f(x)$ mod $g(x) = 3x + 2$ und $f(x)$ div $g(x) = 3x^2 + 3$

(3) $f(x)$ mod $g(x) = 0$ und $f(x)$ div $g(x) = 3x$

Kanalkodierung

Satz:

Es sei $(K, +, \cdot)$ ein Körper und $x_0, \ldots, x_{n-1}$ paarweise verschiedene Elemente von K sowie $y_0, \ldots, y_{n-1}$ beliebige Elemente von K.

Dann existiert genau ein Polynom f vom Grad $\leq n-1$ mit $f(x_i) = y_i$ für alle $i = 0, \ldots, n-1$.

Dieses Polynom heisst Interpolationspolynom.

Man kann (relativ einfach) zeigen, dass die Koeffizientendeterminante des entsprechenden Gleichungssystems von Null verschieden ist. Wir verzichten hier darauf.

Beispiele:

(1) Es sei $(K, +, \cdot) = (\mathbb{R}, +, \cdot)$ und $x_0 = 0$, $x_1 = 1$, $x_2 = 2$ sowie $y_0 = 0$, $y_1 = 2$ und $y_2 = 0$.

Das Interpolationspolynom (vom Grad $\leq 3 - 1 = 2$) hat also die Form $p(x) = a_2 x^2 + a_1 x + a_0$.

Die Koeffizienten a_2, a_1 und a_0 ergeben sich aus $p(x_i) = y_i$, also aus:

$$\begin{array}{rcrcrcl} & & a_0 & = & 0 & & \\ a_2 & + & a_1 & + & a_0 & = & 2 \\ 4a_2 & + & 2a_1 & + & a_0 & = & 0 \end{array} \quad \rightarrow \quad \begin{array}{rcrcl} a_2 & + & a_1 & = & 2 \\ 4a_2 & + & 2a_1 & = & 0 \end{array} \quad \rightarrow \quad 2a_1 = 8 \quad \rightarrow \quad a_1 = 4, a_2 = -2$$

woraus man $p(x) = -2x^2 + 4x$ erhält.

Beispiele:

(2) Es sei $(K, +, \cdot) = (\mathbb{Z}_3, +_3, \cdot_3)$ und $x_0 = 0$, $x_1 = 1$ sowie $y_0 = 2$ und $y_1 = 1$.

Das Interpolationspolynom (vom Grad $\leq 2 - 1 = 1$) hat also die Form $p(x) = a_1 x + a_0$.

Die Koeffizienten a_1 und a_0 ergeben sich aus $p(x_i) = y_i$, also aus:

$$\begin{array}{rcrcl} & & a_0 & = & 2 \\ a_1 & + & a_0 & = & 1 \quad \rightarrow \; a_1 = 2 \; (= -1) \end{array}$$

woraus man $p(x) = 2x + 2$ erhält.

Aufgabe:

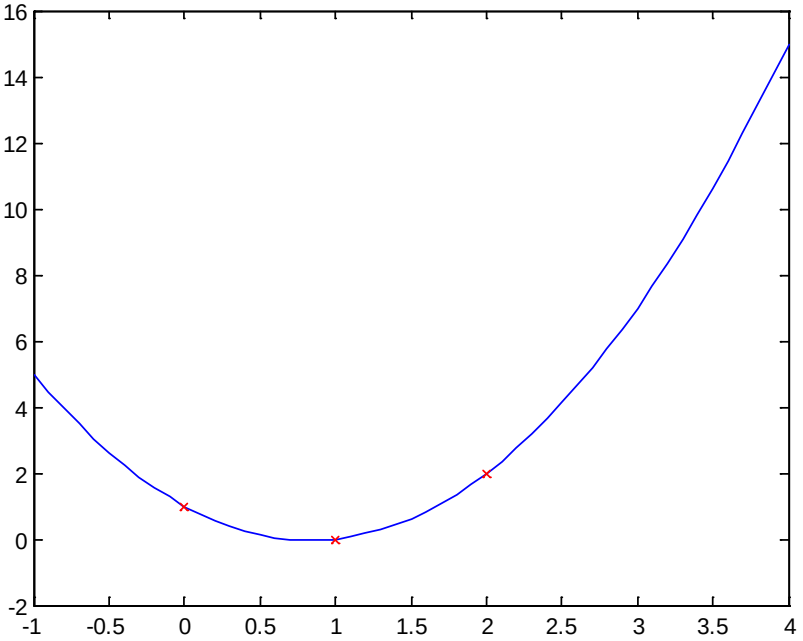
Berechnen Sie das Interpolationspolynom vom Grad ≤ 2 zu den Stützstellen $x_0 = 0$, $y_0 = 1$, $x_1 = 1$, $y_1 = 0$, $x_2 = 2$, $y_2 = 2$ im

(1) Körper $(\mathbb{R}, +, \cdot)$

(2) Körper $(\mathbb{Z}_5, +_5, \cdot_5)$

Lösung:

(1) $p(x) = \frac{3}{2}x^2 - \frac{5}{2}x + 1$



(2) $p(x) = 4x^2 + 1$

bis 3.19

Kanalkodierung

Satz:

Wenn $f(x)$ und $g(x)$ zwei Polynome von jeweils einem Grad $\leq n-1$ sind, und $f(x)$ und $g(x)$ an mindestens n Stellen übereinstimmen, dann sind die Koeffizienten von $f(x)$ und $g(x)$ alle gleich, insbesondere stimmen $f(x)$ und $g(x)$ an allen Stellen überein.

Beweis:

Die Funktion $h(x) = f(x) - g(x)$ ist ebenfalls ein Polynom vom Grad höchstens $n-1$.

Diese Funktion ist an n Stellen Null.

An diesen Stellen ist aber auch das Nullpolynom Null.

Da es mit dem vorigen Satz aber nur genau ein Poylnom vom Grad $\leq n-1$ gibt, welches an n Stellen Null ist, muss $h(x)$ das Nullpolynom sein, womit der Satz bewiesen ist.

Endliche Körper

Wir haben gesehen, dass $(\mathbb{Z}_p, +_p, \cdot_p)$ ein Körper ist, genau dann, wenn p eine Primzahl ist.

Insbesondere ist $(\mathbb{Z}_4, +_4, \cdot_4)$ kein Körper.

Man kann zeigen, dass es einen Körper mit q Elementen gibt, genau dann, wenn $q = p^k$ für eine Primzahl p und eine natürliche Zahl k ist.

Man kann zudem zeigen, dass alle Körper mit q Elementen zueinander isomorph sind, d.h. alle diese Körper sind im Wesentlichen gleich. Es gibt also im Wesentlichen nur einen solchen Körper, und den bezeichnet man mit $\mathbb{F}_q$.

Falls q eine Primzahl ist (und nur dann!), entspricht $\mathbb{F}_q$ also gerade $\mathbb{Z}_q$.

Kodierungstheorie – endliche Körper

nicht klausurrelevant!

Den Körper $\mathbb{F}_q$ kann man über Polynome definieren.

Und zwar kann man zeigen, dass es für jede Primzahl p und jede natürliche Zahl n ein irreduzibles Polynom $g(x)$ vom Grad n über dem Körper $(\mathbb{Z}_p, +_p, \cdot_p)$ gibt.
Ein Polynom heisst dabei irreduzibel, falls es sich nicht als Produkt zweier Polynome vom Grad ≥ 1 schreiben lässt.

Beispiele:

(1) $g(x) = x^3 + x$ ist nicht irreduzibel über dem Körper $(\mathbb{Z}_2, +_2, \cdot_2)$:
Es gilt nämlich $g(x) = x \cdot (x^2 + 1)$, und sowohl $f(x) = x$ als auch $h(x) = x^2 + 1$ haben Grad ≥ 1.

(2) $g(x) = x^2 + 1$ ist irreduzibel über dem Körper $(\mathbb{R}, +, \cdot)$:
Wenn $g(x) = f(x) \cdot h(x)$ mit Polynomen $f(x)$ und $g(x)$ jeweils vom Grad ≥ 1, dann müssten $f(x)$ und $h(x)$ jeweils Grad 1 haben.

Polynome vom Grad 1 haben die Form $ax + b$ mit einer Zahl $a \neq 0$.

Insbesondere gibt es dann eine Zahl $x_0 \in \mathbb{R}$ so dass $f(x_0) = 0$.

Damit wäre auch $g(x_0) = 0$, also $x_0^2 = -1$, was für keine Zahl $x_0 \in \mathbb{R}$ sein kann.

(3) $g(x) = x^2 + 1$ ist nicht irreduzibel über dem Körper $(\mathbb{Z}_2, +_2, \cdot_2)$:
Es gilt $x^2 + 1 = (x + 1) \cdot (x + 1)$.

Kodierungstheorie – endliche Körper

nicht klausurrelevant!

Es sei nun $q = p^n$ mit einer Primzahl p. Dann gibt es also ein irreduzibles Polynom $g(x)$ vom Grad n über dem Körper $(\mathbb{Z}_p, +_p, \cdot_p)$.

Für zwei Polynome $f(x)$ und $h(x)$ definiert man nun

$$f(x) +_{p^n} h(x) = f(x) + h(x)$$

und

$$f(x) \cdot_{p^n} h(x) = (f(x) \cdot h(x)) \mod g(x).$$

Üblicherweise identifiziert man $(a_0, \ldots, a_{n-1}) \in \mathbb{Z}_p^n$ mit dem Polynom $a_7x^7 + a_6x^6 + \ldots + a_1x + a_0$ vom Grad $< n$.

Über den Umweg über die Addition bzw. die Multiplikation der entsprechenden Polynome hat man also auf $\mathbb{Z}_p^n$ eine Addition $+_{p^n}$ und $\cdot_{p^n}$ definiert.

Dann kann man zeigen, dass $(\mathbb{Z}_p^n, +_{p^n}, \cdot_{p^n})$ ein Körper ist.

Dieser Körper enthält gerade $|\mathbb{Z}_p^n| = p^n$ Elemente.

nicht klausurrelevant!

Beispiel:

Das Polynom $g(x) = x^2 + x + 1$ ist irreduzibel über $(\mathbb{Z}_2, +_2, \cdot_2)$.

Wir berechnen $01 \cdot_{2^2} 11$, wobei 01 bzw. 11 als Elemente von $\mathbb{Z}_2^2$ aufgefasst werden.

01 entspricht dem Polynom $f(x) = 1 \cdot x + 0 = x$.

11 entspricht dem Polynom $h(x) = 1 \cdot x + 1 = x + 1$.

Nun ist $f(x) \cdot h(x) = x^2 + x$.

Wir berechnen $f(x) \cdot h(x) \bmod g(x)$:

$$\begin{array}{rl} x^2 + x & : x^2 + x + 1 = 1 \\ \underline{-(x^2 + x + 1)} & \\ 1 & \end{array}$$

Es ergibt sich $f(x) \cdot h(x) \bmod g(x) = 1$, und somit $01 \cdot_{2^2} 11 = 10$

gleich dem Polynom $0 \cdot x + 1$ vom Grad 2

RS-Code: Kodierung

Der Reed-Solomon-Code

Beim $RS(q,m,n)$-Code für $m \leq n \leq q$ werden Nachrichten aus $\mathbb{F}_q^m$ zu Codewörtern aus $\mathbb{F}_q^n$ kodiert.

$u_0, \ldots, u_{n-1}$ seien paarweise verschiedene Elemente aus $\mathbb{F}_q$.

Eine Nachricht $(a_0, a_1, \ldots, a_{m-1}) \in \mathbb{F}_q^m$ wird zu

$$(p(u_0), p(u_1), \ldots, p(u_{n-1}))$$

kodiert, wobei $p(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_1 x + a_0$.

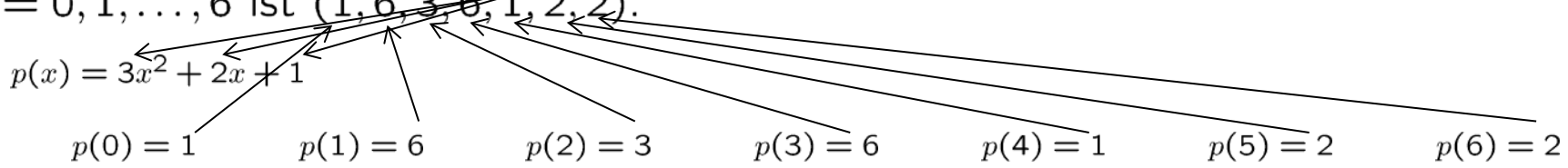
Bei der Kodierung wird die Nachricht also als Koeffizienten eines Polynoms interpretiert, und dieses Polynom an Stützstellen ausgewertet.

Die Kodierung hängt von der Wahl der (sogenannten Stützstellen) u_i ab. Die Wahl der u_i hat keinen Einfluss auf die Eigenschaften (wie etwa die Anzahl an korrigierbaren Fehlern) des Codes. Deshalb spricht man von **dem** $RS(q,m,n)$-Code.

Beispiel:

7 ist Primzahl, also $\mathbb{F}_7 = \mathbb{Z}_7$, also "mod7"

Die Kodierung der Nachricht $(1,2,3)$ mit dem $RS(7,3,7)$ Code mit $u_i = i$ für $i = 0,1,\ldots,6$ ist $(1,6,3,6,1,2,2)$.



RS-Code: Kodierung

Lernkontrolle:

Berechnen Sie die Kodierung der Nachricht $(0, 2, 3)$ mit dem $RS(7, 3, 7)$ Code mit $u_i = i$ für $i = 0, 1, \ldots, 6$.

Lösung: $(0, 5, 2, 5, 0, 1, 1)$

RS-Code: Minimaldistanz

Satz: Der $RS(q, m, n)$-Code für $m \leq n \leq q$ ist ein $(n, m, n - m + 1)$-Code-

Beweis:

Bei der Kodierung einer Nachricht (der Länge m) wird ein Polynom vom Grad $\leq m-1$ an n Stellen ausgewertet.

Wenn zwei Codewörter also an m Stellen übereinstimmen, dann stimmen die den Nachrichten entsprechenden Polynome an m Stellen überein.

Und damit sind die Polynome schon gleich, insbesondere also die Nachrichten.

Das bedeutet, dass die Codewörter zweier unterschiedlicher Nachrichten an höchstens $m-1$ Stellen gleich sein können, d.h. sie unterscheiden sich an mindestens $n-(m-1) = n - m + 1$ Stellen.

Die Minimaldistanz beträgt also mindestens $n - m + 1$.

Die Singleton-Schranke besagt, dass die Minimaldistanz höchstens $n-m+1$ beträgt.

Daraus folgt sofort, dass die Minimaldistanz $n - m + 1$ beträgt.

Der $RS(q, m, n)$-Code, als $(n, m, n - m + 1)$-Code, ermöglicht also die Korrektur von $\lfloor \frac{n-m}{2} \rfloor$ Fehlern.

Wir behandeln nun den Berlekamp-Welch-Algorithmus, der eine effiziente Art zur Dekodierung inklusive Fehlerkorrektur darstellt.

RS-Code: Korrektur von Auslöschungen

Zunächst zeigen wir noch kurz, dass es RS-Codes auch ermöglichen, eine grosse Anzahl an unleserlichen Stellen zu rekonstruieren.

Satz: Wenn beim RS-Code höchstens $n-m$ Auslöschungen stattfinden, kann die ursprüngliche Nachricht rekonstruiert werden.

Beweis: Es sind also mindestens m Stellen des empfangenen Codewortes leserlich.

Es gibt genau ein zugehöriges Interpolationspolynom vom Grad $\leq m-1$. Die Koeffizienten dieses Polynoms stellen die ursprüngliche Nachricht dar.

Beispiel:

Es wird der $RS(7,3,7)$ Code mit $u_i = i$ für $i = 0,1,\ldots,6$ verwendet. Sie empfangen $(1,?,?,6,1,2,?)$. Was ist die ursprüngliche Nachricht, wenn die empfangenen Stellen alle korrekt sind?

Lösung: $(1,2,3)$

Dies sind die Koeffizienten des Interpolationspolynoms, welches zu beliebigen 3 Stützstellen und den entsprechenden Werten, etwa $x_0 = 0, y_0 = 1$, $x_1 = 3, y_1 = 6$ und $x_2 = 4, y_2 = 1$, konstruiert wird.

Das Interpolationspolynom (vom Grad $\leq 3-1=2$) hat also die Form $p(x) = a_2x^2 + a_1x + a_0$.

Die Koeffizienten a_2, a_1 und a_0 ergeben sich aus $p(x_i) = y_i$, also aus:

$$\begin{array}{rcl} a_0 & = & 1 \\ 2a_2 + 3a_1 + a_0 & = & 6 \\ 2a_2 + 4a_1 + a_0 & = & 1 \end{array} \rightarrow \begin{array}{rcl} 2a_2 + 3a_1 & = & 5 \\ 2a_2 + 4a_1 & = & 0 \end{array} \overset{\text{(II)} \leftarrow \text{(II)}-\text{(I)}}{\rightarrow} \quad a_1 = 2 \quad \rightarrow \quad a_2 = 3$$

RS-Code: Korrektur von Auslöschungen

Wir kommen nun zum Berlekamp-Welch-Algorithmus, der es gestattet, bis zu $\lfloor \frac{n-m}{2} \rfloor$ Fehler zu korrigieren.

Bemerkung:

Man kann weniger Fehler korrigieren als Auslöschungen rekonstruieren, da es schwerer ist, Fehler zu korrigieren, da man dann eben nicht weiss, wo die Fehler passiert sind.

RS-Code: Dekodierung

Es bezeichne $(y_0, \ldots, y_{n-1})$ das empfangene Codewort, welches an $\leq \lfloor \frac{n-m}{2} \rfloor$ vom eigentlichen Codewort $(p(u_0), p(u_1), \ldots, p(u_{n-1}))$ abweicht.

Hierbei bezeichnet $p(x)$ das zur Nachricht gehörende Polynom, d.h. die Koeffizienten von $p(x)$ stellen die Nachricht dar.

Es bezeichne S die Menge des Positionen, an denen ein Fehler passiert ist.

Es sei $E(x) = \prod_{i \in S} (x - u_i)$.

Dieses Polynom (vom Grad $\leq \lfloor \frac{n-m}{2} \rfloor$) heisst Fehlerlokalisationspolynom.

Es erfüllt $E(u_i) = 0$, falls an der Position i ein Fehler passiert ist.

Für alle $i = 0, \ldots, n-1$ gilt $(y_i - p(u_i)) \cdot E(u_i) = 0$.

Falls an Position i kein Fehler passiert ist, gilt $p(u_i) = y_i$, andernfalls gilt $E(u_i) = 0$.

Für $N(x) = E(x) \cdot p(x)$, einem Polynom vom Grad $\leq \lfloor \frac{n-m}{2} \rfloor + m - 1$, gilt nun

$$N(u_i) = y_i \cdot E(u_i) \text{ für alle } i = 0, 1, \ldots, n-1.$$

Mittels dieser n linearen Gleichungen wird ein vom Nullpolynom verschiedenes Polynom $E(x)$ und ein Polynom $N(x)$ bestimmt, um dann $p(x)$ via $N(x)/E(x)$ zu berechnen.

Es gibt mehrere Lösungen für $E(x)$ und $N(x)$, der Quotient $p(x) = N(x)/E(x)$ ist aber stets eindeutig!

RS-Code: Dekodierung

Beispiel: Jeweils im Körper $(\mathbb{Z}_5, +_5, \cdot_5)$, also alles modulo 5.

Es wurde der $RS(5,2,4)$-Code mit $u_i = i$, $i = 0,\ldots,3$ verwendet (bei dem also maximal ein Fehler korrigiert werden kann). Wir erhalten $(0,0,1,4)$.

Wie lautet die entsprechende Nachricht, wenn wir davon ausgehen, dass höchstens ein Fehler passiert ist?

Wir setzen $N(x)$ als allgemeines Polynom vom Grad $\lfloor\frac{n-m}{2}\rfloor + m - 1 = 2$ an:

$$N(x) = n_2x^2 + n_1x + n_0$$

Wir setzen $E(x)$ als allgemeines Polynom vom Grad $\lfloor\frac{n-m}{2}\rfloor = 1$ an: $E(x) = e_1x + e_0$.

$N(u_i) = y_i \cdot E(u_i)$ für $0,\ldots,n-1 = 3$ entspricht:

$$\begin{aligned} n_0 &= 0 \\ n_2 + n_1 + n_0 &= 0(e_1 + e_0) \\ 2^2n_2 + 2n_1 + n_0 &= 1(2e_1 + e_0) \\ 3^2n_2 + 3n_1 + n_0 &= 4(3e_1 + e_0) \end{aligned}$$

$$\rightarrow \begin{array}{rcrcrcl} n_2 & + & n_1 & = & 0 & & \\ 4n_2 & + & 2n_1 & = & 2e1 & + & e_0 \\ 4n_2 & + & 3n_1 & = & 2e1 & + & 4e_0 \end{array} \overset{\substack{\text{(II)} \leftarrow \text{(II)}-4\cdot\text{(I)} \\ \text{(III)} \leftarrow \text{(III)}-4\cdot\text{(I)}}}{\rightarrow} \begin{array}{rcrcrcl} 3n_1 & = & 2e1 & + & e_0 & & \\ 4n_1 & = & 2e1 & + & 4e_0 & & \end{array}$$

$$\overset{\text{(II)} \leftarrow \text{(II)}-\text{(I)}}{\rightarrow} \quad n_1 = 3e_0$$

Es sei $e_0 = 1$. Dann erhalten wir $n_1 = 3$, $e_1 = 4$ und $n_2 = 2$.

Wir erhalten also als (eine) Lösung $N(x) = 2x^2 + 3x$ und $E(x) = 4x + 1$.

Mittels Polynomdivision erhalten wir $p(x) = N(x)/E(x) = 3x = 3x + 0$.

Die Nachricht war also $(0,3)$.

RS-Code: Bemerkungen

Abschliessende Bermerkungen zu RS-Codes

(1) In der Praxis wird oft ein Körper mit 2^8 Elementen verwendet. Damit können also Bytes kodiert werden. Beim Dekodieren wird dann also ein fehlerhaftes Byte durch ein korrektes ersetzt. (Dabei ist es egal, ob nur ein Bit im Byte falsch ist, oder alle.)

(2) Bei CDs etc. passieren Fehler oft in ganzen "Bündeln". Dafür sind RS-Codes besonders gut geeignet (da es egal ist, ob ein Bit oder alle Bits eines Bytes fehlerhaft sind).

(3) Bei der vorgestellten Variante kann man die Nachricht, selbst bei fehlerfreier Übertragung, nicht aus dem Codewort ablesen. Es gibt eine Variante, wo das Codewort dadurch ensteht, dass man der ursprünglichen Nachricht zusätzliche Stellen hinzufügt, das Codewort also mit der Nachricht beginnt.

(4) Man kann mit dem $RS(q, n, m)$ auch Kombinationen von Auslöschungen und Fehlern korrigieren. Wenn α die Anzahl Auslöschungen und β die Anzahl Fehler bezeichnet, muss dazu $\alpha + 2\beta \leq n - m + 1$ gelten. Wir können darauf hier nicht eingehen.

(5) Beim Dekodieren muss im Wesentlichen ein Gleichungssystem gelöst werden, und dann eine Polynomdivision gemacht werden. Beides kann algorithmisch effizient umgesetzt werden.