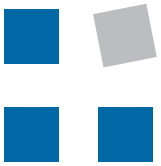
Zusammenfassung ICTh

Informations- und Codierungstheorie

Emanuel Duss
emanuel.duss@gmail.com

18. Januar 2014



HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Zusammenfassung ICTh
Informations- und Codierungstheorie

Dieses Dokument basiert auf der Vorlesung „Informations- und Codierungstheorie“ der HSR (Hochschule für Technik Rapperswil) vom HS 2013.

Revision 23a753d vom 2013-12-11.

MITMACHEN
Falls Du an diesem Dokument mitarbeiten willst, kannst Du das Dokument auf GitHub unter http://github.com/mindfuckup/HSR_ICTh_Zusammenfassung forken.

MITWIRKENDE
Folgende Personen haben an diesem Dokument mitgewirkt:
Emanuel Duss (eduss@hsr.ch)

LIZENZ

Inhaltsverzeichnis

1 Einführung

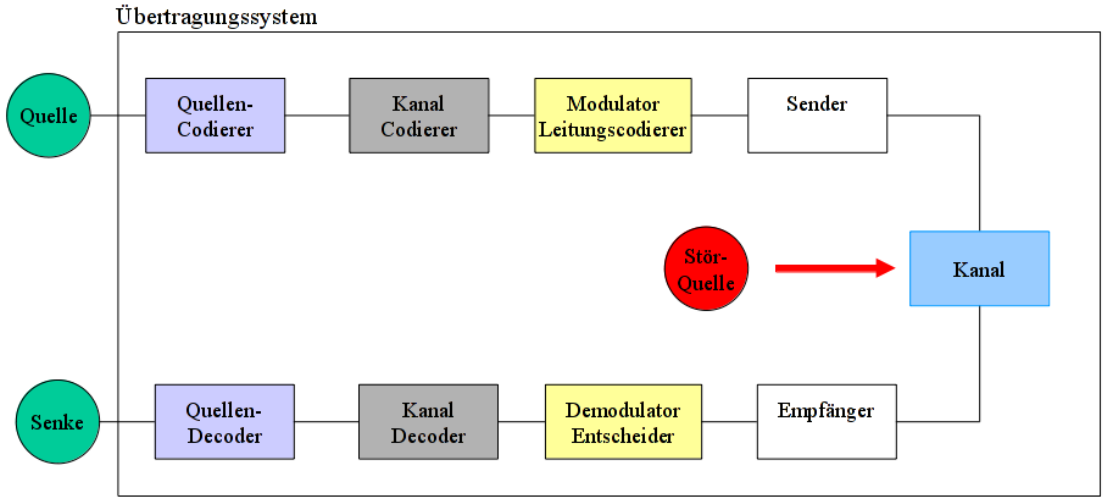


Abbildung 1: Modell der Informationsverarbeitung

Kanal Stellt das Medium dar, das zur Übertragung der Nachrichtensignale verwendet wird. Beispiel sind: Doppeldrahtader, Glassfaser, Radiowellen Der Kanal kann durch seine Eigenschaften abstrahiert werden. Eigenschaften sind z.B: Fehlerrate, Dämpfung, Latenz: Diese Eigenschaften werden zusätzlich überlagert durch zusätzliche Störquellen.

Sender Setzt das Nachrichtensignal auf dem Medium ab. Typische Parameter: Sendeleistung, Spannungspegel.

Empfänger Nimmt das Signal vom Medium ab.

Modulator Moduliert bzw. überträgt die Information auf ein geeignetes Trägersignal.

Demodulator Gewinnt aus dem empfangen Signal die ursprünglich Information zurück. Hier können Fehler in der Rückgewinnung auftreten.

Kanal Coder Stellt codes und Verfahren bereit um Übertragungsfehler zu erkennen und zu beheben. Beispiele sind Blockcodes, zyklische Codes oder Faltungscodes. Dazu muss der der Kanalcodierer mit dem Kanaldecoder zusammenarbeiten.

Quellen Codierer: Bearbeitet die Zeichen bzw. Symbolfolge der Quelle, um diese z.B. zu komprimieren oder zu verschlüsseln. Wie bei der Kanalcodierung müssen Quellen deoder und Coder auf einander abgestimmt sein.

Quelle/Senke Stellen einen gemeinsamen Zeichenvorrat bereit. Die Eigenschaften einer Quelle können durch stochastisch Methoden beschriebne werden. Grundlage für alle weiteren Überlegungen liefert die Wahrscheinlichkeitsrechnung.

2 Signale: Träger der Information und einfach Leitungscodes

2.1 ASCII-Tabelle zur Codierung

Beispiel: `A` = `0x41` oder `a` = `0x61`

	2	3	4	5	6	7
0:		0	@	P	‘	p
1:	!	1	A	Q	a	q
2:	"	2	B	R	b	r
3:	#	3	C	S	c	s
4:	$	4	D	T	d	t
5:	%	5	E	U	e	u
6:	&	6	F	V	f	v
7:	’	7	G	W	g	w
8:	(	8	H	X	h	x
9:	)	9	I	Y	i	y
A:	*	:	J	Z	j	z
B:	+	;	K	[	k	{
C:	,	<	L	\	l	|
D:	-	=	M	]	m	}
E:	.	>	N	^	n	~
F:	/	?	O	_	o	DEL

	30	40	50	60	70	80	90	100	110	120
0:		(	2	<	F	P	Z	d	n	x
1:		)	3	=	G	Q	[	e	o	y
2:		*	4	>	H	R	\	f	p	z
3:	!	+	5	?	I	S	]	g	q	{
4:	"	,	6	@	J	T	^	h	r	|
5:	#	-	7	A	K	U	_	i	s	}
6:	$	.	8	B	L	V	‘	j	t	~
7:	%	/	9	C	M	W	a	k	u	DEL
8:	&	0	:	D	N	X	b	l	v	
9:	’	1	;	E	O	Y	c	m	w	

`Quelle: Manpage ascii(7)`

2.2 Kenngrössen eines Signals

2.2.1 Sinus und Cosinus

- Grad = Bogenmass $\cdot 180/\pi$
- Bogenmass = Grad $\cdot \pi/180$
- $sin(x) = \frac{2\pi}{T} \cdot t$
- Frequenz $= f = \frac{1}{T} = \sin(2\pi f)t$
- Kreisfrequenz $= \omega = 2\pi f$

2.3 Bell

Bel ist ein Verhältnismass.

$$1\text{Bel} = 10\text{dB} = \log_{10} \frac{s_{out}}{s_{in}}$$

Bel kann auch für absolute Werte mit folgenden Bezugsgrössen verwendet werden. Eine Verdoppelung des Signals entspricht +3dBDämpfung von 3dB = Verstärkung von -3dB dBm: Leistungspegel mit der Bezugsgrösse 1mW

2.4 Leitungscodierung

- Ist 1 = High, 0 = Low, wird es bei langen Folgen schwierig, das Codewort wiederherzustellen
- Unterstützung der Takt- und Phasenrückgewinnung im Empfäger
- Vermeidung von Gleichstromkomponenten
- Optimierung des Bandbreitenbedarfs
- Unempfindlichkeit gegenüber Störungen

2.4.1 Eigenschaften

Unipolares Signal Nur positive Spannungen Nachteil: Hoher Gleichstromanteil

Bipolares Signal Negative sowie positive Spannungen Vorteil: Geringen Gleichstromanteil, Problem sind lange 1 und 0 Folgen

RZ-Impuls Return to Zero:

NRZ Non Retorn to Zero:

$$f(x) = \sum_{n=0}^{\infty} a_n \cos(nx) + b_n \sin(nx)$$

2.5 Fast Fourier Transformation

Vorgehensweise Fourier-Analyse mit Tabellen.

3 Sprache Transponieren, Modulieren und Multiplexen

3.1 Transponieren

3.2 Modulieren

3.3 Multiplexen

4 Sprache digitalisieren und übertragen

5 Leitungscodes und Multiplextechniken

6 Informationstheorie Grundlagen

Zeichenvorrat Die Quelle hat eine endliche Menge von Zeichen. Die Quelle kann aber unendlich viele Zeichen aus dem Zeichenvorrat generieren. Die Senke muss über den gleichen Zeichenvorrat verfügen.

$$X = \{a, b, c, d\} \text{ mit } x_1 = a \text{ und Anzahl } N = 4$$

Entscheidungsgehalt

$$H_0 = \log_2(N) \qquad \text{in [Bit]}$$

Entscheidungsfluss Dauer für die Übertragung eines Zeichens = τ

$$H_0^* = \frac{\log_2(N)}{\tau} \qquad \text{in [Bit/s]}$$

Auftrittswahrscheinlichkeit

$$P(x_i) = \frac{\text{Vorkommen}}{\text{Total}}$$

Informationsgehalt Der Informationsgehalt ist umgekehrt proportional zur Auftrittswahrscheinliichkeit. Ein Zeichen, welches selten vorkommt, hat ein hoher Informationsgehalt.

$$I(x_k) = \log_2\left(\frac{1}{P(x_k)}\right) \qquad \text{in [Bit]}$$

Entropie Mittlerer Informationsgehalt der Quelle:

$$H(X) = \sum_{k=1}^{N} p(x_k) I(x_k) \qquad \text{in [Bit / Zeichen]}$$

- $H(X)$ ist maximal, wenn alle Zeichen gleich wahrscheinlich sind.
- Je kleiner $H(X)$, umso grösser die Redundanz und umso komprimierbarer.

Mittlere Codewortlänge

$$\text{Mittlere Codewortlänge} = L = \sum_{k=1}^{N} P(x_k) \cdot \text{length}_2(x_k)$$

- ASCII = 8

Redundanz der Quelle [Bit / Zeichen] Kommt nur ein einziges Zeichen vor, ist es redundant, da es sicher vorhersagbar ist und es somit keine Information enthält.

$$R_Q = H_0 - H(X) \qquad \text{in [Bit / Zeichen]}$$

Redundanz des Codes

$$R_c = L - H(X) \qquad \text{in [Bit / Zeichen]}$$

7 Blockcodes und zyklische Codes

7.1 Einführung

- Leichte Implementierung durch Schieberegister
- Hohe Fehlererkennbarkeit
- Blockbildung nötig $\Leftrightarrow$ keine fortlaufende Kodierung eines Datenstroms möglich

Generatorpolynom

$$G(x) = x^4 + x + 1$$

Anzahl erkennbare Fehler

$$e^* = h - 1$$

Kontrollstellen

$$k = \text{Grad des höchsten Polynoms}$$

Anzahl sicher korrigierbaren Fehler

$$h = \begin{cases} 2e+2 & h \text{ gerade} \\ 2e+1 & h \text{ ungerade} \end{cases} \qquad \Rightarrow e = \begin{cases} \frac{h-2}{2} & h \text{ gerade} \\ \frac{h-1}{2} & h \text{ ungerade} \end{cases}$$

Kontrollstellen

$$k$$

Hammingdistanz Mindestabstand zwischen zwei Codewörtern.

$$h = \min_{i,j}(d(x_i, x_j))$$

Abrahamson: $h = 4$, Golois $h = 7$, Hamming $h = 3$

Codewortlänge

$$n = \begin{cases} 2^k - 1 & \text{Hamming} \\ 2^{k-1} - 1 & \text{Abramson} \end{cases}$$

Nachrichtenlänge

$$m = n - k$$

Dichtgepackter Code Der Coderaum ist Duchtgepackt, wenn sich alle Codewörter (gültig und ungültig) in einer Korrigierkugel befinden.

$$\underbrace{2^m}_{Foo} \qquad \underbrace{\sum_{w=0}^{e} \binom{n}{w}}_{AnzahlCW\,pro\,Korrigierkugel} \qquad \leq \qquad \underbrace{2^n}_{Anzahl\,aller\,CW}$$

- n Dimension des Codes (Anzahl aller Codewörter = 2^n)
- m Dimension der Nachricht (Anzahl aller Gültiger Codewörter = 2^m
- k Dimension der Kontrollstellen mit $n = m + k$

7.2 Zyklische Codes

Ein Polynom ist irreduzibel, wenn es einen Rest gibt, wenn man es durch $(x + 1)$ teilt.

8 Faltungscodes

8.1 Einführung

- Fortlaufende Codierung eines Datenstroms möglich $\Leftrightarrow$ keine Blockbildung nötig
- Für die Decodierung ist keine Blocksynchronisation nötig
- Gute Faltungscodes werden durch Rechnersimulation gefunden

9 Quellencodierung und Komprimierung

9.1 Diskrete Quelle mit Gedächtnis

Markoff-Diagramm

x	P(x)
A	1/3
B	16/27
C	2/27

9.2 Diskrete Quelle ohne Gedächtnis

9.3 Huffmann-Codierung

- Binärbaum von den Blättern zur Wurzel bilden
- Zeichen nach Wahrscheinlichkeit ordnen
- Beide Zeichen mit der kleinsten Auftrittswahrscheinlichkeit haben die gleich Codewortlänge L_N
- Sei L_N die mittlere Codewortlänge für eine Quelle mit N Zeichen und L_N, die mittlere Codewortlänge für den FAll, dass die beiden letzten zu einem einzigen Zeichen zusammengefasst werden, dann gilt:

9.4 Lempel-Ziv

- Gut, wenn der zu komprimierende Code wiederkehrende Muster aufweist
- Kann zur Laufzeit gemacht werden ???

10 Quellencodierung und Verschlüsslungsverfahren

10.1 Symmetrisches Verschlüsselungsverfahren

10.1.1 Schlüssel

Anzahl symmetrische Schlüssel für n Teilnehmer:

$$\frac{n(n-1)}{2}$$

Jeder Teilnehmer muss $n-1$ Schlüssel speichern.

10.1.2 Caesar-Verschlüsselung

$$\begin{aligned}\text{Encrypt}_K(P) &= (P+K) \mod 26\\ \text{Decrypt}_K(C) &= (C-K) \mod 26\end{aligned}$$

10.2 Asymmetrisches Verschlüsselungsverfahren

10.2.1 Schlüssel

10.2.2 Begriffe

Inverse Zahlen Für zwei teilerfremde Zahlen a und b existiert eine Zahl c, für die gilt:

$$a \cdot c \mod b = 1$$

Eulerfunktion Gibt die Anzahl der zu einer Zahl n teilerfremden Zahlen an. Das heisst die Zahlenpaare (n, x) mit $x < n$, die keinen gemeinsamen Teiler haben.

- $\Phi(n) =$ Anzahl der relativ primen Zahlen
- Für Primzahlen gilt: $\Phi(p) = p - 1$
- Für das Produkt zweier Primzahlen p und q gilt: $\Phi(pq) = (p-1)(q-1)$

Satz von Euler Für zwei teilerfremde Zahlen a und b gilt:

$$a^{\Phi(b)} \mod b = 1$$

Ist b das Produkt zweier Primzahlen, gilt:

$$a^{\Phi(pq)} \mod pq = 1 \qquad \Rightarrow \qquad a^{(p-1)(q-1)} \mod pq = 1$$

Anwendung

$$a^y \mod pq = a^{y \mod \Phi(pq)} \mod pq$$

Mit setzen von $y = ed$ mit d als inverses Element folgt:

$$y \mod \Phi(n) = ed \mod \Phi(n) = 1$$

Mit e als Schlüsseln zum Verschlüsseln (encrypt) und d zum Entschlüsseln (decrypt).

10.3 RSA

10.3.1 Schlüsselpaar generieren

Primzahl Bestimme zwei sehr grosse Primzahlen p und q mit $p \neq q$.

RSA-Modul Bilde das RSA-Modul $N = pq$

Eulersche Φ-Funktion Bestimme $\Phi(N) = \Phi(qp) = \Phi(p) \cdot \Phi(q) = (p-1)(q-1)$

Öffentlicher Schlüssel Wähle eine Zahl e, für die gilt:

$$1 < e < \Phi(N) \qquad \text{und} \qquad \text{ggT}(e, \Phi(N)) = 1$$

Öffentlicher Schlüssel = (e, N)

Privater Schlüssel Berechne die Inverse d zu e mit dem erweiterten Euklidschen Algorithmus. ($ed = 1 \mod \Phi(N)$)

$$e \cdot d + k \cdot \Phi(N) = 1 = \text{ggT}(e, \Phi(N))$$

Privater Schlüssel = (d, N). k wird nicht mehr benötigt.

10.3.2 Ver- und Entschlüsseln

Verschlüsseln Verschlüsseln der Nachricht D: Encrypted $C = D^e \mod N$

Entschlüssel Entschlüsseln der Nachricht C: Decrypted $D = C^d \mod N$

10.3.3 Beispiel

Schlüsselpaar generieren

- Wähle Primzahl $p = 7$ und $q = 11$
- RSA-Modul $N = pq = 7 \cdot 11 = 77$
- Bestimme $\Phi(N) = (p-1)(q-1) = 6 \cdot 10 = 60$

- Wähle e mit $1 < e < 60 \wedge \text{ggT}(e, 60) = 1 \Rightarrow e = 47$
  - Öffentlicher Schlsel = $(e, N) = (47, 77)$
- Erweiterter Euklidscher Algorithmus
  - Privater Schlüssel = $(d, N) = (23, 77)$

Text D Verschlüsseln Mit $D = 2$

- $C = D^e \mod N = 2^{47} \mod 77 = 18$

Text C Entschlüsseln Mit $C = 18$

- $D = C^d \mod N = 18^{23} \mod 77 = 2$

10.3.4 Beispiel mit `sage`

Aus 29C3 Talk "Facthacks": `29c3-5275-en-facthacks_h264.mp4`. Mehr Infos: `http://facthacks.cr.yp.to/`

```
sage: # Bestimme Primzahlen p und q
sage: p = random_prime(2^5) # = 7
sage: q = random_prime(2^5  # = 11

sage: # Bestimme RSA-Modul N = p*q
sage: N=p*q # = 77

sage: # Public Key = (e, N) = (47, 77)
sage: e = 47 # Oder andere die gültig sind

sage: # Private Key = (d, N) = (23, 77)
sage: d = inverse_mod(e, (p-1)*(q-1)) # = 23

sage: # Nachricht = 2
sage: D = 2

sage: # Nachricht D Verschlüsseln
sage: C = pow(D,e)%N

sage: # Nachricht C Entschlüsseln
sage: D = pow(C,d)%N
sage: D
sage: 2
sage: # Yay! \o/
```