**Zusammenfassung**

# Informations und Codierungstheorie

Julian Klaiber und Severin Dellsperger

Hochschule für Technik Rapperswil

23. Januar 2019

# Inhaltsverzeichnis

# 1 Teil Steffen

## 1.1 Umrechnungen

| Zahl | Potenz | Name | Kürzel |
|---:|---|---|---|
| 1 | $10^0$ | Eins | |
| 10 | $10^1$ | Deka | da |
| 100 | $10^2$ | Hekto | h |
| 1 000 | $10^3$ | Kilo | k |
| 10 000 | $10^4$ | | |
| 100 000 | $10^5$ | | |
| 1 000 000 | $10^6$ | Mega | M |
| 10 000 000 | $10^7$ | | |
| 100 000 000 | $10^8$ | | |
| 1 000 000 000 | $10^9$ | Giga | G |
| 10 000 000 000 | $10^{10}$ | | |
| 100 000 000 000 | $10^{11}$ | | |
| 1 000 000 000 000 | $10^{12}$ | Tera | T |
| 10 000 000 000 000 | $10^{13}$ | | |
| 100 000 000 000 000 | $10^{14}$ | | |
| 1 000 000 000 000 000 | $10^{15}$ | Peta | P |
| 10 000 000 000 000 000 | $10^{16}$ | | |
| 100 000 000 000 000 000 | $10^{17}$ | | |
| 1 000 000 000 000 000 000 | $10^{18}$ | Exa | E |

Abbildung 1: Zehnerpotenzen Tabelle

| Dezimal | Mit Präfix | Potenz in s |
|---|---|---|
| 0,000 000 000 000 001 s | 1 fs | $10^{-15}$ |
| 0,000 000 000 000 01 s | 10 fs | $10^{-14}$ |
| 0,000 000 000 000 1 s | 100 fs | $10^{-13}$ |
| 0,000 000 000 001 s | 1 ps | $10^{-12}$ |
| 0,000 000 000 01 s | 10 ps | $10^{-11}$ |
| 0,000 000 000 1 s | 100 ps | $10^{-10}$ |
| 0,000 000 001 s | 1 ns | $10^{-9}$ |
| 0,000 000 01 s | 10 ns | $10^{-8}$ |
| 0,000 000 1 s | 100 ns | $10^{-7}$ |
| 0,000 001 s | 1 µs | $10^{-6}$ |
| 0,000 01 s | 10 µs | $10^{-5}$ |
| 0,000 1 s | 100 µs | $10^{-4}$ |
| 0,001 s | 1 ms | $10^{-3}$ |
| 0,01 s | 1 cs | $10^{-2}$ |
| 0,1 s | 1 ds | $10^{-1}$ |
| 1 s | 1 s | $10^0$ |

Abbildung 2: Zeiteinheiten

## 1.2 Signal-to-noise Ratio und Pegelplan

### 1.2.1 Thermische Rauschleistung

**Formel:**

$N[dBm] = -174dBm + 10log_{10}(\Delta f)$
$\Delta f = Frequenzintervall[Hz]$
**Beispiel:**
Systembandbreite = 4GHz $n = -174dBm + 10log_{10}(4 * 10^9) = -78dBm$

### 1.2.2 Pegelplan

**SNR Berechnung:** Der SNR für den Pegelplan berechnet sich wie folgt **Thermische Rauschleistung + Abstand (aus der Aufgabenstellung)**

**Signaldynamik:** $S_{max} - S_{min}$ in dB

| Symbol | Element | Dämpfung $\tilde{A}$ [dB] | Verstärkung $\tilde{G}$ [dB] | Zusatzinformation |
|---|---|---|---|---|
| | **Tiefpassfilter** low pass filter | **$\tilde{A}$ [dB]** | -$\tilde{A}$ [dB] | - Eckfrequenz,<br>- Sperrdämpfung |
| | **Hochpassfilter** high pass filter | **$\tilde{A}$ [dB]** | -$\tilde{A}$ [dB] | - Eckfrequenz,<br>- Sperrdämpfung |
| | **Bandpassfilter** band pass filter | **$\tilde{A}$ [dB]** | -$\tilde{A}$ [dB] | - Mittenfrequenz,<br>- Durchlassbandbreite,<br>- Sperrdämpfung |
| | **Verstärker** amplifier | -$\tilde{G}$ [dB] | **$\tilde{G}$ [dB]** | - Max. Ausgangspegel,<br>- Intercept-Punkte<br>- Rauschzahl |
| | **Abschwächer** attenuator | **$\tilde{A}$ [dB]** | -$\tilde{A}$ [dB] | - Max. Verlustleistung |
| | **Mischer** mixer | **$\tilde{A}$ [dB]** | -$\tilde{A}$ [dB] | - Max. Eingangspegel,<br>- Optimaler LO-Pegel,<br>- Rauschzahl |

Abbildung 3: Pegelplan Elemente

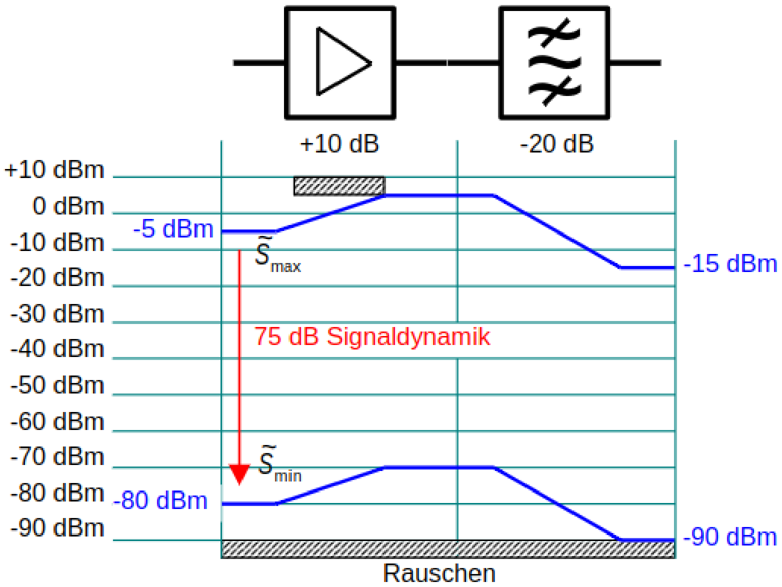



Abbildung 4: Beispiel Pegelplan

## 1.3 Abtastung von Signalen

**Aliasing** tritt auf, wenn im abzutastenden Signal Frequenzanteile vorkommen, die höher sind als die halbe Abtastfrequenz. Signal $> 1/2 * Abtastfrequenz$. Durch vorschalten eines Tiefpassfilters mit einer Grenzfrequenz $f_g < f_s/2$, kann Aliasing vermieden werden.
**Frage:** Was bewirkt die gezielte Wahl der Smapling Frequenz $f_s = f_0$? **Antwort:** Das Sampling mit der Trägerfrequenz $f_0$ bewirkt eine Verschiebung des Datensignals in das Basisband und damit eine Produktdemodulation mit $f_0$.

## 1.4 Dauer und Bandbreite von Einzelpulsen

### 1.4.1 Vorgehen Amplitudendichte

**Fragestellung:** Wie kann S(0), d.h. die Amplitudendichte bei der Frequenz f=0 Hz, einfach aus dem Verlauf des Pulses berechnet werden? Geben Sie die Formel für S(0), sowie den numerischen Wert in [V/Hz] an.

**Lösung:** $S(0) = \int_{-\infty}^{\infty} s(t)dt = \int_{-T}^{T} s(t)dt = AT$

d.h. die Gesamtfläche unter der Dreiecksfunktion s(t).

**Hinweis:**
Bei Reckteckssignalen wäre es $2 * AT$
Bei ms führt es zu mV/Hz

### 1.4.2 Energie berechnen

$E = \int_{-\infty}^{\infty} \frac{s^2(t)}{R} dt = \frac{1}{R} \int_{-\infty}^{\infty} s^2(t)dt = \frac{3}{4} * \frac{A^2 T}{R}$

Achtung: 3/4 und T ist modular

**Hinweis:** Resultat wenn ms dann mWs (Beispiel: 3/4mWs = 0.75mJ)

### 1.4.3 Dauer eines Pulses berechnen

**Formel:** $E = \frac{A^2 \tau}{R} = \frac{3}{4} * \frac{A^2 T}{R}$
**nach $\tau$ auflösen $\tau$ in ms oder s**

Achtung: 3/4 und T ist modular

### 1.4.4 Bandbreite

**Formel:** $E = \frac{|S(0)|^2 * B}{R} = \frac{A^2 T^2 B}{R} = \frac{3}{4} * \frac{A^2 T}{R}$
$B = \frac{3}{4} * \frac{1}{T}$ und damit $B = \frac{3}{4} kHz = 0.75 kHz$

Achtung: 3/4 und T ist modular

### 1.4.5 Zeit-Bandbreitenprodukt

Wie gross ist das Zeit-Bandbreitenprodukt B$\tau$?

**Formel:**B$\tau = \frac{3}{4} * \frac{1}{T} * \frac{3}{4} * T$

Achtung: 3/4 und T ist modular
**Hinweis:** kHz und ms lösen sich auf = keine Masseinheit

## 1.5 Leitungscodes

| Leitungscode | DC-Freiheit | | Taktinformation | |
|---|---|---|---|---|
| | Ja | Nein | Ja | Nein |
| Bipolarer NRZ Code | | X | | X |
| Unipolarer NRZ Code | X | | | X |
| Unipolarer NRZ Mark Code | * | * | | X |
| Bipolarer Manchester-Code | X | | X | |
| Unipolarer Manchester-Code | | X | X | |
| Bipolarer AMI Code | X | | | X |
| Unipolarer RZ Code | X | | | X |

Tabelle 1: Nur Nullstellen

Tabelle 2: *=Nicht entscheidbar kommt auf vorheriges Zeichen an.

| Leitungscode | DC-Freiheit | | Taktinformation | |
|---|---|---|---|---|
| | Ja | Nein | Ja | Nein |
| Bipolarer NRZ Code | | X | | X |
| Unipolarer NRZ Code | | X | | X |
| Unipolarer NRZ Mark Code | X | | X | |
| Bipolarer Manchester-Code | X | | X | |
| Unipolarer Manchester-Code | X | | X | |
| Bipolarer AMI Code | X | | X | |
| Unipolarer RZ Code | | X | X | |

Tabelle 3: Nur Einsstellen

## 1.6 Modulationsarten

### 1.6.1 Beispiel

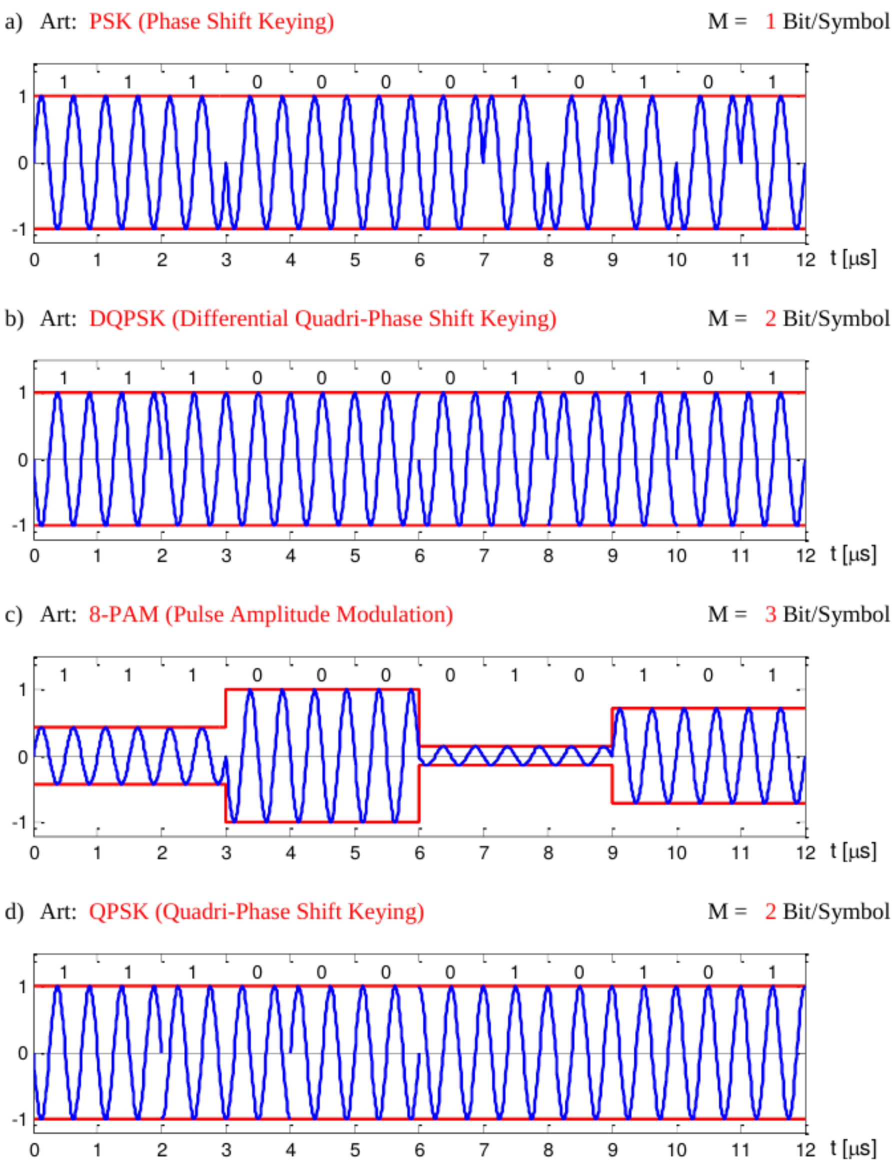



Abbildung 5: Modulationsarten Beispiel

## 1.7 Tonhöhenverschiebung von Audiosignalen

### 1.7.1 Mickey Mouse

**Lösung 1:**

- Elimination des oberen Seitenbandes (USB) durch ein Tiefpassfilter mit der Grendfrequenz (diese auslesen was muss wohin geschoben werden)
- Demodulation mit einer LO-Frequenz von X (Auslesen), welches das untere Seitenband (LSB) mit einem Frequenzshift von X in das Basisband zurückschiebt

**Lösung 2:**

- Elimination des unteren Seitenbandes (LSB) durch ein Hochpassfilter mit der Grenzfrequenz X (auslesen)
- Demodulation mit einer LO-Frequenz von X (auslesen), welches das obere Seitenband (USB) mit einem Frequenzshift von X in das Basisband zurückschiebt.

**Massnahmen nach der Verschiebung:**

- Durch die Demodulation ensteht eine Spektrumskomponente bei der doppelten Trägerfrequenz von ca. 16kHz
- Die hörbaren Frequenzanteile können mit einem Tiefpassfilter eliminiert werden.

# 2 Teil Meili

## 2.1 Entscheidungsgehalt

Mass für den Aufwand der zur Bildung einer Nachricht bzw. für die Entscheidung einer Nachricht notwendig ist.

$H_0 = log_2(N)[bit]$

## 2.2 Entscheidungsfluss

$H_0^* = \frac{log_2(N)}{\tau}[\frac{bit}{s}]$

wobei $\tau$ die Zeit zur Übertragung eines Quellzeichens.

## 2.3 Ergebnis und Ergebnismenge

**Definition:** Die Menge aller möglichen Ausgänge eines Zufallsvorgangs heisst **Ergebnismenge** und wird mit $\omega$ bezeichnet. Ein einzelnes Element heisst Ergebnis. Wir notieren die Anzahl aller Elemente von $\omega$ d.h. die Anzahl aller Ergebnisse mit $|\omega|$

## 2.4 Informationsgehalt

**Definition:** Der Informationsgehalt eines Zeichens sagt aus, wie viele Elementarentscheidungen zur Bestimmung dieses Zeichens zu treffen sind.

$I(x_k) = log_2(\frac{1}{p(x_k)})[bit]$

**Taschenrechner:** icth\i_info(x)

## 2.5 Entropie

**Definition:** Die Entropie bezeichnet den mittleren Informationsgehalt der Quelle. Sie zeigt also auf, wie viele Elementarentscheidungen die Quelle/Senke im Mittel pro Zeichen treffen muss.

$H(X) = \sum_{k=1}^{N} p(x_k) * I(x_k) = \sum_{k=1}^{N} p(x_k) * log_2(\frac{1}{p(x_k)})[bit/Zeichen]$

Durchschnittliche Anzahl Entscheidungen die von der Quelle getrofen werden müssen.

**Taschenrechner:** icth\h_entropie(Wahrscheinlichkeiten)

## 2.6 Redundanz

Der mittlere Informationsgehalt, die Entropie, einer Quelle/Senke wird maximal wenn beide Zeichen gleich oft vorkommen.

Je kleiner die Entropie desto grösser die Redundanz. Wenn alle Zeichen gleich wahrscheinlich sind ist die Entropie maximal und die Redundanz = 0.

$R_Q = H_0 - H(X)[bit/Zeichen]$

## 2.7 Kanalmodell

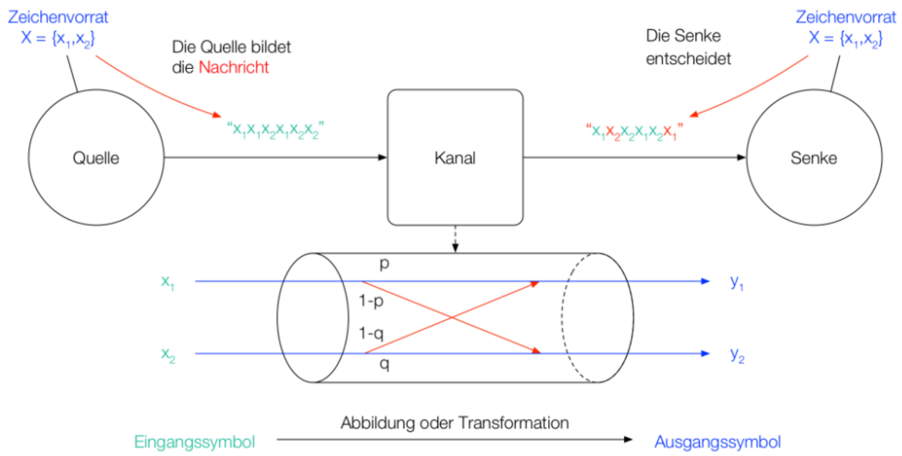



Abbildung 6: Kanalmodell

### 2.7.1 Kanalmatrix

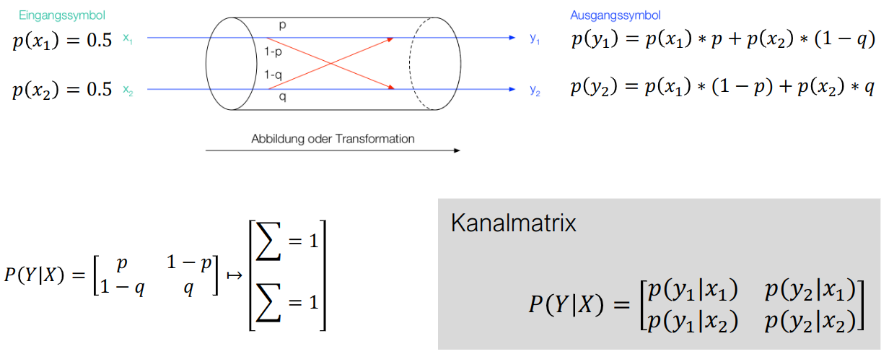



Abbildung 7: Kanalmatrix

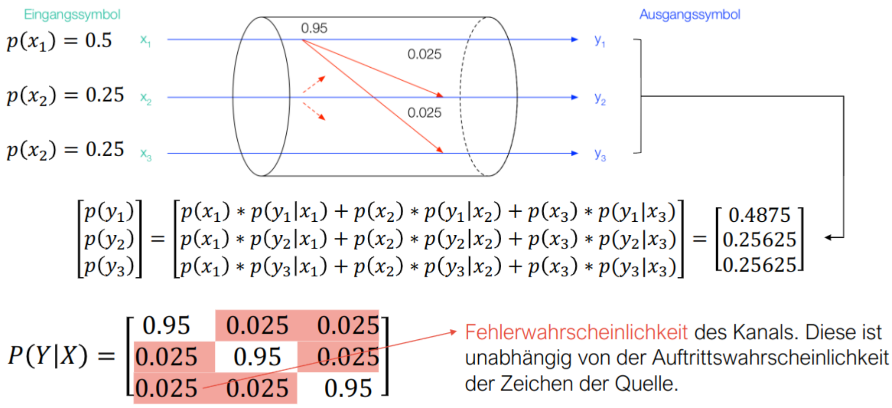
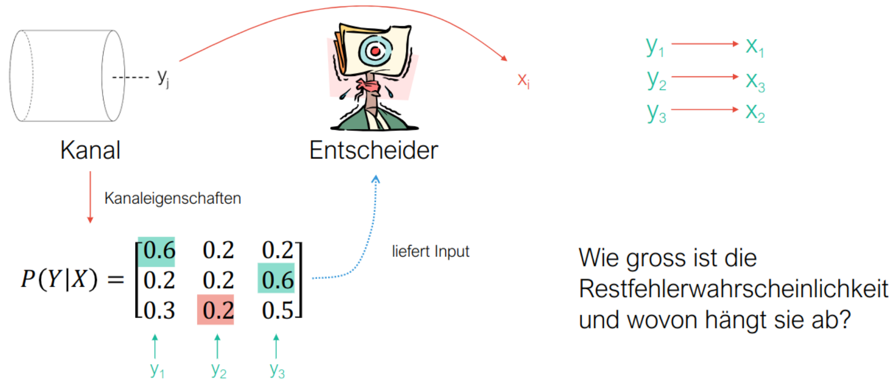



Abbildung 8: Kanalmatrix Beispiel

**Frage:** Wie kann die Kanalmatrix eines Kanals praktisch ermittelt werden?
**Lösung:** Viele Messungen durchführen und aus diesen die Häufigkeiten berrechnen. Daraus kann dann die Kanalmatrix erstellt werden.

### 2.7.2 Maximum-Likelihood Verfahren



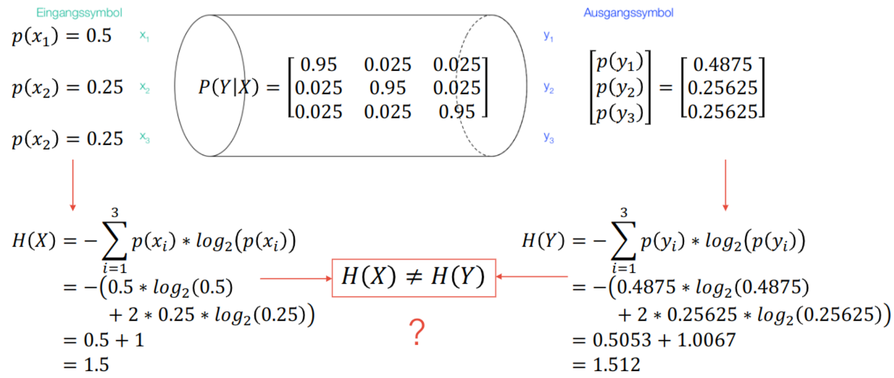

Maximum-Likelihood Verfahren

**Restfehlerwahrscheinlichkeit:**

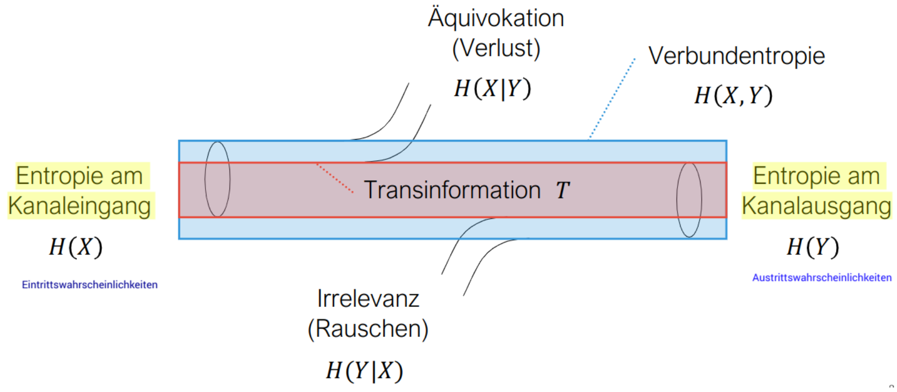
$p(keineFehler) = 0.6 * p(x_1) + 0.2 * p(x_3) + 0.6 * p(x_2)$
$p(Restfehlerwahrscheinlichkeit) = 1 - p(keineFehler)$

**Taschenrechner:** icth\res_ err_ prop

### 2.7.3 Transinformation




Transinformation




Transinformation am Kanal

### 2.7.4 Transinformation Berechnung

$T = H(X) - H(X|Y)[bit/Zeichen]$

$T = H(Y) - H(Y|X)[bit/Zeichen]$

### 2.7.5 Äquivokation (Verlust)

$H(X|Y) = -\sum_j^n \sum_j^n p(x_j, y_j) * log_2(p(x_j|y_j))$

- Auch Rückschlussentropie genannt
- Ungewissheit über das gesendete Zeichen bei bekanntem Empfangszeichen
- Merke: Ist der Kanal fehlerfrei, so ist die Äquivokation gleich 0

### 2.7.6 Irrelevanz (Rauschen)

$H(X|Y) = -\sum_j^n \sum_j^n p(x_j, y_j) * log_2(p(y_j|x_j))$

- Auch Streuentropie genannt
- Ungewissheit der empfangenen Zeichen bei vorgegeben Sendezeichen

**Taschenrechner:** icth\float h_ yx

### 2.7.7 Verbundentropie

$H(X|Y) = -\sum_j^n \sum_j^n p(x_j, y_j) * log_2(p(x_j, y_j))$

Der mittlere Informationsgehalt über alle Zeichen (bestehend aus einem Zeichen der Quelle und einem Zeichen der Senke)

$p(x, y) = p(x) * p(y|x)$

### 2.7.8 Bedingte Entropie

$H(Y|X) = H(X, Y) - H(X)$

### 2.7.9 Symbolrate Rmax

$T * Bandbreite$

## 2.8 Blockcodes

### 2.8.1 Nachrichtenzahl

$m = 2^k - k - 1$

### 2.8.2 Hamming Distanz

**Hammingdistanz h=** Kürzeste Distanz (Änderungen) von einem gültigen Codewort zum nächsten gültigen Codewort.

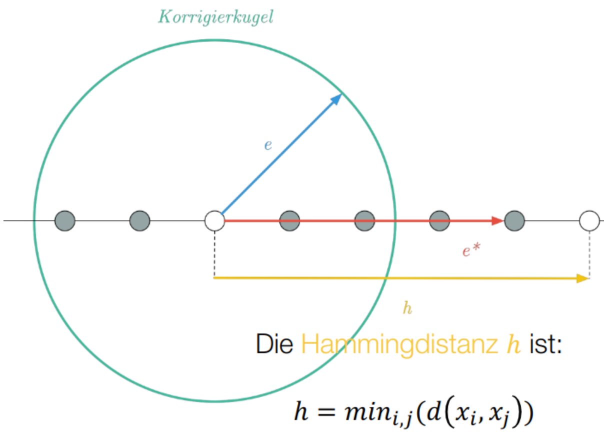



Abbildung 9: Hammingdistanz

### 2.8.3 Kontrollstellen für Codewort ermitteln

- Codewort über Matrix schreiben
- Kontrollieren ob 1 und 1 matched (markieren)
- Auf jeder Zeile alle matched 1 zusammenzählen mod 2 rechnen

$$\begin{array}{c} \begin{matrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & & & & \end{matrix} \\ \left[\begin{matrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{matrix}\right] \end{array} \begin{array}{l} x_1 + x_3 + x_4 \mod 2 = 1 \\ x_1 + x_3 + x_{10} \mod 2 = 1 \\ x_1 + x_4 \mod 2 = 0 \\ x_1 + x_3 + x_4 + x_{10} \mod 2 = 0 \end{array}$$

Abbildung 10: Codewort ermitteln Beispiel

## 2.9 Hamming Codeblock aus Messwerten

**Aufgabe 7.**
*Hamming Blockcode 2:* Sie haben die Aufgabe, die Übertragung von Messwerten abzusichern. Es werden insgesamt 32 Messwerte unterschieden. Entwickeln Sie einen Hamming Blockcode. Zeigen Sie, dass Sie eine Fehlerstelle sicher korrigieren können. Prüfen Sie, ob der Code "dichtgepackt" ist und interpretieren Sie das Ergebnis.

Um 32 Messwerte unterscheiden zu können, werden 5 bit benötigt ($2^5 = 32$). Die Anzahl der Nachrichtenbits $m$ muss also mindestens gleich 5 sein. Einen Hamming Blockcode mit dieser Anzahl Nachrichtenstellen gibt es aber nicht. Den nächst grösseren Hamming Blockcode findet man, indem man die Formel $m = 2^k - k - 1$ anwendet. Wir suchen also den kleinsten Wert für $k$, bei dem $m$ grösser oder gleich 5 wird:

$$\begin{aligned}
k = 2: &\qquad m = 2^2 - 2 - 1 = 1 &\qquad (< 5)\\
k = 3: &\qquad m = 2^3 - 3 - 1 = 4 &\qquad (< 5)\\
k = 4: &\qquad m = 2^4 - 4 - 1 = 11 &\qquad (> 5)
\end{aligned}$$

Es werden also 4 Kontrollstellen benötigt, um den gewünschten Hamming Blockcode zu konstruieren. Die Blocklänge kann dann einfach berechnet werden aus der Formel $n = m + k$. Für $k = 4$ ergibt das eine Blocklänge von $n = 11 + 4 = 15$.

Ein Beispiel eines solchen Hamming Blockcodes wurde bereits in der letzten Aufgabe bearbeitet:

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

Die Hammingdistanz für diesen Code ist $h = 3$, wie das für alle Hamming Blockcodes der Fall ist. Die Anzahl der sicher korrigierbaren Fehler ist dann $e = \frac{h-1}{2} = \frac{3-1}{2} = 1$, wie es in der Aufgabenstellung gefordert war.

Abbildung 11: Hammming Codeblock Beispiel

#### 2.9.1 Fehlersyndrom ermitteln

Wenn $x_1$ falsch dann aus Matrix Spalte 1 herausschreiben.
Wenn zwei Fehler beide Spalten addieren.

#### 2.9.2 Anzahl der sicher erkennbaren Fehler

$e^* = h - 1$

#### 2.9.3 Anzahl der sicher korrigierbaren Fehler

**h gerade:**

$h = 2e + 2 => e = \frac{h-2}{2}$

**h ungerade:**

$h = 2e + 1 => e = \frac{h-1}{2}$

Wenn mehr als die Anzahl der sicher korrigierbaren Fehler auftreten, dann wird entweder falsch korrigiert oder der Fehler wird nicht gefunden.

#### 2.9.4 Anzahl berechnen

**Anzahl möglicher Codeworte:** $2^{m+k}$
**Anzahl gültiger Codeworte:** $2^m$

#### 2.9.5 Dichtgepackt

Der Coderaum ist dichtgepackt, wenn sich alle Codewörter(gültige und ungültige) in einer Korrigierkugel befinden. Es sei:

- **n** die Dimension des Codes (Anzahl aller CW=$2^n$)
- **m** die Dimension der Nachrichten (Anzahl aller gültigen CW=$2^m$)
- **k** die Dimension der Kontrollstellen mit $n = m + k$

**Wichtig:** Falls Hammingdistanz gerade –> kann nicht dichtgepackt sein.

$2^m * \sum_{w=0}^{e} \binom{n}{w} \leqslant 2^n$

#### 2.9.6 Blockcodes

#### 2.9.7 Hamming Code

- Hammingdistanz immer 3
- Linearer Blockcode

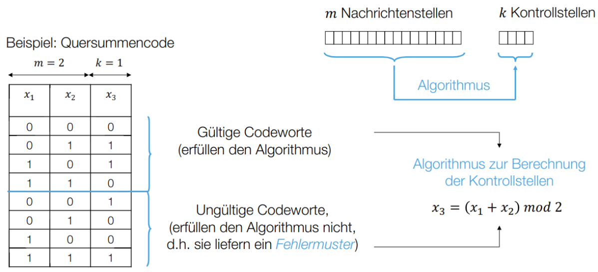



Abbildung 12: Blockcodes

## 2.10 Zyklische Codes

- Generatormatrix (Prüfmatrix) kann durch Generatorpolynom beschrieben werden.
- Höchster Grad des Generatorpolynoms entspricht der Anzahl Kontrollstellen (gilt auch für CRC)
- $n = 2^k - 1$
- $n = m + k$

### 2.10.1 Kontrollstellen bestimmen

**Mehrfachaddition:**

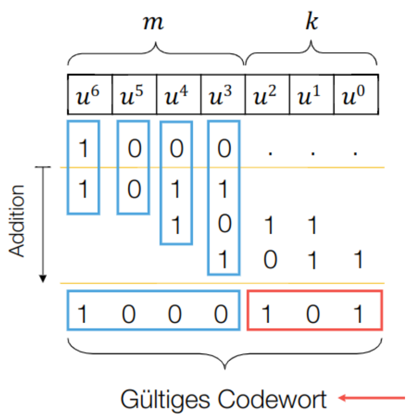



Abbildung 13: Mehrfachaddition

**Polynomdivision:**

Das Generatorpolynom von **1011** lautet => $1 * u^3 + 0 * u^2 + 1 * u^1 + 1 * u^0$

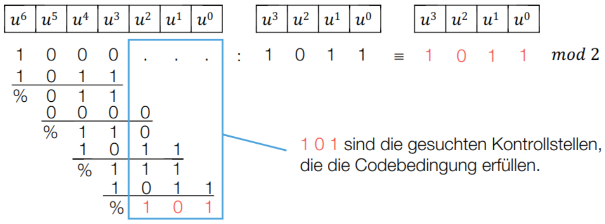



Abbildung 14: Polynomdivision

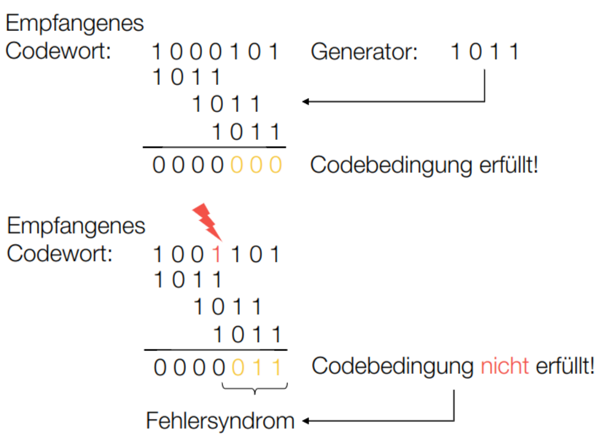



Abbildung 15: Beispiel

### 2.10.2 Prüfen der Codebedingung

### 2.10.3 Prüfmatrix

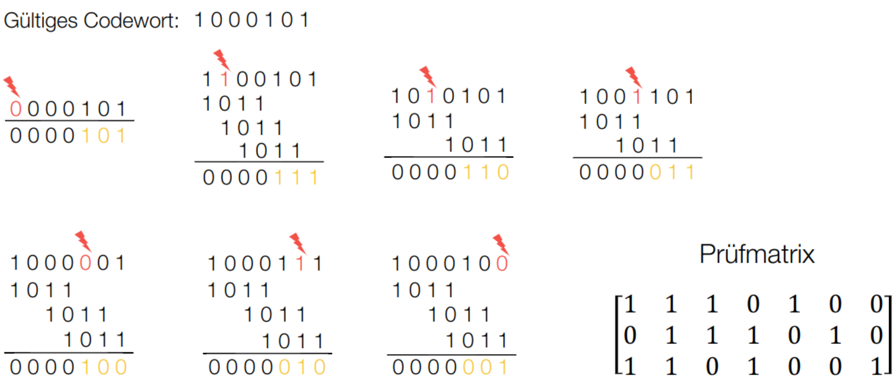



Abbildung 16: Prüfmatrix

#### 2.10.4 CRC Code

- Hammingdistanz h = 4
- Wird gebildet durch die Multiplikation eines primitiven Polynoms mit dem Term (1+x)

$g(x) = (p(x) * (1 + x))mod2$

**Abramson-Code:** $2^{k-1} - 1$

#### 2.10.5 Rückgekoppeltes Schieberegister

Schieberegister von $1 + x^2 + x^4 + x^5$

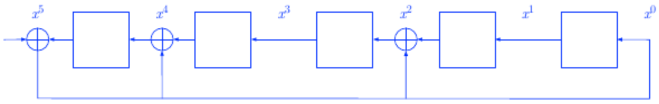



Abbildung 17: Schieberegister

## 2.11 Faltungscodes

**Bedeutung:**

- (3,1,2) Faltungscode
  - 3 Ausgänge
  - 1 Eingang
  - 2 Speicherzellen

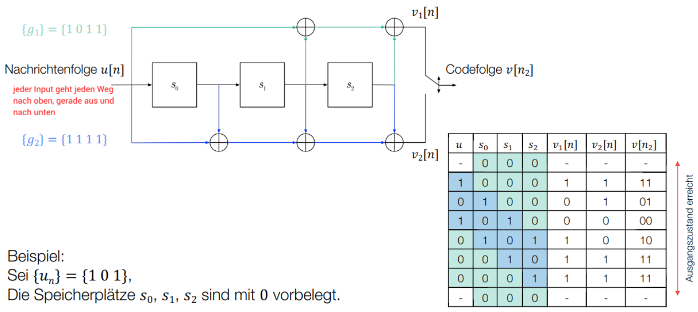



Abbildung 18: Encoderschaltung

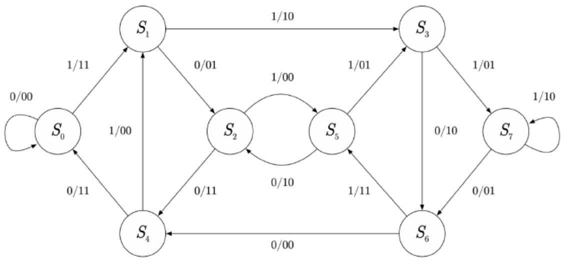



Abbildung 19: Zustandsdarstellung

| Zustandsgrösse | | | Zustand $S_i$ |
|---|---|---|---|
| $s_0$ | $s_1$ | $s_2$ | $(i = s_0 \cdot 2^0 + s_1 \cdot 2^1 + s_2 \cdot 2^2)$ |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 2 |
| 1 | 1 | 0 | 3 |
| 0 | 0 | 1 | 4 |
| 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 6 |
| 1 | 1 | 1 | 7 |

Abbildung 20: Zustandstabelle

### 2.11.1 Encodergedächtnis und Tailbits

**Encodergedächtnis:** Anzahl Speicherstellen
**Tailbits:** Anzahl Speicherstellen

### 2.11.2 Blockcoderate

Blockcoderate R = $\frac{AnzahlCodierteBits}{AnzahlGeneratorpolynom*(AnzahlCodierteBits+AnzahlSpeicherstellen}$

### 2.11.3 Guter Code?

Es handelt sich um einen guten Code wenn der Unterschied der Ausgabe bei einem Zustandsübergang immer maximal ist.

### 2.11.4 Katastrophaler Code?

Wenn es Zyklen ohne Gewichtszunahme gibt dann ist es ein katastrophaler Code.

### 2.11.5 Anzahl Bits für Berechnung der Ausgangsbits

Immer das aktuelle Bit + Anzahl Speicherstellen.

### 2.11.6 Impulsantwort der Encoderschaltung

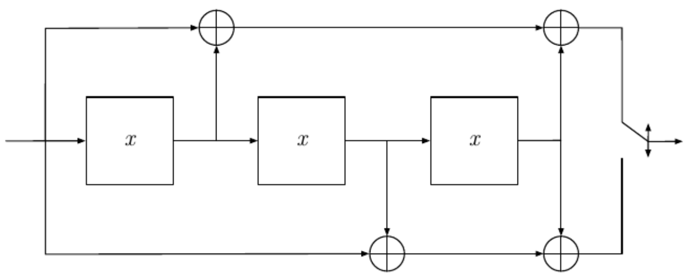



Abbildung 21: Impulsantwort Beispiel

**Anzahl Bits:** Aktuelles + 3 Speicherstellen = 4

**Impulsantwort:**
$u[n] = 1, 0, 0, 0$
$v[n] = 11, 01, 10, 11$

**Als Polynom dargestellt:**
$g_1(x) = 1 + x +^3$
$g_2(x) = 1 + x^2 + x^3$

### 2.11.7 Anzahl Zustände

$2^{AnzahlSpeicherstellen}$

## 2.12 Quellencodierung

### 2.12.1 Mittlere Codewortlänge

$L = \sum_{i=1}^{N} p(x_i) * L(x_i) [bit/Zeichen]$

- Die diskreten Zeichen der Quelle werden auf binäre CW abgebildet
- Günstig ist wenn die mittlere Codewortlänge L möglichst klein ist

### 2.12.2 Shannon'sches Codierungstheorem

- Für jede beliebige zugehörige Binärcodierung mit Präfixeigenschaft ist die mittlere Codewortlänge nicht kleiner als die Entropie **H(X)**
- Für jede beliebige Quelle kann eine Binärcodierung gefunden werden, so dass die folgende Ungleichung gilt:

  $H(X) \leqslant L \leqslant H(X) + 1$

**Redundanz der Quelle:**
$R_Q = H_0 - H(X) [bit/Zeichen]$

**Redundanz des Codes:**
$R_C = L - H(X) [bit/Zeichen]$

| $x$ | $P(x)$ | $L(x)$ | $P(x) * L(x)$ |
|---|---|---|---|
| $A$ | 0.5 | 1 | 0.5 |
| $B$ | 0.25 | 2 | 0.5 |
| $C$ | 0.1 | 3 | 0.3 |
| $D$ | 0.1 | 4 | 0.4 |
| $E$ | 0.05 | 4 | 0.2 |
| | | $L$ | 1.9 |

$$R_C = L - H(X) = 1.9 - 1.88 = 0.02\ bit$$

Das entspricht einer Verbesserung von $100\% - 100\% * \frac{0.02}{1.12} = 98.21\%$.

Abbildung 22: Redundanz des Codes berechnen

**Aufgabe 3.**

Zur Übertragung von Nachrichten werden 8 verschiedene Zeichen $(A, B, \ldots, H)$ verwendet. Bisher waren die Nachrichtenzeichen alle mit der gleichen Wortlänge von drei Bit codiert. Bei einer mittleren Übertragungsrate von $6000 \frac{\text{Zeichen}}{s}$ entspricht das einer Datenrate von $18000 \frac{bit}{s}$. Die Auswertung von 18000 übertragenen Zeichen ergab die folgende Häufigkeitsverteilung:

| Zeichen | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ | $G$ | $H$ |
|---|---|---|---|---|---|---|---|---|
| Anzahl | 1200 | 4800 | 900 | 3990 | 1445 | 2900 | 2005 | 760 |

1. Wie gross ist die Redundanz $R_Q$ der Quelle?

| $x$ | $P(x)$ | $I(x)$ | $P(x) * I(x)$ |
|---|---|---|---|
| $A$ | 0.0667 | 3.9 | 0.26 |
| $B$ | 0.2667 | 1.9 | 0.51 |
| $C$ | 0.05 | 4.32 | 0.22 |
| $D$ | 0.221 | 2.17 | 0.48 |
| $E$ | 0.08 | 3.64 | 0.29 |
| $F$ | 0.161 | 2.63 | 0.42 |
| $G$ | 0.111 | 3.17 | 0.35 |
| $H$ | 0.042 | 4.57 | 0.19 |
| | | $H(X)$ | 2.72 |

$$R_Q = H_0 - H(X) = log_2(8) - 2.72 = 0.28\ bit$$

2. Entwickeln Sie nach Huffman eine redundanzärmere Codierung der Codeworte. Was ist die resultierende Redundanzminderung?

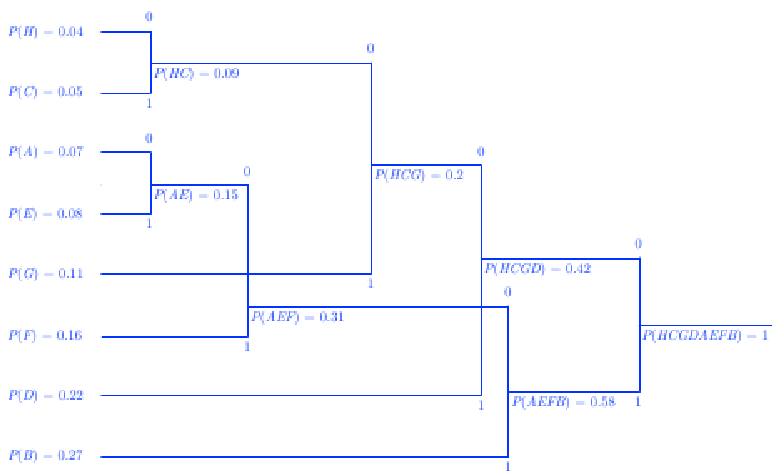



Abbildung 23: Huffmann Beispiel 1

Gemäss dieser Huffman-Codierung werden die Zeichen wie folgt codiert:

| Zeichen | $H$ | $C$ | $A$ | $E$ | $G$ | $F$ | $D$ | $B$ |
|---|---|---|---|---|---|---|---|---|
| CW | 0000 | 0001 | 1000 | 1001 | 001 | 101 | 01 | 11 |

Um daraus die Redundanz $R_C$ zu berechnen, braucht man zuerst die mittlere Codewortlänge $L$:

| $x$ | $P(x)$ | $L(x)$ | $P(x) * L(x)$ |
|---|---|---|---|
| $A$ | 0.0667 | 4 | 0.2668 |
| $B$ | 0.2667 | 2 | 0.5334 |
| $C$ | 0.05 | 4 | 0.2 |
| $D$ | 0.221 | 2 | 0.442 |
| $E$ | 0.08 | 4 | 0.32 |
| $F$ | 0.161 | 3 | 0.483 |
| $G$ | 0.111 | 3 | 0.333 |
| $H$ | 0.042 | 4 | 0.168 |
| | | $L$ | 2.7462 |

$$R_C = L - H(X) = 2.75 - 2.72 = 0.03\ bit$$

Diese Redundanz gilt es nun zu vergleichen mit der Redundanz bei einer fixen Codewortlänge von 3 Bit. Diese ergibt sich wie folgt:

$$R'_C = 3 - H(X) = 3 - 2.72 = 0.28\ bit$$

Somit ergibt sich eine Redundanzminderung von $R'_C - R_C = 0.28 - 0.03 = 0.25\ bit$.

Um wie viel Prozent kann die mittlere Datenrate gesenkt werden, damit die gleiche Information für die Übertragung gleich lange braucht?

Die Verbesserung kann ohne die Verwendung der $18000\frac{bit}{s}$ berechnet werden. Dazu muss eigentlich nur das Verhältnis der ganzen Nachrichten berechnet werden.

$$100\% - 100\% * \frac{1.03}{1.28} = 19.53\%$$

Die neue Datenrate könnte man folgendermassen berechnen:

$$18000\frac{bit}{s} * (1 - 0.1953) = 14484.6\frac{bit}{s}$$

Abbildung 24: Huffmann Beispiel 2

### 2.12.3 Markov Diagramm

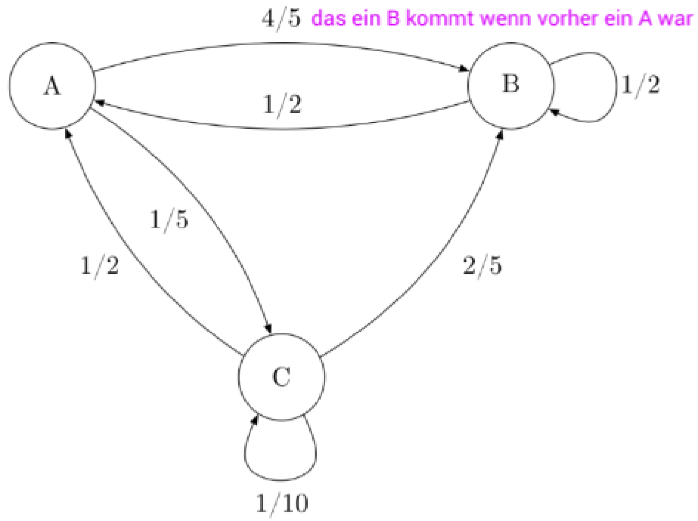
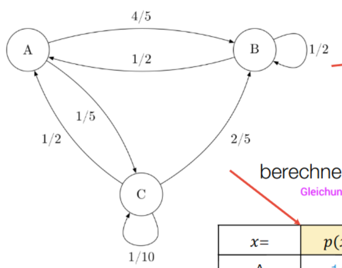



Abbildung 25: Markov Diagram

### 2.12.4 Diskrete Quelle mit Gedächtnis




ablesen

bedingte Wahrscheinlichkeit

| $p(y\|x)$ | | $y=$ A | $y=$ B | $y=$ C |
|---|---|---|---|---|
| $x=$ | A | 0 | 4/5 | 1/5 |
| | B | 1/2 | 1/2 | 0 |
| | C | 1/2 | 2/5 | 1/10 |

berechnen

Gleichungssystem

| $x=$ | $p(x)$ |
|---|---|
| A | 1/3 |
| B | 16/27 |
| C | 2/27 |

berechnen

| $p(x,y)$ | | $y=$ A | $y=$ B | $y=$ C |
|---|---|---|---|---|
| $x=$ | A | 0 | 4/15 | 1/15 |
| | B | 8/27 | 8/27 | 0 |
| | C | 1/27 | 4/135 | 1/135 |

Abbildung 26: Beispiel

**Taschenrechner Gleichungssystem:**

$$\operatorname{solve}\left(\begin{cases} a = b \cdot 1/2 + c \cdot 1/2 \\ b = a \cdot 4/5 + b \cdot 1/2 + c \cdot 2/5 \\ c = a \cdot 1/5 + c \cdot 1/10 \\ 1 = a + b + c \end{cases}, a, b, c\right)$$

Abbildung 27: Gleichungssystem Nspire

## 2.13 Komprimierung

Das Ziel der Datenkomprimierung ist, den Aufwand der Datenspeicherung und Datenübertragung zu reduzieren.

D.h. Entfernen von Redundanz und Irrelevanz

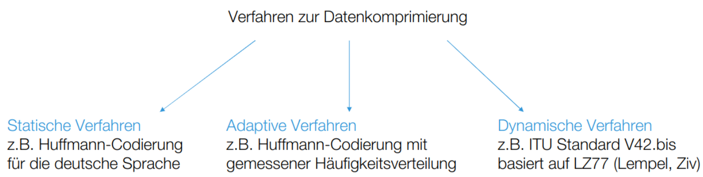



Abbildung 28: Komprimierungsarten

### 2.13.1 Huffman-Codierung

Verfahren zur Entwicklung eines Codes mit minimaler mittlerer Codewortlänge.
Rekursives Verfahren, d.h. der Binärbaum wird nicht von der Wurzel, sondern von den Blättern aus entwickelt.

**Verfahren:**

- Ordne die Zeichen gemäss ihrer Auftrittswahrscheinlichkeit
- Die beiden Zeichen mit der kleinsten Auftrittswahrscheinlichkeit haben die gleiche CW-Länge $L_N$
- Sei $L_N$ die mittlere CW-Länge für eine Quelle mit N Zeichen und $L_{N-1}$ die mittlere CW-Länge für den Fall, dass die beiden letzten zu einem einzigen Zeichen zusammengefasst werden.

## 2.14 Kryptologie

- Symmetrische Verfahren (Ein Schlüssel für Ver/Entschlüsseln)

  Caesar Chiffre

  Transpositionverfahren

- Asymmetrische Verfahren (Ein Schlüssel Ver- Ein Schlüssel Entschlüsseln)
  Modulo Rechnung und inverse Zahlen
  Eulerfunktion
  Satz von Euler
  RSA
  Euklidischer Algorithmus und Inverser Euklidischer Algorithmus
  Grosse Zahlen
- Substitutionsverfahren
  Die Buchstaben des Klartextes werden durch andere Symbole ersetzt
- Transpositionsverfahren
  Die Zeichenfolge des Klartextes wird nicht ersetzt sondern verwürfelt
- Playfair-Chiffre
  Gruppen von Zeichen werden codiert

### 2.14.1 Caesar Chiffre

**Knacken:**
Bei einer Cesar-Verschlüsselung werden die Häufigkeiten der Quellzeichen nicht verwürfelt. Ist die Zielsprache bekannt ist auch das häufigste Zeichen der Sprache bekannt. Ist der Erhaltene Code gross genug kannn mit einer einfachen Häufigkeitsanalyse der Schlüssel ermittelt werden.

### 2.14.2 RSA

$n = p * q$
$\phi(n) = (p-1) * (q-1)$