1. Wahrscheinlichkeits-Rechnung

E	Ereignis
h	Anzahl zutreffende Fälle (gültige Fälle)
n	Gesamt Anzahl Fälle (alle Fälle)
p	Wahrscheinlichkeit für das Eintreffen des Ereignisses
q	Wahrscheinlichkeit des Nicht-Eintreffens
Pr{*}	Wahrscheinlichkeit von *, * steht für ein oder mehrere Ereignisse)
E*	Ereignisse

Einführung

Der Wertebereich der Wahrscheinlichkeit lieg zwischen [0,1]. Tritt ein Ereignis sicher nicht auf, so ist die Wahrscheinlichkeit gleich 0. Tritt es jedoch sicher auf, sprich in jedem Fall, ist die Wahrscheinlichkeit 1.

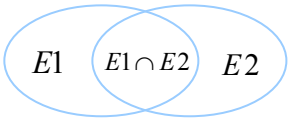
Eintreffen

$$p = \Pr\{E\} = \frac{h}{n}$$

Nicht-Eintreffen

$$q = \Pr\{\overline{E}\} = \frac{n-h}{n} = 1 - p$$

Regeln für mehrere Ereignisse



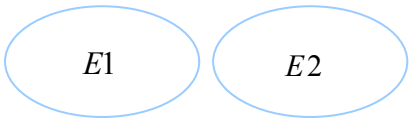
$E1$ *und* $E2$	$\Pr(E1 \cap E2) \equiv \Pr\{E1E2\} \equiv \Pr\{E1\} \cdot \Pr\{E2\}$
$E1$ *oder* $E2$	$\Pr\{E1 \cup E2\} \equiv \Pr\{E1 + E2\} \equiv \Pr\{E1\} + \Pr\{E2\} - \Pr\{E1E2\}$
	$-\Pr\{E1E2\}$, weil die Vereinigung doppelt gezählt wurde.

Bedingte Wahrscheinlichkeit (Abhängige Ereignisse)

Eintreten von $E2$ *unter der Voraussetzung* $E1$ *bereits eingetreten.*

$$\Pr\{E2 \mid E1\} = \frac{\Pr\{E1E2\}}{\Pr\{E1\}}$$

Ausschliessende Ereignisse



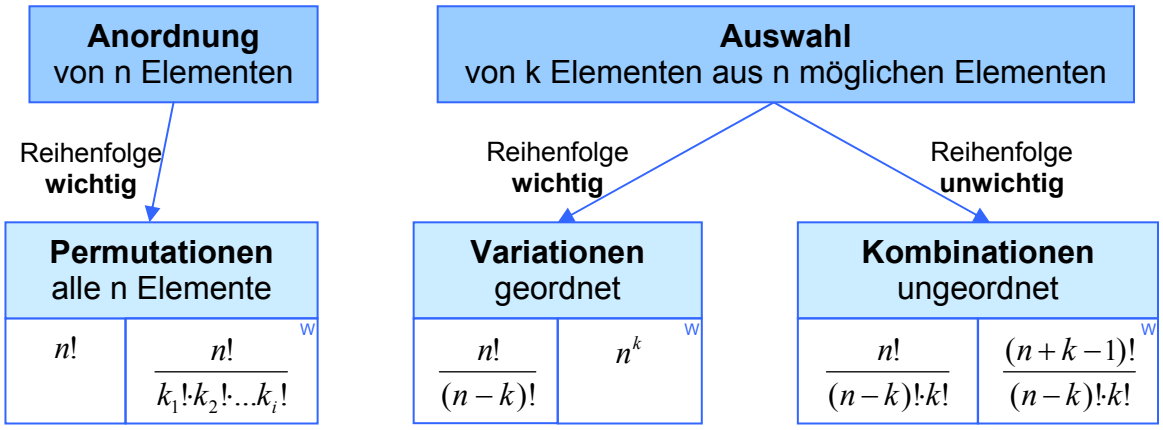
$\Pr\{E1E2\} = 0$ *Eintreffen von* $E1$ *und* $E2$ *gleichzeitig ist nicht möglich.*
Beispiel: Münz würf, nur Kopf oder Zahl möglich, beides geht nicht.

$E1$ *oder* $E2$	$\Pr\{E1 + E2\} = \Pr\{E1\} + \Pr\{E2\}$

2. Kombinatorik

$n!$	n-Fakultät
A	Anzahl Möglichkeiten (Eigene Definition)
W	Mit Wiederholungen (Eigene Definition)

Formeln



Permutationen

(Anzahl der Anordnungen ohne Wiederholungen)

$$P_n \quad gibt \quad A = n!$$

Permutationen mit Wiederholungen

(Jeweils k_i Elemente sind gleich)

$$P_n^{(k_1,k_2,\ldots,k_i)} \quad gibt \quad A = \frac{n!}{k_1!\cdot k_2!\cdot\ldots k_i!}$$

Variationen

(geordnete k-Tupel ohne Wiederholungen aus n Zeichen)

$$V_n^k \quad gibt \quad A = \frac{n!}{(n-k)!} = \binom{n}{k}\cdot k!$$

Variationen mit Wiederholungen

(geordnete k-Tupel mit Wiederholungen aus n Zeichen)

$$^{w}V_n^k \quad gibt \quad A = n^k$$

Kombinationen

(ungeordnete k-Teilmengen aus einer n-Menge)

$$C_n^k = C_n^{n-k} = \binom{n}{k} = \binom{n}{n-k} = \frac{V_n^k}{P_n} \quad gibt \quad A = \frac{n!}{(n-k)!\cdot k!}$$

$$\binom{n}{0} = \binom{n}{n} = 1 \;,\; \binom{n}{1} = n \;,\; C_n^k = C_{n+1}^{k+1} - C_n^{k+n} \;,\; C_{n+1}^{k+1} = C_n^k + C_n^{k+n}$$

Kombinationen mit Wiederholungen

(ungeordnete k-Tupel mit Wiederholungen aus n Zeichen)

$$^{w}C_n^k = \binom{n+k-1}{k} \quad gibt \quad A = \frac{(n+k-1)!}{(n-k)!\cdot k!}$$

3. Information, Entropie

N	Anzahl Zeichen der Nachricht
τ	Übertragungszeit für ein Quellzeichen
x_k	Das k-te Zeichen der Nachricht

Entscheidungsgehalt

Anzahl Elementar-Entscheidungen [0,1] für eine Nachricht.

Entscheidungsgehalt: $H_0 = lb(N)\ ,\ [Bit]$

Entscheidungsfluss: $H_0^* = \frac{lb(N)}{\tau}\ ,\ [\frac{Bit}{s}]$

Informationsgehalt

Anzahl Entscheidungen für die Bestimmung eines Zeichens.

Auftrittswahrscheinlichkeit: $p(x_k)$

Informationsgehalt: $I(x_k) = lb(\frac{1}{p(x_k)})\ ,\ [Bit]$

Entropie

Mittlerer Informationsgehalt der Quelle, Entscheidungen im Mittel pro Zeichen.

Entropie: $H(X) = \sum_{k=1}^{N} p(x_k) \cdot I(x_k)\ ,\ [\frac{Bit}{Zeichen}]$

Entropie ist maximal, wenn jedes Zeichen gleichhäufig vorkommt.

Redundanz

Um wie viel die Codewortlänge optimiert werden kann, bei Anpassen der Codewortlänge an die Auftrittswahrscheinlichkeit.

Redundanz der Quelle: $R_Q = H_0 - H(X)\ ,\ [\frac{Bit}{Zeichen}]$

Redundanz des Codes: $R_C = L - H(X)\ ,\ [\frac{Bit}{Zeichen}]$

Mittlere Codewortlänge: $L = \sum_{k=1}^{N} p(x_k) \cdot L(x_k)\ ,\ [\frac{Bit}{Zeichen}]$ *(nur ganze Zahlen)*

4. Binärcodierung und Quellencodierung

Binärcodierung

Präfixeigenschaft (kommafrei)

Ein Code mit Präfixeigenschaft ist ein kommafreier Code, das bedeutet jeder Code ist eindeutig einem Zeichen zuweisbar, auch wenn die Codewortlänge unterschiedlich ist und nicht bekannt ist.

Shannon'sches Codierungstheorem: $H(X) \le L$ $H(X) \le L \le H(X)+1$

Quelle mit und ohne Gedächtnis

Bei einer Quelle mit Gedächtnis können wir voraus ahnen welches Zeichen als nächstes kommt, daher sinkt der Informationsgehalt und die Redundanz nimmt zu.

Quelle ohne Gedächtnis: $P(x_i, y_k) = p(x_i) \cdot p(y_k)$

Quelle mit Gedächtnis: $P(x_i, y_k) = p(x_i) \cdot p(y_k \mid x_i)$ *, z.B. Deutsche Sprache*

Verbund Entropie: $H\langle X, Y\rangle = H\langle X\rangle + H\langle Y \mid X\rangle$

Quellencodierung - Datenkompression

Ziel

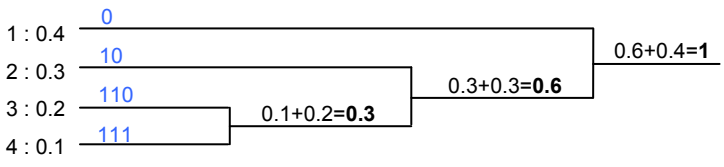
Ziel ist es den Aufwand der Datenspeicherung und Datenübertragung zu reduzieren, das heisst Entfernen von Redundanz und Irrelevanz.

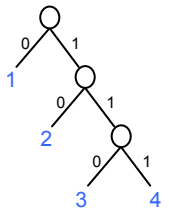
Huffmann-Codierung

Adaptives Verfahren (gemessene Häufigkeitsverteilung, hat Präfixeigenschaft) Verfahren zur Entwicklung eines Codes mit minimaler mittlerer Codewortlänge.

Verfahren:

1. Zeichen gemäss ihrer Auftrittswahrscheinlichkeit ordnen.
2. Jeweils die beiden Zeichen mit der kleinsten Auftrittswahrscheinlichkeit zusammenzählen. Beispiel: $x_k : p(x_k)$ - 1 : 0.4, 2 : 0.3, 3 : 0.2, 4 : 0.1





Lempel-Ziv

Dynamische Verfahren

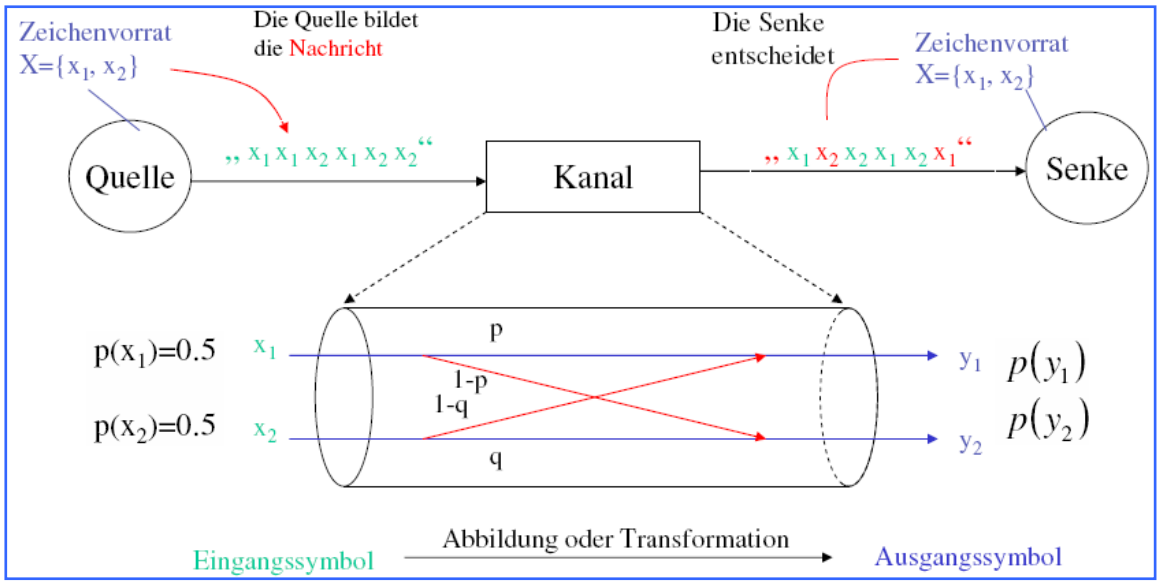
Verfahren zur Ausnutzung wiederkehrender Muster. Es wird nicht der Code sondern die codierten Phrasen übertragen und in einem Phrasenspeicher („Wörterbuch") gespeichert. Als Phrasenspeicher wird ein ständig wachsender Baum erzeugt mit den Knoten als Referenzen. Treten dieselben Phrasen wiederholt auf, wird nicht die Phrase sondern nur der Knoten des Phrasenspeichers referenziert.

Problem: Effiziente Umsetzung des Phrasenspeichers.

5. Kanalcodierung – Modell, Matrix und Transinformation

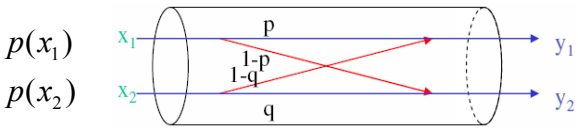
$p(x_n)$ Eingangswahrscheinlichkeit für das Zeichen x_n
$p(y_n)$ Ausgangswahrscheinlichkeit für das Zeichen y_n. Symbol y_1 entspricht x_1.

Kanalmodell



Kanalmatrix

Zeigt die Wahrscheinlichkeit für den Übergang von einem Eingangssymbol in das gleiche oder ein anderes Ausgangssymbol.

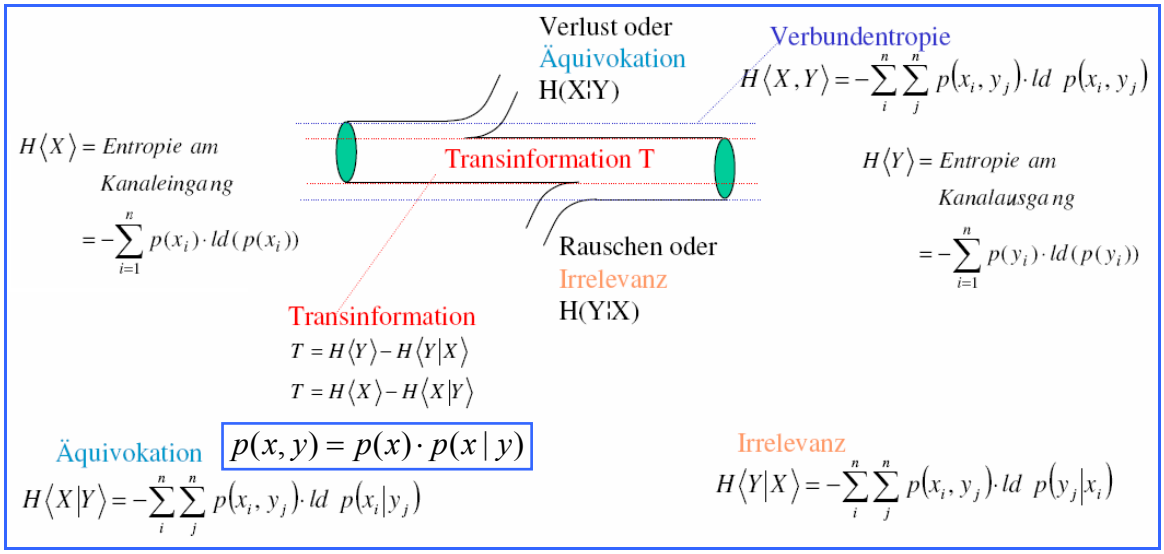


$$p(y_1) = p(x_1) \cdot p + p(x_2) \cdot (1-q)$$
$$p(y_2) = p(x_1) \cdot (1-p) + p(x_2) \cdot q$$

$$p(Y \mid X) = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix} \mapsto \begin{bmatrix} \sum = 1 \\ \sum = 1 \end{bmatrix}$$

$$p(Y \mid X) = \begin{bmatrix} p(y_1 \mid x_1) & p(y_2 \mid x_1) \\ p(y_1 \mid x_2) & p(y_2 \mid x_2) \end{bmatrix}$$

Transinformation

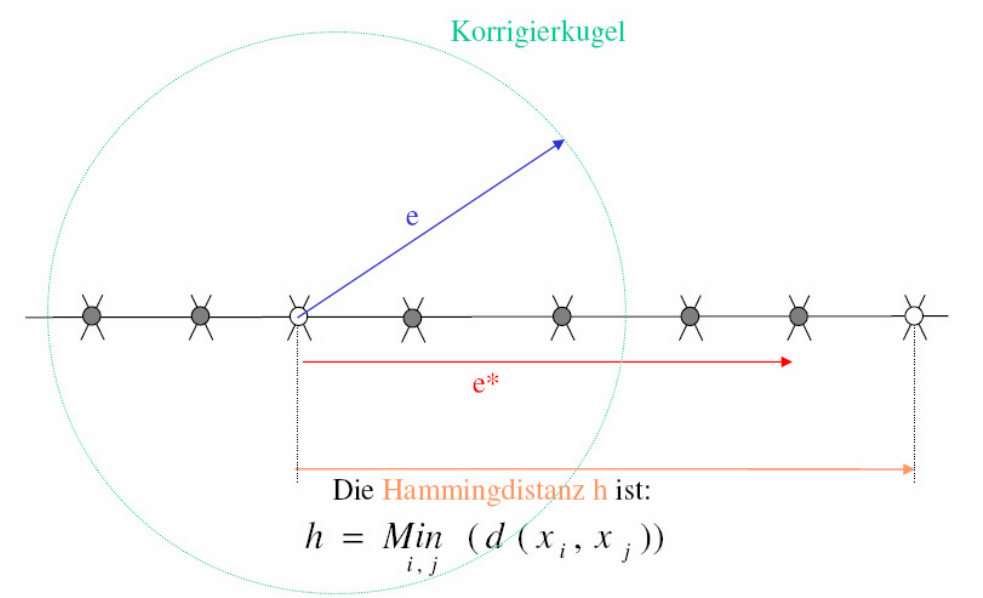


6. Kanalcodierung – Coderaum

h	Hammingdistanz
e	Korrigierbare Fehler

Coderaum

Der Coderaum enthält alle gültigen und alle ungültigen Codewörter. Folgende Abbildung zeigt den Coderaum als dreidimensionalen, flachgedrückten Würfel und veranschaulicht die **Hammingdistanz** (Distanz zum nächsten gültigen Codewort) und die **Korrigierkugel**.



Sicher erkennbare Fehler: $e^* = h - 1$

Korrigierbare Fehler: $e = \frac{e^*}{2} = \frac{h-1}{2}$ *und Abrunden auf ganze Zahlen*

Dichtgepackt

Code ist Dichtgepackt wenn sich alle Codewörter in einer Korrigierkugel befinden. Ist möglich wenn die Hammingdistanz überall gleich ist und ungerade.

Sei :

- n die Dimension des Code (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

⇒ So folgt die Codeabschätzung:

$$2^m \cdot \sum_{w=0}^{e} \binom{n}{w} \leq 2^n$$

Anzahl der CW bzw. Korrigierkugeln

Anzahl der CW pro Korrigierkugel

Anzahl aller CW

Gilt:

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

So ist der Code dichtgepackt!

7. Kanalcodierung – Blockcode, Hamming-Code

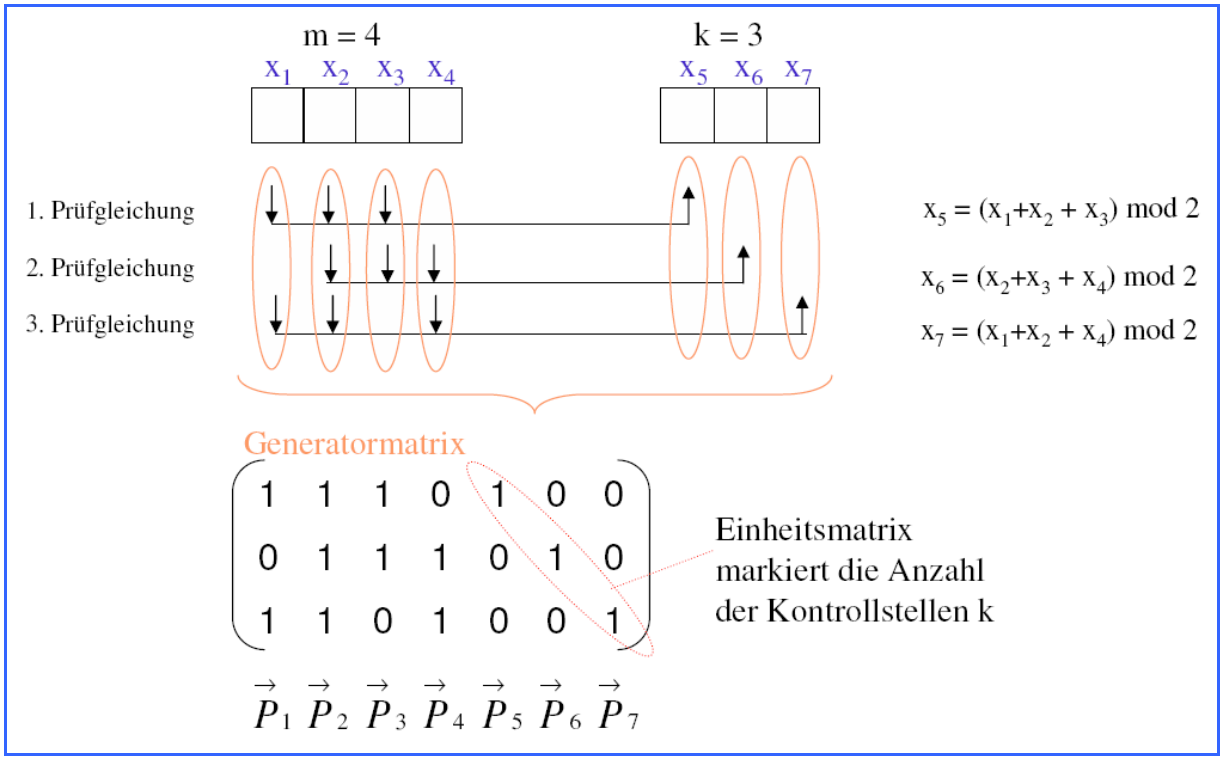
m	Anzahl Nachrichtenstellen
k	Anzahl Kontrollstellen, Prüfstellen
n	Anzahl Codewortstellen: $n = m + k = 2^k - 1$

Blockcode

Code wird als Block angeschaut und hat m Nachrichtenstellen und k Kontrollstellen, welche über einen Algorithmus berechnet werden.
Bei allen ungültigen Codeworten erfüllen die Nachrichtenstellen mit den Kontrollstellen den Algorithmus nicht und liefern ein Fehlermuster.

Hamming-Code

Ist ein Algorithmus zur Berechnung der Kontrollstellen für den Blockcode.
Alle Vektoren sind paarweise verschieden, somit gibt es ein Fehlersyndrom, das es erlaubt den Fehlerort zu lokalisieren.



Codebedingung: $\sum_i x_i \cdot \vec{P}_i \equiv \vec{0} \bmod 2$

Mod 2 berechnen: Bei allen ungeraden Zahlen ist mod 2 = 1, sonst 0

In der **Generatormatrix** stehen die ersten Spalten jeweils für je ein Nachrichtenbit und die letzten Spalten für je ein Kontrollbit. Zusätzlich steht jede Zeile für ein Kontrollbit. Im Quadrat wo sich die Kontrollbit Spalten und Kontrollbit Zeilen treffen entsteht die Einheitsmatrix.

Für das **Berechnen der Kontrollstellen** anhand der Generatormatrix ist es am einfachsten den Code über die Generatormatrix zu schreiben. Für jede Kontrollstelle (Zeile in der G.) wird bei jedem 1 in der Generatormatrix die zugehörige Nachrichtenstelle addiert und die Summe am Ende mod 2 gerechnet.

8. Kanalcodierung – Zyklische Codes

m	Anzahl Nachrichtenstellen
k	Anzahl Kontrollstellen, Prüfstellen
n	Anzahl Codewortstellen: $n = m + k$

Idee und Ziel des Zyklischen Codes

Die Generatormatrix kann durch ein Generatorpolynom beschrieben werden. Daraus folgt eine vereinfachte Berechnung der Kontrollstellen durch rückgekoppelte Schieberegister.

Generatorpolynom: $G(u) = \sum_{i=0}^{k} g_i \cdot u^i$

Codewortpolynom: $X(u) = \sum_{i=0}^{n} g_i \cdot u^i$

$$X(u) \div G(u) \equiv Q(u) \bmod 2$$
$$X(u) \equiv Q(u) \cdot G(u) \bmod 2$$

Das Codewortpolynom ist ohne Rest durch das Generatorpolynom teilbar.

Sei: m = 4, k = 3, n = 7
Nachricht: $(x_1,x_2,x_3,x_4,) = (1\ 0\ 0\ 0)$
Generator: $G(u) = u^3 + u + 1 \Rightarrow (g_3\ g_2\ g_1\ g_0) = (1\ 0\ 1\ 1)$

u^6	u^5	u^4	u^3	u^2	u^1	u^0		u^3	u^2	u^1	u^0		u^3	u^2	u^1	u^0

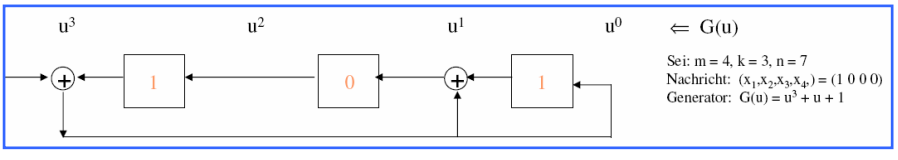
1 0 0 0 1 0 1 : 1 0 1 1 ≡ 1 0 1 1 **mod 2**

1 0 1 1
% 0 1 1
0 0 0 0
% 1 1 0
i 0 i 1
% 1 1 1
1 0 1 1
% 1 0 1

	u^6	u^5	u^4	u^3	u^2	u^1	u^0
	1	0	0	0	.	.	.
	1	0	1	1			
Addition			1	0	1	1	
				1	0	1	1
	1	0	0	0	1	0	1

101 sind die gesuchten Kontrollstellen, die die Codebedingung erfüllen.

Rückgekoppeltes Schieberegister, u^2 gehört nicht zum Generator, deshalb keine Verbindung.



Durch die Codebedingung muss die fortgesetzte Addition (mod 2) des Generators zum empfangenen CW das Nullwort ergeben. Wenn ein Fehler vorhanden ist, können die hintersten Bits mit der Generatormatrix verglichen werden um so die fehlerhafte Bitstelle zu erkennen/korrigieren.

Empfangenes Codewort:

1 0 0 0 1 0 1
1 0 1 1
1 0 1 1
1 0 1 1
0 0 0 0 0 0 0

1 0 0 1 1 0 1
1 0 1 1
1 0 1 1
0 0 0 0 0 1 1

Generatormatrix

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

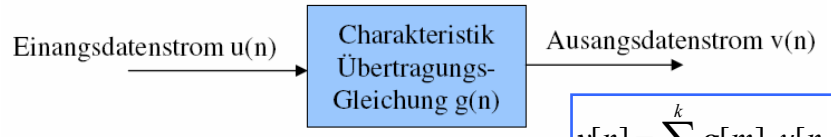
9. Kanalcodierung – Faltungscodes

$u(n)$	Eingangsdatenstrom
$v(n)$	Ausgangsdatenstrom
$g(n)$	Generator (Übertragungs-Gleichung)

Eigenschaften der Faltungscodes

Faltungscodes erlauben die fortlaufende Codierung eines kontinuierlichen Datenstroms, somit ist keine Blockbildung erforderlich. Sie werden in den GMS und UMTS Netzen und in der Funkübertragung eingesetzt.
Faltungscodes werden über ein Tupel beschrieben (Ausgänge, Eingänge, Speicher). Beispiel: (2, 1, 3), 2 Ausgänge, 1 Eingang und 3 Speicher.

Idee der Faltung

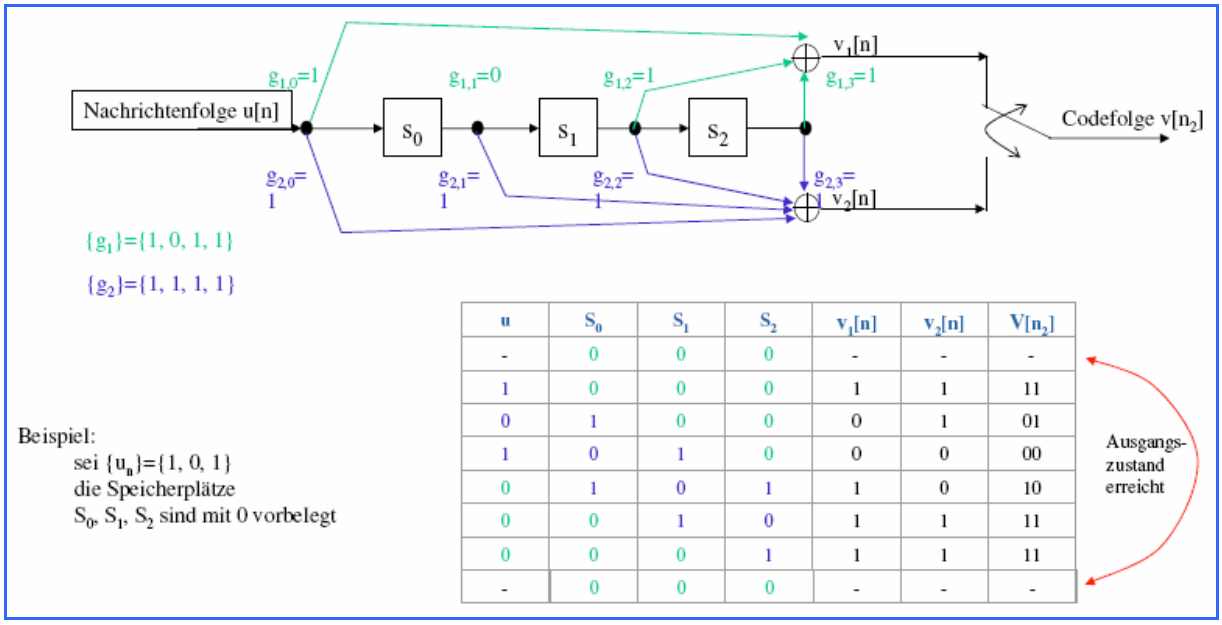


Transformation: $v(n) = u(n) * g(n)$

$$v[n] = \sum_{m=0}^{k} g[m] \cdot u[n-m]$$

Encoder-Schaltung des (2,1,3) Encoders

Hat ein Erinnerungsvermögen von 3 Bits (Anzahl Speicher) und die Codelänge verdoppelt sich, wegen den zwei Ausgängen.



u	S_0	S_1	S_2	$v_1[n]$	$v_2[n]$	$V[n_2]$
-	0	0	0	-	-	-
1	0	0	0	1	1	11
0	1	0	0	0	1	01
1	0	1	0	0	0	00
0	1	0	1	1	0	10
0	0	1	0	1	1	11
0	0	0	1	1	1	11
-	0	0	0	-	-	-

Ausgangs-zustand erreicht

Beispiel:
sei $\{u_n\}=\{1, 0, 1\}$
die Speicherplätze
S_0, S_1, S_2 sind mit 0 vorbelegt

Spezifischer Weg und Fundamentalwege

Jeder gültige Code hat einen spezifischen Weg (Pfad) durch den Coderaum (Zustandsdiagramm über Zeit). Diese Wege gliedern sich in Teilfolgen, die der Nullfolge verschieden sind. Solche Teilfolgen werden als Fundamentalwege bezeichnet.

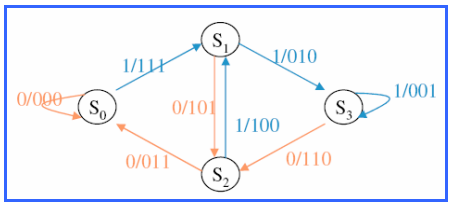
Decodierung des Faltungscodes

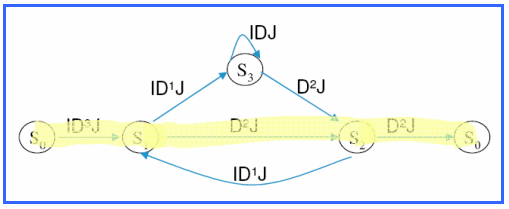
Üblicherweise wird zur Dekodierung der Viterbi Algorithmus “suche nach dem kürzesten Weg” verwendet.

10. Kanalcodierung – Beispiel Faltungscode (3,1,2)

Fundamentalwege des (3,1,2) Faltungscodes

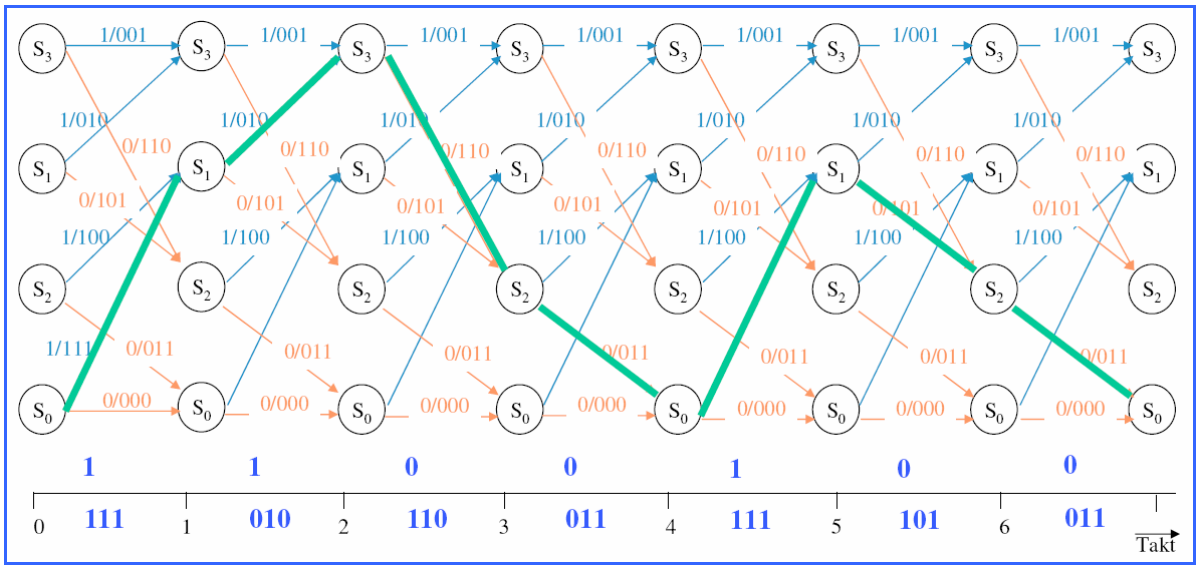
Sind die (Teil-) Wege eines Codes, die im Zustand S0 beginnen und wieder im Zustand S0 enden. Die Analyse der Fundamentalwege liefert die Struktur des Faltungscodes.





- **I** bezeichne den Zustands-übergang, der durch „1“ ausgelöst wird
- **D^I** bezeichne die Anzahl der durch den Übergang zur Codefolge hinzukommenden „1“ Bitstellen (Gewichtszunahme)
- **J** sei eine Zählvariable, die die Anzahl der Übergänge zählt
- Jede Kante eines Fundamentalweges lässt sich durch das Triplet (**I D^I J**) beschreiben
  ➔ „Kantengewicht“

Codierung des (3,1,2) Faltungscodes



Decodierung des (3,1,2) Faltungscodes