

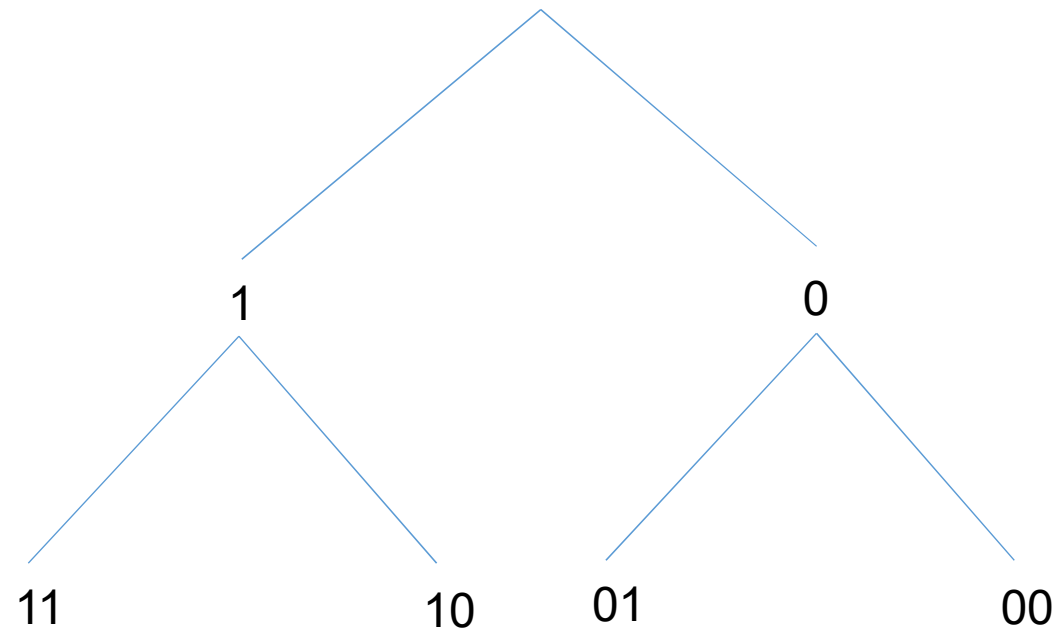
Informationsgehalt eines Zeichens

- Definition: **Zeichen** ist zu codierender „Zustand“
- Wird mit einem **Codewort** codiert/dargestellt/assoziiert
- *Informationsgehalt* eines Zeichens $H_e = \text{ld}(1/p)$
- Einheit $[H_e] = 1 \text{ Bit}$
- Durchschnittlicher Informationsgehalt bei mehreren Zeichen
- Nennt man *Entropie* einer Quelle:

$$H = \sum_{i=1}^n p(x_i) \cdot \text{ld}\left(\frac{1}{p(x_i)}\right)$$

Typische Codewortzuweisung

- Variable Länge der Codewörter möglich
- Mehr freie CW als $N=2^n$
- Präfixfreiheit bei Nachrichten wichtig



➡ Geschickte Zuweisung verkleinert mittlere Codewortlänge

Optimale Codes

- *Mittlere Codewortlänge:*
- In Bits

$$\bar{m} = \sum_{i=1}^n p(x_i) \cdot m(x_i)$$

- Minimalwert: Durchschnittlicher Informationsgehalt pro Zeichen (*Entropie*)

 Wäre optimaler Code

Annäherung über **Shannon-Fano**- oder **Huffman-Kodierung**
Es gilt allgemein:

$$\bar{m}_{Huffman} \leq \bar{m}_{Shannon-Fano}$$

Shannon Fanø Kodierung

1. Zeichen nach aufsteigender Auftrittswahrscheinlichkeit (AWS) ordnen.
2. In zwei Teilmengen teilen, sodass Unterschied der Gesamt-AWS der Mengen minimal ist.
3. Linker Zweig: mit „0“ beschriften
Rechter Zweig: mit „1“ beschriften
4. Jeweilige Teilmengen rekursiv wieder unterteilen und Vorgang wiederholen.

Durchschnittlicher Informationsgehalt:

$$H = \sum_{i=1}^n p(x_i) \cdot \lg\left(\frac{1}{p(x_i)}\right)$$

Mittlere Codewortlänge:

$$\bar{m} = \sum_{i=1}^n p(x_i) \cdot m(x_i)$$

Huffman - Kodierung

1. Aufsteigend nach AWS sortieren
2. Die beiden Zeichen mit kleinster AWS zu einer Teilmenge zusammenfügen
3. Baum beschriften: Kleinere AWS (links) mit „0“
Höhere AWS (rechts) mit „1“
4. Neuen Knoten entsprechend seiner AWS in die Zeichenliste einsortieren.

Durchschnittlicher Informationsgehalt:

$$H = \sum_{i=1}^n p(x_i) \cdot \lg\left(\frac{1}{p(x_i)}\right)$$

Mittlere Codewortlänge:

$$\bar{m} = \sum_{i=1}^n p(x_i) \cdot m(x_i)$$

Scrambling

- Übertragungsfehler häufig Bündelfehler, d.h. zeitlich konzentriert
- Bei j Zeilen und Erkennung von k Fehlern werden so $k * j$ Fehler hintereinander erkannt.

$$\textit{Overhead} = \frac{\textit{Anzahl}_{\textit{Sicherungsbits}}}{\textit{Anzahl}_{\textit{Nutzdatenbits}}}$$

- *Overhead*: Maß für die Qualität der Sicherung:

Musterformelsammlung

Informationsgehalt

Informationsgehalt H_e eines Zeichens: $H_e = \lg \frac{1}{p}$

Informationsgehalt H einer Quelle: $H = \sum_{i=1}^N p(i) \cdot \lg \frac{1}{p(i)}$

mit der Auftrittswahrscheinlichkeit $p(i)$ und $\sum_{i=1}^N p(i) = 1$

*Garantiert eine Aufgabe zu Huffman **oder** Shannon Fanø!*