

\* Informationsgehalt eines Zeichens:  $H_e = \text{ld} \left( \frac{1}{p} \right)$  Einheit: bit  
 (mehr Informationen je seltener ein Zeichen auftritt)

\* Entropie einer Quelle  
 (durchschnittlicher Informationsgehalt pro Zeichen):  $H = \sum_{i=1}^n p(i) \cdot \text{ld} \left( \frac{1}{p(i)} \right)$

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

Optimale Codes: versuchen  $\bar{m}$  zu minimieren  
 (die Entropie ist der minimal erreichbare Idealwert)

### Shannon-Fano-Code:

- (1) Zeichen nach aufsteigender ANS anordnen
- (2) in 2 Teilmengen aufteilen, sodass die Differenz der Summenwahrscheinlichkeiten der Teilmengen minimal ist
- (3) • linker Zweig: mit „0“ beschriften  
 • rechter Zweig: mit „1“ beschriften
- (4) Verfahren mit verbleibenden Teilmengen wiederholen

(ANS: Auftretenswahrscheinlichkeit)

### Huffman-Code

- (1) zu Beginn: Sortieren nach ANS
- (2) die beiden Zeichen mit der niedrigsten ANS zu einem neuen Knoten zusammenfassen
- (3) neuen Knoten entsprechend seiner ANS in die Zeichenliste einsortieren
- (4) Baum beschriften: • linker Zweig: mit „0“  
 • rechter Zweig: mit „1“

\* i.A.  $\bar{m}$  kleiner/gleich  $\bar{m}$  bei Shannon-Fano

↳ Codewörter werden in beiden Fällen von oben nach unten abgelesen!

⇒ Präfixfreiheit: kein Codewort ist Anfang eines anderen Codewortes

### Scrambling

• Fehler treten meist nicht einzeln, sondern zeitlich konzentriert auf (Bündelfehler)

• bei  $j$  Zeilen und Fehlererkennung von  $k$  Fehlern je Zeile können  $k \cdot j$  Fehler erkannt werden

