Huffman-Kodierung

1. Idee

Im deutschen und englischen Sprachgebrauch wird der Buchstabe ‘e‘ häufiger verwendet als zum Beispiel der Buchstabe ‘q‘. Um einen Text mit möglichst wenig Bits zu codieren, muss ein Weg gefunden werden, dass häufiger benutzte Zeichen eine möglichst kurze Codewortlänge besitzen.

Zusätzlich muss die FANO-Bedingung, „Wenn kein Codewort Anfangswort eines anderen Codewortes ist, dann ist jede codierte Zeichenreihe eindeutig dekodierbar.“, gelten. Diese wird durch die Eigenschaft des Huffman Algorithmus erfüllt, denn wenn man vom Anfang des Codewortes den Ästen bis zu den Blättern, den Endknoten, folgt und das entschlüsselte Zeichen identifiziert hat, fängt man wieder beim Anfangsknoten mit der Häufigkeit von 1 an.

Diese Kodierung von Zeichen bietet zudem die kleinste mittlere Codewortlänge im Gegensatz zu anderen gängigen Bit-Kodierungsverfahren.

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

Formel mittlere Codewortlänge

p(xi) → Häufigkeit des entsprechenden Symbols

m(xi) → Länge des Codewortes des entsprechenden Symbols

2. Konstruktion des Huffman-Codes

Das Verfahren konstruiert einen binären Baum mit einer Knotenmarkierung *p* und einer Kantenmarkierung *h*.

Eingabe: Text

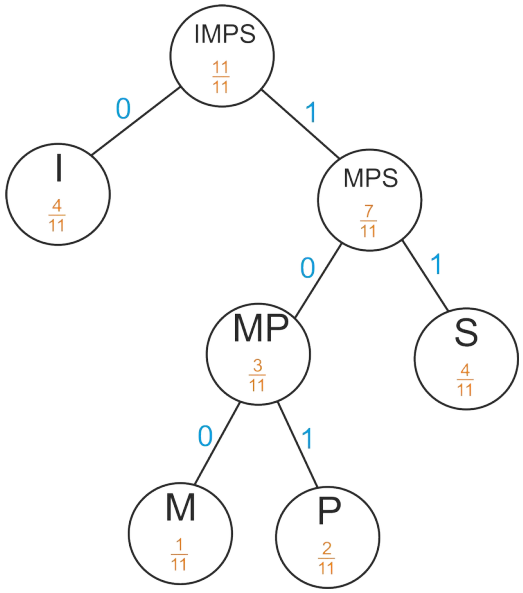
Ausgabe: Binärer Baum mit einer Knotenmarkierung *p* und einer Kantenmarkierung *h*

Methode:

1. erzeuge für jedes Zeichen *x*, das im zu kodierenden Text *t* vorkommt, einen Knoten und markiere den Knoten mit der Häufigkeit, mit der *x* im Text vorkommt
2. solange es mehr als einen Knoten gibt, zu dem keine Kante hinführt, wiederhole
   a. suche zwei Knoten *u* und *v* mit minimaler Häufigkeit $p(u)$ bzw. $p(v)$, zu denen noch keine Kante hinführt
   b. erzeuge einen neuen Knoten *w* und verbinde *w* mit *u* und *v*; markiere die linke Kante mit 0, die rechte mit 1; markiere den Knoten *w* mit $p(u) + p(v)$

Nach Konstruktion dieses Baumes ergibt sich für jedes Zeichen *x* die Kodierung *c*(*x*) als Folge der Kantenmarkierungen auf dem Pfad von der Wurzel zu dem Blatt, das zu *x* gehört.

Beispielbaum: MISSISSIPPI → M($\frac{1}{11}$), I ($\frac{4}{11}$), S ($\frac{4}{11}$), P ($\frac{2}{11}$)



3. Kodierung und Dekodierung

Zur Kodierung der Zeichenkette muss nun der o.a. Baum Zeichen für Zeichen durchgegangen werden und vom obersten Knoten bis zum Zielknoten in den Blättern jedes Bit aufschreiben, welches an den Kanten steht.
Vom obigen Beispiel „MISSISSIPPI“ wird nun die Bit-Kette „100011110111101011010“ generiert.

Zur Dekodierung muss der Baum von der Wurzel bis zu den Blättern Bit für Bit durchgegangen werden. Wenn man in den Blättern angekommen ist und ein Zeichen dekodiert hat, fängt man wieder bei der Wurzel an mit den nächsten Bits.

Bei einer 8-Bit Zeichenkodierung würde die Zeichenkette „MISSISSIPPI“ 88 Bits benötigen. Bei einer 4-Bit Kodierung würden es 44 Bits sein.
Mit Hilfe der Huffman-Kodierung konnte eine Bit-Kette von 21 Bits generiert werden, welche ca. 76% weniger Bits benötigt.

Wenn die o.a. Werte für die Häufigkeit und Codewortlänge pro Zeichen in die, in Abschnitt 1, Formel für die mittlere Codewortlänge eingibt, bekommt man eine mittlere Codewortlänge von ca 1,91 Bits pro Zeichen.

$$\overline{m}=(\frac{1}{11}*3)+(\frac{4}{11}*1)+(\frac{4}{11}*2)+(\frac{2}{11}*3)=\frac{21}{11}\approx 1{,}91$$

Nachteile dieser Kodierung ist, dass entweder der Häufigkeitsbaum vollständig gesendet werden muss oder im Ziel ein Muster vorhanden sein muss, wo nur die Häufigkeiten übermittelt werden müssen.