

Übungen zu Einführung in die Informatik II

Aufgabe 8 **Eigenschaften der Entropie**

Auf dem ersten Blatt Aufgabe 1d) haben wir durch Einsetzen nur eine Richtung des Beweis gezeigt für $H(S) = \text{ld } q \Leftrightarrow p_i = \frac{1}{q} \quad \forall i$.

Zeigen Sie nun, dass auch die andere Richtung der Äquivalenzrelation gilt.

Aufgabe 9 **Optimalität der Huffmancodierung**

Gegeben sei eine Shannonsche Nachrichtenquelle S mit einem Alphabet \mathbf{Z} , $|\mathbf{Z}| = q$, ein binäres Codierungsalphabet $\mathbf{T} = \{O, L\}$ und eine Codierung $c : \mathbf{Z} \rightarrow \mathbf{T}^*$.

Beweisen Sie, dass die Huffmancodierung c_H einen optimalen Präfixcode für \mathbf{Z} liefert. D.h. es lässt sich davon eine Codierung mit der kürzestmöglichen mittleren Codelänge ableiten.

Aufgabe 10 **LZW-Kompressionsalgorithmus**

In dieser Aufgabe soll die Funktionsweise des LZW-Algorithmus aus der Vorlesung anhand eines Beispiels von Hand nachvollzogen werden.

- a) Gehen Sie die Schritte durch, die bei der Komprimierung der Zeichenkette ABRAKADABRA-KADABRAKADABRA... ablaufen. Wie groß ist die Einsparung durch die Codierung in diesem Fall?
- b) Gehen Sie die Schritte durch, die bei der Dekomprimierung von [8.15.11.21.19.16.28.30.27.29.31.33.19.35.30.32.36.] ablaufen.
- c) **Implementierung des LZW Algorithmus in Java**
 - (i) Schreiben Sie eine Funktion *compress(String)* die eine Zeichenkette nimmt und als Liste von Zahlen (Zahlen mit '.' getrennt in einer Zeichenkette) entsprechend dem LZW-Algorithmus zurückgibt.
 - (ii) Schreiben Sie eine Funktion *uncompress(String)* die eine so kodierte Liste wieder in eine Zeichenkette umwandelt.
 - (iii) Fassen Sie die Funktionen zu einem Programm zusammen, das entsprechende Zeichenketten kodieren bzw. dekodieren kann.

Aufgabe 11 **Prüfpolynome und CRC-Berechnung**

Für die nachfolgenden Programmieraufgaben kann die OCaml-Funktion `List.length` verwendet werden.

- a) Gegeben sei das Generator Polynom $G_1(x) = x^{16} + x^{12} + x^5 + 1$. Bestimmen Sie die Prüfsumme der Nachricht 100010000011111 mittels $G_1(x)$ nach dem CRC-Verfahren.
- b) Ein Empfänger erhält eine mit $G_2(x) = x^3 + x + 1$ gesicherte CRC-Nachricht 1101110111010. Überprüfen Sie, ob ein Übertragungsfehler eingetreten ist.
- c) Schreiben Sie ein Funktion `append_zeroes`, welche an eine übergeben Liste von 0-en und 1-en (in der Form `[1; 0; 1]`) eine festgelegte Anzahl von 0-en anhängt.
- d) Schreiben Sie ein Funktion `xor_begin`, welche die Anfänge zweier übergebener Listen über XOR miteinander verknüpft und die verbleibenden Zeichen der längeren Liste anhängt.
- e) Schreiben Sie ein Funktion `crc`, welche aus einem übergebenen Datenwort und einem Generatorpolynom das Codewort berechnet. Berechnen Sie hierfür zuerst den Rest und hängen Sie diesen anschließend an das Datenwort an.