

Datenkompression

Sommersemester 2018

Übungsblatt 1

Prof. Ohlebusch

Institut für Theoretische Informatik

Ausgegeben am 26.04.2018

Besprechung am 03.05.2018

Aufgabe 1.1

Wir betrachten die folgenden Mengen von Codewörtern:

$$\begin{aligned} M_1 &:= \{01, 10, 000, 011, 100, 111\}, & M_2 &:= \{0, 10, 11, 101\}, \\ M_3 &:= \{00, 01, 11, 100, 101\}, & M_4 &:= \{00, 11, 010, 0100\}. \end{aligned}$$

- Welche Mengen erfüllen die Kraft-McMillan-Ungleichung?
- Welche Mengen sind Bildbereich eines eindeutig dekodierbaren Codes, und warum?
- Geben Sie für jede Menge M_i einen Präfixcode an, welcher dieselben Codewortlänge wie M_i hat. Für welche Mengen ist dies nicht möglich?

Aufgabe 1.2

Auf dem Alphabet $\{d, i, l, m, n, r, u\}$ sei eine Wahrscheinlichkeitsfunktion P definiert:

x	d	i	l	m	n	r	u
$P(x)$	1/24	1/24	1/6	7/24	1/12	1/24	1/3

- Bestimmen Sie einen Präfixcode nach dem Shannon-Verfahren.
- Bestimmen Sie einen Präfixcode nach dem Shannon-Fano-Verfahren (Variante 2 in Aufgabe 1.3).
- Bestimmen Sie einen Präfixcode nach dem Huffman-Verfahren.
- Berechnen Sie die erwarteten Codewortlängen der Codes aus den Teilaufgaben a) bis c).

Aufgabe 1.3

Es gibt zwei Varianten des Shannon-Fano-Algorithmus. In der ersten Version wird die Liste der Symbole in zwei Teile $M_1 = (c_1, \dots, c_k)$ und $M_2 = (c_{k+1}, \dots, c_n)$ gespalten, sodass $|P(M_1) - P(M_2)|$ minimal ist. In der zweiten Version werden Teilmengen \tilde{M}_1, \tilde{M}_2 betrachtet. Dabei gilt $\tilde{M}_1 \cup \tilde{M}_2 = \{c_1, c_2, \dots, c_n\}$ und $\tilde{M}_1 \cap \tilde{M}_2 = \emptyset$. \tilde{M}_1 und \tilde{M}_2 sollen möglichst „gleichmäßig“ gewählt werden, also $P(\tilde{M}_1) \approx P(\tilde{M}_2)$. Wir betrachten eine Variante des Algorithmus, die unter *allen* Partitionierungen stets eine „gleichmäßigste“ wählt.

- Geben Sie eine Wahrscheinlichkeitsfunktion P an, bei der die zweite Version Algorithmus einen Code mit größerer erwarteter Codewortlänge generiert als die erste Version des Shannon-Fano-Algorithmus.
- (*schwieriger!*) Geben Sie ein Beispiel an, bei dem die zweite Version des Algorithmus einen Code mit kleinerer erwarteter Codewortlänge generiert als die erste Version.