Informationstheorie, FS 2021 Prof. Joachim M. Buhmann
ETH Zürich, Dept. Informatik Fabian Laumer

Lösungen zu Übungsblatt 4

4.1

Gegeben seien die Mengen $\{0, 10, 11\}$, $\{00, 01, 10, 110\}$ und $\{01, 10\}$. Geben Sie jeweils eine Zufallsvariable X samt Verteilung an, so dass die Menge die Menge der Codewörter eines Huffman-Codes für X ist, oder begründen Sie, dass es keine solche gibt.

Eine mögliche Verteilung für $\{0, 10, 11\}$ ist $\mathbf{p} = (0.5, 0.3, 0.2)$. Weder $\{00, 01, 10, 110\}$ noch $\{01, 10\}$ sind die Mengen der Codewörter eines Huffman-Codes, denn unter den längsten Codewörtern eines Huffman-Codes gibt es immer zwei, die sich nur im letzten Bit unterscheiden (ausserdem sieht man zumindest bei der letzten Menge sofort, dass sie nicht optimal ist, was die erwartete Länge angeht).

4.2

Entscheiden Sie für die Tupel $(1, 2, 2)$, $(2, 2, 3, 3)$ und $(1, 3, 3, 3, 4, 5)$ jeweils, ob sie die Wortlängen eines Huffman-Codes sind. Können Sie ein allgemeines Kriterium dafür angegeben, dass ein gegebenes Tupel $(l_1, \dots, l_n)$ die Wortlängen eines Huffman-Codes darstellt?

$(1, 2, 2)$ sind die Längen des Huffman-Codes $\{0, 10, 11\}$ z.B. für $\mathbf{p} = (0.5, 0.3, 0.2)$. Für $(2, 2, 3, 3)$ und $(1, 3, 3, 3, 4, 5)$ gibt es keinen Huffman-Code mit diesen Wortlängen. Im letzten Fall sieht man das direkt daran, dass es ein Code mit diesen Wortlängen ein eindeutiges längstes Wort haben müsste, was bei einem Huffman-Code nicht der Fall ist. Wären $(2, 2, 3, 3)$ die Wortlängen eines Huffman-Codes, dann müsste das auch für $(2, 2, 2)$ der Fall sein (aufgrund der rekursiven Definition von Huffman-Codes), und ebenfalls für $(2, 1)$; im letzten Fall gibt es aber wiederum ein eindeutiges längstes Wort.

4.3

Geben Sie ein Beispiel für eine Verteilung $\mathbf{p} = (p_1, \dots, p_5)$ an, für die es mindestens zwei optimale Codes gibt, für die sich die (nach Länge geordneten) Tupel der Codewortlängen unterscheiden.

Z.B. für $\mathbf{p} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{6}, \frac{1}{9}, \frac{1}{18})$ sind sowohl $(2, 2, 2, 3, 3)$ und $(1, 2, 3, 4, 4)$ Wortlängen möglicher Huffman-Codes. Die erwartete Länge ist jeweils gleich $\frac{39}{18}$.

4.4

Beweisen Sie folgende Behauptung aus der Vorlesung (siehe Beweis der Optimalität von Huffman-Codes): Sei X eine Zufallsvariable X mit der Verteilung $\mathbf{p} = (p_1, \dots, p_n)$, wobei $p_i = P_X(x_i)$ für $x_i \in X(\Omega)$. Dann gibt es einen optimalen präfixfreien Code C^* für X mit folgender Eigenschaft: Es gibt $x_{i_0} \neq x_{i_1}$ mit kleinster Wahrscheinlichkeit (d.h. $p_{i_0} + p_{i_1} \leq p_j + p_k$ für alle $j \neq k$), so dass sich $C^*(x_{i_0})$ und $C^*(x_{i_1})$ nur im letzten Bit unterscheiden.

Sei zunächst C' irgendein optimaler präfixfreier Code für X. Dann ist klar, dass es für jedes Codewort w_0 mit maximaler Länge ein weiteres Codewort w_1 geben muss, so dass sich w_0 und w_1 nur im letzten Bit unterscheiden (sonst könnte man w_0 und w_1 um ein Bit kürzen und hätte damit einen Code mit geringerer Länge als C', im Widerspruch zur Optimalität). Ausserdem ist klar, dass ein Paar $x_{i_0} \neq x_{i_1}$ mit kleinster Wahrscheinlichkeit $p_{i_0} + p_{i_1}$ existiert, so dass $C'(x_{i_0})$ und $C'(x_{i_1})$ maximale Länge haben: Wäre nämlich für alle x_i, x_j, für die $C'(x_i), C'(x_j)$ maximale Länge haben, die Wahrscheinlichkeit $p_i + p_j < p_{i_0} + p_{i_1}$, dann könnte man durch Vertauschen von Codewörtern wiederum einen kürzeren Code erzeugen. Indem wir C' durch Vertauschen von Codewörtern mit maximaler Länge so modifizieren, dass x_{i_0} auf w_0 und x_{i_1} auf w_1 abgebildet wird (wobei sich w_0 und w_1 nur im letzten Bit unterscheiden), erhalten wir einen Code C^* mit der geforderten Eigenschaft.

4.5

Finden Sie für die Zufallsvariable X mit der Verteilung

X	x_1	x_2	x_3	x_4	x_5	x_6	x_7
P_X	0.49	0.26	0.12	0.04	0.04	0.03	0.02

einen (binären) Huffman-Code und berechnen Sie seine erwartete Länge.

Ein möglicher Huffman-Code für X ist gegeben durch

X	x_1	x_2	x_3	x_4	x_5	x_6	x_7
C	0	10	110	11100	11101	11110	11111

Die erwartete Länge ist $L_C = 0.49 + 0.26 \cdot 2 + 0.12 \cdot 3 + 0.13 \cdot 5 = 2.02$ Bits pro Symbol.