Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. Fabian Kuhn
P. Bamberger, Y. Maus, P. Schneider

Informatik 2 - Sommersemester 2018
Musterlösung Übungsblatt 11*

Abgabe: Freitag, 27. Juli, 14:00 Uhr

Aufgabe 1: Huffman-Kodierung *(12 Punkte)*

Sei $\Sigma = \{a, b, c, d\}$ ein Alphabet. Die Wahrscheinlichkeit p_x (Vorlesung: Frequenz) für ein Zeichen $x \in \Sigma$, sei wie folgt: $p_a = 0.4, p_b = 0.3, p_c = 0.2, p_d = 0.1$.

(a) Geben Sie die Entropie der gegebenen Wahrscheinlichkeitsverteilung mit zwei Nachkommastellen Genauigkeit an (schneiden Sie die anderen Nachkommastellen ab). *(2 Punkte)*

(b) Berechnen Sie eine Huffman-Kodierung von Σ und anschließend die *erwartete Länge der Kodierung* eines zufälligen Zeichens bzgl. dieser Wahrscheinlichkeitsverteilung. *(3 Punkte)*

Sei $\Sigma^2 = \{aa, ab, \ldots, dd\}$ das Alphabet der Tupel der Zeichen aus Σ. Wir nehmen an, dass die obige Wahrscheinlichkeitsverteilung stochastisch unabhängig ist. Deshalb kann die Wahrscheinlichkeit, dass man nach obiger Verteilung zwei Zeichen $xy \in \Sigma^2$ erhält, als $p_{xy} := p_x \cdot p_y$ ausgedrückt werden.

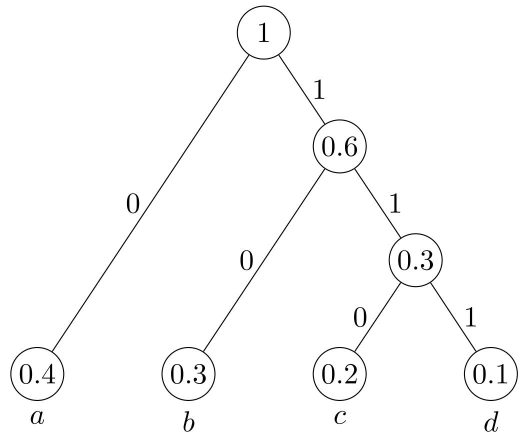
(c) Geben Sie die Entropie der Wahrscheinlichkeitsverteilung p_{xy} für $xy \in \Sigma^2$ mit zwei Nachkommastellen Genauigkeit an. Teilen Sie das Ergebnis durch 2. *(2 Punkte)*

(d) Berechnen Sie eine Huffman-Kodierung von Σ^2 bzgl. p_{xy}. Berechnen Sie die *erwartete Länge der Kodierung* eines zufälligen Tupels (verteilt mit p_{xy}). Teilen Sie das Ergebnis durch 2. *(5 Punkte)*

Musterlösung

(a) Es ist $H(p_x) = \sum_{x \in \Sigma} -p_x \log p_x \approx 1.84$.

(b) Wir führen den gierigen Ansatz aus der Vorlesung (von Hand) durch, um einen Binärbaum zu konstruieren, aus der wir die Huffman-Kodierung ablesen können, indem wir die Bits von der Wurzel zum entsprechenden Blatt notieren.



*Dies ist ein Bonusübungsblatt dessen Punkte zwar angerechnet werden aber nicht in die Summe der maximal erreichbaren Punkte einfließen.

Wir erhalten die folgende präfixfreie Kodierung: $a = 0, b = 10, c = 110, d = 111$. Wenn wir die Codelängen der Zeichen mit den Wahrscheinlichkeiten multipliziert aufaddieren erhalten wir die erwartete Codelänge

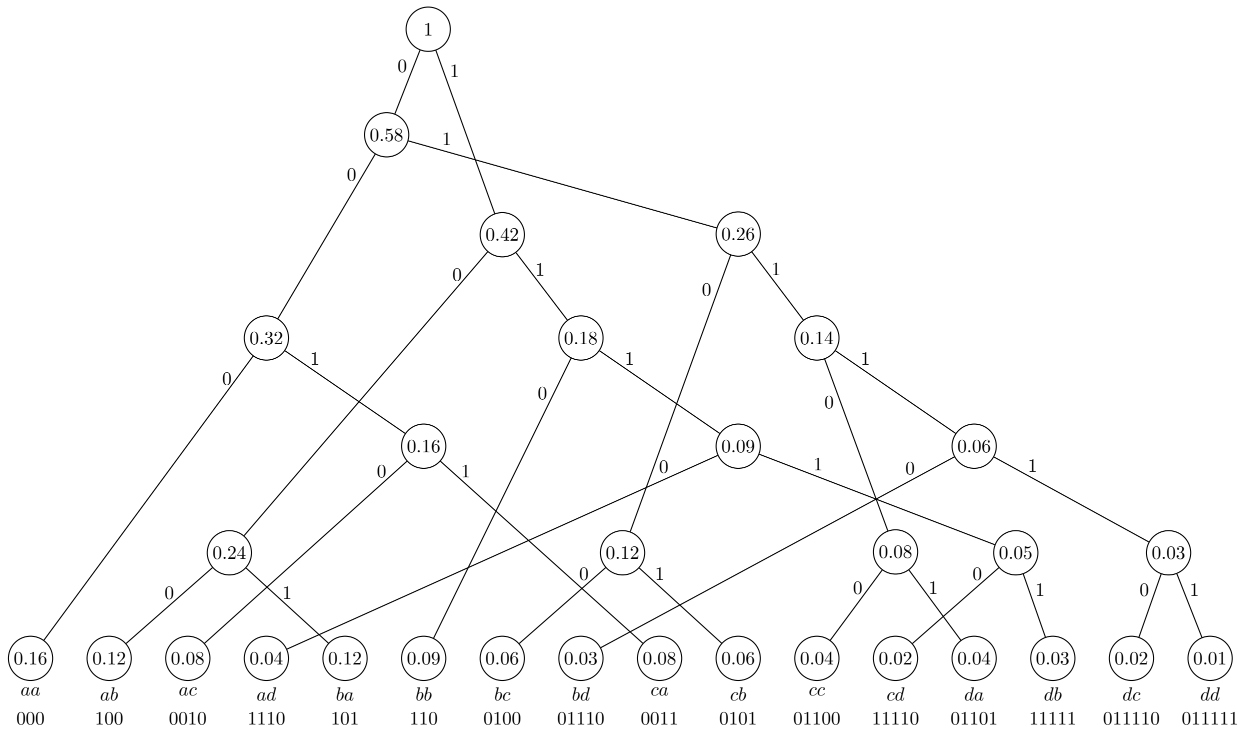
$$\mathbb{E}(\text{Codelänge}) = \sum_{x \in \Sigma} p_x \cdot |\text{Kodierung}(x)| = 1 \cdot 0.4 + 2 \cdot 0.3 + 3 \cdot 0.2 + 3 \cdot 0.1 = 1.9.$$

Das ist zwar nahe an der Entropie (die eine untere Schranke für die gegebene Verteilung darstellt) aber es geht noch ein bisschen besser.

(c) Es ist $H(p_{xy}) = \sum_{xy \in \Sigma^2} -p_{xy} \log p_{xy} \approx 3.69$ und $H(p_{xy})/2 \approx 1.84 \approx H(p_x)$. Die Entropie pro Zeichen ändert sich also nicht wenn wir uns Blöcke von Buchstaben ansehen.

Bemerkung: Wenn man zusätzliches Wissen hat, z.B. dass die Wahrscheinlichkeiten der einzelnen Zeichen korrelieren, erhält man eine niedrigere Entropie. Kompressionsverfahren machen sich das oft zunutze.

(d) Wir berechnen die Huffman-Kodierung von Σ^2 bzgl. p_{xy} wie zuvor.



Die erwartete Länge der Kodierung ist dann

$$\mathbb{E}(\text{Codelänge}) = \sum_{x{<}\in\Sigma^2} p_{xy} \cdot |\text{Kodierung}(xy)| = 3.73.$$

Halbiert ergibt sich der Erwartungswert pro Zeichen: $\mathbb{E}(\text{Codelänge})/2 \approx 1.87$. Das ist näher an der Entropie als die erwartete Kodierungslänge der Huffman-Kodierung bei der wir Zeichen einzeln kodierten.

Aufgabe 2: Gieriger Dieb *(6 Punkte)*

Ein Dieb in einem Juwelenladen möchte den Gesamtwert seiner gestohlenen Schmuckstücke maximieren. Er kann dabei höchstens ein Gewicht von $W > 0$ tragen und die n ausgestellten Schmuckstücke haben Gewichte $w_1, \ldots, w_n \leqslant W$ und Werte $v_1, \ldots, v_n \geqslant 0$.

(a) Formulieren Sie einen einfachen Greedy Algorithmus der eine Auswahl von Schmuckstücken für den Dieb in $\mathcal{O}(n \log n)$ trifft. *(3 Punkte)*

(b) Geben Sie ein konkretes Beispiel von Schmuckstücken an, in dem der Gesamtwert der Greedy Wahl einen Anteil von höchstens $0 < \varepsilon < 1$ des Gesamtwertes der optimalen Wahl von Schmuckstücken erreicht. *(3 Punkte)*

Hinweis: Zeigen Sie also, dass der Gesamtwert der Greedy Wahl von Schmuckstücken (relativ gesehen) beliebig viel kleiner als der Gesamtwert der optimalen Wahl wird.

Musterlösung

(a)

Algorithm 1: `Greedy Thief I`$(W, w_1, \ldots, w_n \leqslant W, v_1, \ldots, v_n \geqslant 0)$

```
Sort jewellery items (descending) by value-weight ratio v_i/w_i in a list L
while W > 0 do
    Steal item i which maximizes v_i/w_i among the remaining items (L.popFront)
    W ← W − w_i
```

Algorithm 2: `Greedy Thief II`$(W, w_1, \ldots, w_n \leqslant W, v_1, \ldots, v_n \geqslant 0)$

```
Sort jewellery items (descending) by value v_i in a list L
while W > 0 do
    Steal most valueable item i (L.popFront)
    W ← W − w_i
```

(b) `Greedy Thief I`: Wähle $n = 2$, $W = 1$, $v_1 = \varepsilon, w_1 = \delta < \varepsilon$ und $v_2 = 1, w_2 = 1$. Der gierige Dieb packt zuerst Schmuckstück 1 mit $v_1/w_1 = \varepsilon/\delta > 1$ ein. Da er dann nichts mehr tragen kann ist er fertig und hat einen Wert von ε gestohlen. Er hätte aber stattdessen Schmuckstück 2 stehlen können mit Wert 1.

`Greedy Thief II`: Wähle $n = \lceil 2/\varepsilon \rceil + 1$, $W = 1$, $v_1 = \varepsilon, w_1 = 1$ und alle restlichen Schmuckstücke $i \geqslant 2$: $v_i = \varepsilon/2, w_i = \varepsilon/4$. Der gierige Dieb packt zuerst Schmuckstück 1 ein. Da er dann nichts mehr tragen kann ist er fertig und hat einen Wert von ε gestohlen. Er hätte aber stattdessen mindestens $\lceil 2/\varepsilon \rceil$ Schmuckstücke mit Wert $\varepsilon/2$ stehlen können was einen Gesamtwert von mindestens 1 hat.

Aufgabe 3: Graphen färben *(12 Punkte)*

Sei $G = (V, E)$ ein ungerichteter Graph mit Knotenmenge V, Kantenmenge E und maximalem Grad Δ ($\Delta = \max\{\deg(v) \mid v \in V\}$). Eine c-Färbung ($c \in \mathbb{N}$) von G ist eine Abbildung $\phi : V \to \{1, \ldots, c\}$, so dass benachbarte Knoten verschiedene Farben haben, d.h. $\phi(u) \neq \phi(v)$ für alle $\{u, v\} \in E$.
Die chromatische Zahl $\chi(G)$ von G, ist die minimale Anzahl Farben, mit der sich G färben lässt.

(a) Formulieren Sie einen Greedy-Algorithmus zur Färbung eines Graphen. Der Algorithmus soll maximal $\Delta + 1$ Farben verwenden. Begründen Sie die Korrektheit des Algorithmus sowie die obere Schranke $\Delta + 1$ für die Anzahl an Farben. *(4 Punkte)*

(b) Beschreiben Sie eine Klasse von Graphen, für die $\chi(G) = \Delta + 1$ gilt. *(2 Punkte)*

(c) Beschreiben Sie eine Klasse von Graphen, bei der das Verhältnis $(\Delta + 1)/\chi(G)$ beliebig groß wird. *(2 Punkte)*

Nun möchte man zusätzlich auch die Kanten von G färben, d.h. ein $c \in \mathbb{N}$ und eine Abbildung $\phi : V \cup E \to \{1, \ldots, c\}$ finden, so dass

1. Für alle $\{u, v\} \in E$ gilt $\phi(u) \neq \phi(v)$.
2. Für alle adjazenten Kanten $e, e' \in E$ (d.h. $e \neq e'$ teilen einen Endpunkt) gilt $\phi(e) \neq \phi(e')$.

3. Für alle $u \in V, e \in E$ mit $u \in e$ gilt $\phi(u) \neq \phi(e)$.

(d) Formulieren Sie einen Greedy-Algorithmus, um die Kanten und Knoten von G mit $2\Delta + 1$ Farben zu färben (so dass die Bedingungen 1, 2, 3 erfüllt sind). Begründen Sie die Korrektheit und die maximale Anzahl Farben. *(4 Punkte)*

Musterlösung

(a) **Algorithm 3:** `Greedy Sequential I`

while *there is an uncolored vertex v* **do**
  color v with the minimal color (number) that does not conflict with the already colored neighbors

Da jeder Knoten höchstens Δ Nachbarn hat, kann man immer eine Farbe in $\{1, \dots, \Delta+1\}$ wählen, die zu keinem Konflikt führt. Am Ende jedes Schleifendurchlaufs gilt also: Alle bereits gefärbten Knoten haben eine Farbe in $\{1, \dots, \Delta + 1\}$ und es gibt keine Konflikte zwischen gefärbten, benachbarten Knoten. Die Schleife bricht ab, wenn es keine ungefärbten Knoten mehr gibt.

(b) Cliquen (Graphen, in denen es zwischen allen Knotenpaaren eine Kante gibt) und ungerade Kreise.

(c) Bäume sind 2-färbbar, können aber beliebig großen Grad haben.

(d) **Algorithm 4:** `Greedy Sequential II`

while *there is an uncolored vertex v or edge e* **do**
  color v or e with the minimal color (number) that does not violate conditions 1, 2, 3

Jeder Knoten hat maximal Δ benachbarte Knoten und Δ inzidente Kanten. Wenn man also einen Knoten färbt, sind maximal 2Δ Farben 'verboten'. Eine Kante hat zwei inzidente Knoten und maximal $2(\Delta - 1)$ adjazente Kanten, also gibt es bei der Färbung einer Kante maximal 2Δ verbotene Farben. Bei der Färbung eines Knoten oder einer Kante gibt es also immer eine Farbe in $\{1, \dots, 2\Delta + 1\}$, die zu keinem Konflikt führt.