Grundbegriffe der Informatik — Aufgabenblatt 6
Lösungsvorschläge

Tutorium Nr.:		Tutor*in:	
Matr.nr. 1:			
Nach-,Vorname 1:		,	
Matr.nr. 2:			
Nach-,Vorname 2:		,	

Ausgabe: 26. Nobember 2021, 12:00 Uhr

Abgabe: 03. Dezember 2021, 12:30 Uhr
in dem Holzkasten neben Raum -119
im UG des Info-Gebäudes (50.34)

Lösungen werden nur korrigiert, wenn sie
- handschriftlich erstellt sind (Tablet-Ausdruck erlaubt) und
- mit dieser Seite als Deckblatt
- in der oberen **linken** Ecke zusammengeheftet

rechtzeitig abgegeben werden.

Abgaberegeln für Teilnehmer der Online-Tutorien:
- handschriftlich erstellt (lesbare Fotos akzeptiert)
- **rechtzeitig**, mit diesem Deckblatt in **genau einer** PDF-Datei
- direkt an den entsprechenden Tutor abgeben.

---

*Von Tutor*in auszufüllen:* erreichte Punkte

Blatt 6:	/ 17	Blätter 1 – 6, Stud. 1:	/ 116,5
		Blätter 1 – 6, Stud. 2:	/ 116,5

Aufgabe 6.1 (0,5 + 2 + 1 + 2 + 1 + 1 + 1,5 = 9 Punkte)
Sei A das Alphabet der Zeichen $\{\text{␣}, \texttt{a}, \texttt{b}, ..., \texttt{z}, .\}$. Eine vollständige Auflistung der Zeichen aus A finden Sie in der unten aufgeführten Tabelle in Zeilen eins und drei. Sei außerdem eine Abbildung $val : A \to \mathbb{Z}_{28}$ definiert, deren Wert in Zeile zwei, bzw. vier der unteren Tabelle für jedes $x \in A$ explizit aufgelistet ist. Ferner sei die Umkehrfunktion von val als $val^{-1} : \mathbb{Z}_{28} \to A$ definiert. val^{-1} bildet jede Zahl $n \in \mathbb{Z}_{28}$ auf das entsprechende $x \in A$, mit $val(x) = n$ ab.

$x \in A$	␣	a	b	c	d	e	f	g	h	i	j	k	l	m
$val(x)$	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$x \in A$	n	o	p	q	r	s	t	u	v	w	x	y	z	.
$val(x)$	14	15	16	17	18	19	20	21	22	23	24	25	26	27

Damit die Daten der B.I.R.Ds auf dem Weg zu seiner geheimen Zentrale vertraulich bleiben, hat sich Dr. Meta eine Verschlüsselungsfunktion $Enc : A^* \times A^+ \to A^*$ überlegt. Seit seiner Promotion ist er allerdings des Definierens von Funktionen überdrüssig. Da Sie ihre Fähigkeiten auf diesem Gebiet bereits zum wiederholten Male unter Beweis gestellt haben, dürfen Sie diese Aufgabe für ihn übernehmen. Die Verschlüsselung soll eine Nachricht $w \in A^*$, sowie einen Schlüssel $k \in A^+$ wie folgt verarbeiten:
Zum Verschlüsseln eines Zeichens der Nachricht wird es mit einem Zeichen des Schlüsselwortes „aufaddiert". Hierzu wird zunächst mittels val der Zahlenwert der jeweiligen Zeichen berechnet. Diese Werte werden addiert und das Ergebnis wird mittels val^{-1} wieder auf ein Zeichen aus A abgebildet. Die verschlüsselte Nachricht ergibt sich als Konkatenation aller jeweils auf diese Weise verschlüsselten Zeichen. Hierbei wird für alle $i \in \mathbb{Z}_{|w|}$ $w(i)$ unter Verwendung von $k(i)$ verschlüsselt.
Die folgenden Teilaufgaben führen Sie schrittweise zur Definition von Enc:

a) Damit die oben beschriebene Rückberechnung eines Zeichens mit val^{-1} für alle Ergebnisse s einer Addition zweier Zeichenwerte stets funktioniert, muss sichergestellt werden, dass stets $s \in \mathbb{Z}_{28}$ gilt. Welche in der Vorlesung vorgestellte Operation kann verwendet werden um dies sicherzustellen?

b) Definieren Sie eine Funktion $first_n : A^m \to A^n$, die für beliebige $m, n \in \mathbb{N}_0$, mit $m \geq n$ und alle Wörter $w \in A^m$ das Präfix der Länge n von w berechnet.
Tipp: Nutzen Sie eine Hilfsfunktion.
Tipp: Setzen Sie Funktionsargumente ähnlich wie in Aufgabe 5.4 illustriert ein.

c) Damit die gesamte Nachricht w zeichenweise verschlüsselt werden kann, muss für den Schlüssel k stets $|w| \leq |k|$ gelten. Damit sich Dr. Meta keine langen Schlüsselwörter merken muss, soll für den Fall $|k| < |w|$ die Zeichen des Schlüsselwortes Reihenfolge getreu „recycelt" werden. Definieren Sie die Funktion $pad_n : A^+ \to A^n$, die für ein konkretes $n \in \mathbb{N}_0$ und Eingaben $w \in A^+$ im Fall $|w| \geq n$ das Präfix der Länge n und im Fall $|w| < n$ eine Reihenfolge getreue Wiederholung der Zeichen in w mit insgesamt Länge n ausgibt.
Bsp: $pad_{12}(\texttt{dr␣meta}) = \texttt{dr␣metadr␣me}$ $pad_4(\texttt{metamorphose}) = \texttt{meta}$

d) Definieren Sie die Funktion Enc.

e) Wie viele Rechtsinverse hat Enc? Beweisen Sie.

f) Ist Enc eine Codierung im Sinne der Vorlesung oder nicht? Beweisen Sie.

g) Die Folgende Nachricht wurde von den B.I.R.D-Drohnen gesendet:
„`wnlgbnovtezjznqquqtaoneejfmifxfb␣nez.j␣iem.`"
Sie ist mit Dr. Metas Universalpasswort „`genial␣boese`" verschlüsselt. Entschlüsseln Sie sie und geben sie die originale Nachricht an. Rechenweg **nicht** gefordert.

Lösung 6.1

a) Die Summe s kann **mod** 28 gerechnet werden.

b) Für alle $w \in A^m, w' \in A^*, x \in A$:

$$first_n(w) = helper(w, n)$$
$$helper(w', 0) = \varepsilon$$
$$helper(xw', n) = x \cdot helper(w', n-1)$$

c) $pad_n(w) = first_n(w^n)$, für alle $w \in A^+$

d) Für alle $k \in A^+, w, w', k' \in A^*, x, y \in A$:

$$Enc(w, k) = enc(w, pad_{|w|}(k))$$
$$enc(\varepsilon, k) = \varepsilon$$
$$enc(xw', yk') = val^{-1}((val(x) +_{28} val(y))) \cdot enc(w', k')$$

Anmerkung: $a +_{28} b = (a + b)$ **mod** 28

e) Unendlich viele. Schema aus dem sich beliebig Viele erzeugen lassen:
$Enc^{-r} : A^* \to A^* \times A^+$, mit $Enc^{-r}(w) = (w, \text{␣}^i)$ für alle $w \in A^*$ und beliebige $i \in \mathbb{N}_+$, also für unendlich viele i je eine Enc^{-r}.

f) *Enc* ist keine Codierung im Sinne der Vorlesung, da die Funktion nicht injektiv ist. Es wäre allerdings eine Übersetzung. (Ein Grund bereits ausreichend.)
Beweis für Nicht-Injektivität: $Enc(\texttt{a}, \texttt{c}) = \texttt{d} = Enc(\texttt{c}, \texttt{a})$.

g) „`pizzabote␣gesichtet.␣in␣ca.␣elf␣minuten␣da.`“

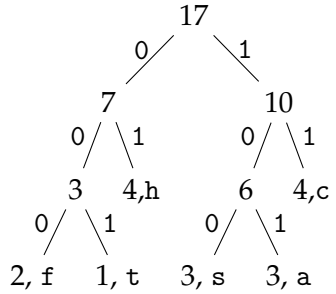
Aufgabe 6.2 (2 [+ 3] = 2 [+ 3] Punkte)

a) Bilden Sie den Huffman-Baum zum Wort $w =$ `fachschaftsschach`.
Beginnen Sie mit folgender Zeichenanordnung der Blätter:
`f t h s a c`
Bilden Sie einen Baum ohne kreuzende Kanten.
Geben Sie außerdem die Huffman-Codierung jedes Zeichens an.

b) Wie kann die Huffman-Codierung angepasst werden um möglichst kurze Codewörter eines ternären Alphabets $\{0, 1, 2\}$ zu erzeugen?
Orientieren Sie sich an der Beschreibung zur Konstruktion des Huffman-Codes in Kapitel 8 (Folien 62-64). Beschreiben Sie ihre Anpassung möglichst präzise.
Diese Aufgabe gibt 3 Bonuspunkte.

Lösung 6.2

a) Huffman-Baum:



$H(\texttt{f}) = 000$, $H(\texttt{t}) = 001$, $H(\texttt{h}) = 01$,
$H(\texttt{s}) = 100$, $H(\texttt{a}) = 101$, $H(\texttt{c}) = 11$

b) Wir verändern die Initialisierung wie folgt:
- Zunächst bilden wir M_0 wie gewohnt

- Falls $|M_0|$ gerade, also $|M_0|$ **mod** $2 = 0$, ergänzen wir $M_0' = M_0 \cup \{(0, \perp)\}$, wobei o.B.d.A. $\perp \notin A$,
- andernfalls: $M_0' = M_0$

Der Algorithmus wird nun auf M_0' gestartet.
Anmerkung: Diese kleine Änderung ist notwendig um die Minimalität der Codewörter sicher zu stellen. Im Fall $|M_0|$ werden sonst am Ende zwei Kanten zur Wurzel führen wodurch noch Kapazität für kürzere h(x) durch Nutzung der „vollen Bandbreite" der Wurzel ungenutzt bleibt. Zum Verständnis können Sie den Algorithmus mal ohne diesen Initialisierungsschritt am Wort aus Teilaufgabe a) ausprobieren.
Nun wird zwar ein $H(\perp)$ mitberechnet, dieses wird jedoch später einfach ignoriert.

Ändere den Iterationsschritt wie folgt:

- Wähle in M_i' bis zu **drei** Paare (k_1, X_1), (k_2, X_2), (k_3, X_3) mit den kleinsten Häufigkeiten und ersetze diese Paare durch $(k_1 + k_2 + k_3, X_1 \cup X_2 \cup X_3)$.
- $M_{i+1}' := (M_i \setminus \{(k_1, X_1), (k_2, X_2), (k_3, X_3)\}) \cup \{(k_1 + k_2 + k_3, X_1 \cup X_2 \cup X_3)\}$

und im **Graphen**:

- neuer Knoten mit Häufigkeit $k_1 + k_2 + k_3$.
- Kanten zu Knoten wie folgt:
  - (k_1, X_1) mit Beschriftung: `0`
  - (k_2, X_2) mit Beschriftung: `1`
  - (k_3, X_3) mit Beschriftung: `2`

Aufgabe 6.3 (0,5 + 0,5 + 0,5 + 0,5 + 4 = 6 Punkte)
Sei A ein beliebiges Alphabet mit $|A| > 2$. Definieren Sie die in den folgenden Teilaufgaben die in natürlicher Sprache spezifizierten Dinge **formal korrekt**. Nutzen sie hierfür keine natürliche Sprache, mit Ausnahme folgender Begriffe:
„für alle" „es gibt (ein)", bzw. „gibt es (ein)"

a) Für ein Wort $w \in A^*$ definieren wir das von w induzierte Teilalphabet A_w als die Menge aller Zeichen die sowohl in A als auch in w enthalten sind. Definieren Sie A_w.
b) Eine Verallgemeinerung der Zeichenhäufigkeitsfunktion $N_x(w)$ auf ganze Mengen M, sodass $N_M(w)$ die gesamte Häufigkeit aller Zeichen M in w ergibt. Definieren Sie $N_M(w)$.
c) Für zwei beliebige Mengen von Zeichen, die in w enthalten sind, gilt, dass die Elemente der größeren Menge insgesamt häufiger in w vorkommen. Definieren Sie diesen Zusammenhang für entsprechende Mengen aus A_w.
d) Sei A ein beliebiges Alphabet, $w \in A^*$ ein Wort und $H(w)$ eine Huffman-Codierung von w. Geben Sie eine geschlossene Formel zur Berechnung von $|H(w)|$ ohne Verwendung von $w(i)$ - oder einer äquivalenten Funktion an.

Sei im Folgenden $w \in A^*$ ein Wort, dass die in Aufgabenteil c) spezifizierte Bedingung erfüllt und H eine Huffman-Codierung von w.

e) Gilt für alle $x_1, x_2 \in A_w$: $|H(x_1)| = |H(x_2)|$?
   - Falls ja: Beweisen Sie.
   - Falls nein: Fügen Sie weitere Bedingungen zu w hinzu, sodass die Behauptung gilt. Formulieren Sie die Bedingungen ebenfalls nach den Kriterien aus a) - c). Beweisen Sie, dass die Behauptung aus der Gesamtmenge aller Bedingungen folgt.

   Hinweis: Überlegen Sie sich, wie ein Huffman-Baum aussehen muss, damit die

Behauptung erfüllt ist.

Lösung 6.3

a) $A_w := \{x \in A \mid \text{es gibt ein } i \in \mathbb{Z}_{|w|} : x = w(i)\}$

b) $N_M(w) := \sum_{x \in M} N_x(w)$

c) Für alle $M_1, M_2 \subseteq A_w$: $|M_1| > |M_2| \Rightarrow N_{M_1} > N_{M_2}$

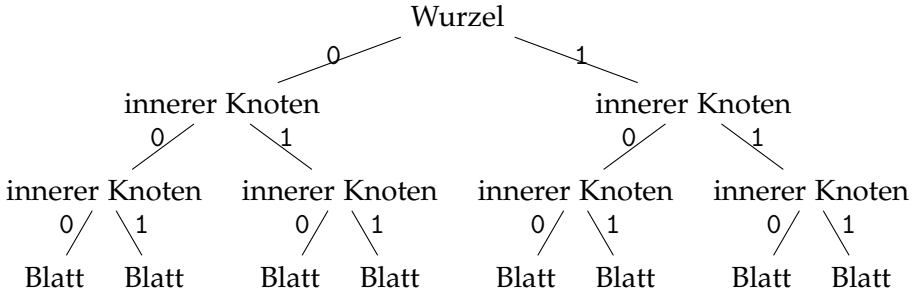
d) $|H(w)| = \sum\limits_{x \in A_w} (N_x(w) \cdot |H(x)|)$

e) Die Bedingung reicht **nicht** aus um $x_1, x_2 \in A_w$: $|H(x_1)| = |H(x_2)|$ für alle $x_1, x_2 \in A_w$ sicherzustellen. Allerdings zusammen mit folgender Bedingung: $|A_w| = 2^k, k \in \mathbb{N}_+$.

Beweis:

Im Folgenden bezeichnen wir die „Spitze“ des Huffman-Baumes von w als „Wurzel“. Die Knoten des Baumes unterteilen wir in „Blattknoten“ (Knoten die für die Häufigkeit genau eines Zeichens $x \in A_w$ stehen) und „innere Knoten“ (Andere).

Wir halten fest, dass das Codewort jedes Zeichens $x \in A_w$ für jede Kante auf dem „Weg“ von der Wurzel bis zu „seinem“ Blattknoten genau um ein Zeichen verlängert wird. Folglich sind die die Codierungen aller $x \in A_w$ sind genau dann gleich lang, wenn auf diesen Wegen für alle $x \in A_w$ gleich viele Kanten passiert werden. Dies ist genau dann der Fall, wenn der Baum die hier skizzierte Struktur hat:



Wobei der Baum natürlich beliebig „tief“ sein kann, also eine beliebige 2er-Potenz als Blattanzahl haben kann.

Damit bei der Ausführung des Huffman-Algorithmus diese Baumstruktur entsteht, müssen 2 Bedingungen erfüllt sein:

1. Die Blätter müssen „gleichmäßig“ zu inneren Knoten verbunden werden, welche wiederum gleichmäßig verbunden werden ..., bis hoch zur Wurzel.
2. Die Anzahl der Blätter muss passen, nämlich genau eine 2er-Potenz sein.

Die in c) vorgestellte Bedingung:
Für alle $M_1, M_2 \subseteq A_w$: $|M_1| > |M_2| \Rightarrow N_{M_1}(w) > N_{M_2}(w)$
stellt dieses „gleichmäßige Verbinden“ der Knoten sicher:
Ein innerer Knoten k repräsentiert eine Menge von Zeichen $M_k \subset A_w$ mit Gesamthäufigkeit $N_{M_k}(w)$. Wenn es einen Knoten l gibt, der eine kleinere Menge von Zeichen M_l repräsentiert, gilt nach c) $N_{M_k}(w) > N_{M_l}(w)$, weil $|M_k| > |M_l|$.
Da beim Huffman-Baum stets zwei Knoten mit der geringsten Häufigkeit verbunden werden, wird ein Knoten also erst weiter verbunden, wenn alle Knoten, die eine kleinere Menge von Zeichen repräsentieren bereits verbunden wurden.
Es werden also zunächst stets zwei Blätter miteinander verbunden, dann zwei Knoten die je 2 Zeichen repräsentieren, usw...

Dieses Prinzip geht **nur deshalb** auf, weil sich auf jeder Ebene des Baumes stets

zwei Knoten miteinander verbinden lassen, also stets jeder Knoten einen „Partner“ mit gleichgroßer Menge repräsentierter Zeichen findet, weil ihre Gesamtanzahl stets gerade ist. Dies wird durch die zusätzlich definierte Bedingung: $|A_w| = 2^k, k \in \mathbb{N}_+$ genau fordert. Die Anzahl Blätter eines Huffman-Baumes ist genau $|A_w|$. Werden Knoten nach dem oben genannten Prinzip verbunden, so halbiert sich ihre Anzahl auf jeder Ebene des Baumes. Eine 2er-Potenz lässt sich stets durch 2 teilen, bis man irgendwann bei $2^0 = 1$ Knoten, also einer Wurzel landet.

Anmerkung:
*Dieser Beweis ist als Beispiel eines „erklärenden Beweises“ gedacht. Wie viel Argumentation für so einen Beweis ausreichend ist, hängt maßgeblich von 2 Faktoren ab: Je komplizierter der Sachverhalt, umso mehr wird als vergleichsweise „trivial“ angesehen (und rausgekürzt). Andererseits gilt es als Beweisführende*r auch das eigene Verständnis des Sachverhaltes zu beweisen. Sie werden lernen hier die richtige Balance zu finden. In der Regel ist es ausreichend ein paar Kernelemente der Argumentation präzise zu formulieren. Mit fortgeschrittenem Semester wird man Ihnen automatisch mehr zutrauen, sodass Sie bei Ihren Beweisen weniger ins Detail gehen müssen, aber andererseits komplexere Sachverhalte auf sie los lassen...*