Grundbegriffe der Informatik — Aufgabenblatt 5
Lösungsvorschläge

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

Ausgabe: Freitag, 25.11.2022, 14:30 Uhr

Abgabe: Freitag, 2.12.2022, 12:30 Uhr
Online, oder in einem Briefkasten mit der Aufschrift GBI 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 Tutorien mit Online-Abgabe:

- handschriftlich erstellt (Scans und lesbare Fotos akzeptiert)
- **rechtzeitig**, mit diesem Deckblatt in **genau einer** PDF-Datei
- in ILIAS unter "Tutorien" im Ordner des richtigen Tutoriums abgeben.

---

*Von Tutor*in auszufüllen:* erreichte Punkte

Blatt 5: / 21 Blätter 1 – 5, Stud. 1: / 105

Blätter 1 – 5, Stud. 2: / 105

Aufgabe 5.1 **(1.5 + 0.5 + 1 + 0.5 + 1.5 + 1 + 2 = 8 Punkte)**
Seien $A = \{\texttt{a}, \texttt{e}, \texttt{m}, \texttt{t}\}$ und $B = \{\texttt{0}, \texttt{1}\}$ zwei Alphabete und $w = \texttt{matetee}$. Wir wollen uns im Folgenden mit Codierungen von Wörten über dem Alphabet A befassen.

a) Berechnen Sie eine Huffman-Codierung, indem Sie einen Huffman-Baum für das Wort w aufstellen. Geben Sie dafür eine Funktion $h\colon A \to B^*$ an, die einen Homomorphismus h^{**} induziert, der die Huffman-Codierung des Wortes w erzeugt.
b) Geben Sie das codierte Wort $h^{**}(w)$ an.
c) Geben Sie eine unendliche Menge $V \subseteq A^*$ an, so dass h^{**} die Huffman-Codierung für jedes $v \in V$ ist.
d) Sei eine weitere Codierung $c^{**} : A^* \to B^*$ gegeben, die durch die Funktion $c\colon A \to B^*$ wie folgt induziert wird:

$$c(\texttt{a}) = \texttt{00}, \quad c(\texttt{e}) = \texttt{01}, \quad c(\texttt{m}) = \texttt{10}, \quad c(\texttt{t}) = \texttt{11}$$

Geben Sie das codierte Wort $c^{**}(\texttt{matetee})$ an.
e) Geben Sie ein Wort $v \in A^*$ minimaler Länge an, so dass
   * jedes Zeichen von A mindestens einmal in v vorkommt, und
   * die Codierung $h^{**}(v)$ echt kürzer ist als $c^{**}(v)$, also $|h^{**}(v)| < |c^{**}(v)|$ gilt.

Bei Datenübertragungen wird häufig (jedoch nicht ausschließlich) das binäre Alphabet B verwendet. In der Realität sind Datenübertragungen allerdings nicht immer fehlerfrei. Im Folgenden wollen wir uns rudimentär mit fehlerbehafteten Übertragungen auseinandersetzen. Dafür wollen wir davon ausgehen, dass während der Übertragung eines Codewortes über B Störungen auftreten können. Diese Störungen machen sich in unserer Betrachtung durch ein „gekipptes“ Bit bemerkbar. Um eine Störung eines bestimmten Bits zu simulieren, wollen wir die Störfunktion $\not{z}_i$ verwenden, die ein Wort über B auf das gleiche Wort, allerdings mit einer Störung an der Stelle i, abbildet – also die $\texttt{1}$ mit der $\texttt{0}$ vertauscht, und andersherum. Die Funktion ist formal wie folgt definiert:

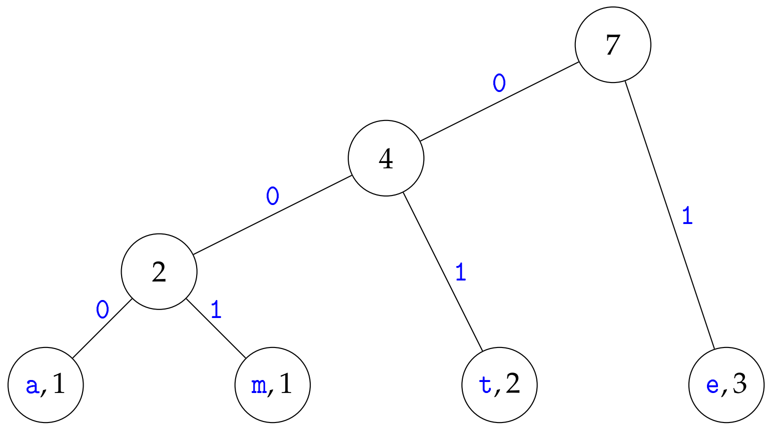
$$\not{z}_i\colon B^* \to B^*, u \mapsto w \text{ mit } w(j) = \begin{cases} \texttt{0} & i = j \text{ und } u(i) = \texttt{1} \\ \texttt{1} & i = j \text{ und } u(i) = \texttt{0} \\ u(j) & i \neq j \end{cases} \qquad (\text{für } 0 \leq j < |u|)$$

mit $|u| = |\not{z}_i(u)|$ für $0 \leq i < |u|$.

f) Decodieren Sie die gestörten Codewörter $\not{z}_2(h^{**}(w))$, $(\not{z}_2 \circ \not{z}_6)(h^{**}(w))$, $\not{z}_2(c^{**}(w))$ und $(\not{z}_2 \circ \not{z}_6)(c^{**}(w))$.
g) Wiegen Sie die Vor-, und Nachteile der beiden Codierungen h^{**} und c^{**} ab, wenn Wörter über A codiert übertragen werden sollen. Nennen Sie dazu ein Szenario in der die Kodierung h^{**} vorteilhafter ist und ein Szenario, in der die Codierung c^{**} vorteilhafter ist.

Lösung 5.1

a) Die Struktur des Huffman-Baums ist eindeutig. Ein gültiger Huffman-Baum lässt sich beispielsweise mit folgener Kantenbeschriftung erstellen:



Indem wir die Zeichen entlang des Pfades von der Wurzel zu den jeweiligen Knoten des Baumes konkatenieren, erhalten wir die folgende Funktion h:

x	a	e	m	t
$h(x)$	000	1	001	01

b)

$$\begin{aligned} h^{**}(\mathtt{matetee}) &= h(\mathtt{m})h(\mathtt{a})h(\mathtt{t})h(\mathtt{e})h(\mathtt{t})h(\mathtt{e})h(\mathtt{e}) \\ &= \mathtt{001} \cdot \mathtt{000} \cdot \mathtt{01} \cdot \mathtt{1} \cdot \mathtt{01} \cdot \mathtt{1} \cdot \mathtt{1} \\ &= \mathtt{0010000110111} \end{aligned}$$

c) Beispielsweise kann man V wählen als

$$V = \{\mathtt{m}^i\mathtt{a}^i\mathtt{t}^{2i}\mathtt{e}^{3i} \mid i \in \mathbb{N}_+\}$$

Die Funktion h^{**} ist eine Huffman-Codierung für alle Wörter $v \in V$, da die relativen Häufigkeiten der Zeichen in allen $v \in V$ dieselben sind, wie im Wort w, für das Huffman-Codierung h^{**} erstellt wurde.

d)

$$\begin{aligned} c^{**}(\mathtt{matetee}) &= c(\mathtt{m})c(\mathtt{a})c(\mathtt{t})c(\mathtt{e})c(\mathtt{t})c(\mathtt{e})c(\mathtt{e}) \\ &= \mathtt{10} \cdot \mathtt{00} \cdot \mathtt{11} \cdot \mathtt{01} \cdot \mathtt{11} \cdot \mathtt{01} \cdot \mathtt{01} \\ &= \mathtt{10001101110101} \end{aligned}$$

e) Das Wort $\mathtt{aeeemt} \in A^6$ leistet das Gewünschte. Für jede Huffman-Codierung h haben zwei der Zeichen aus der Menge $\{\mathtt{a}, \mathtt{m}, \mathtt{t}\}$ die Codelänge 3, eines hat die Codelänge 2, und das Zeichen e hat die Codelänge 1. Daraus ergibt sich eine Codelänge von $|h^*(\mathtt{aeeemt})| = 2 \cdot 3 + 2 + 3 \cdot 1 = 11$. Für die andere Codierung gilt $|c^*(\mathtt{aeeemt})| = 6 \cdot 2 = 12 > 11$, also die gesuchte Eigenschaft.

f)

Ursprungswort w	matetee	matetee
Codierung f	h	c
Codiertes Wort $f^{**}(w) = x$	0010000110111	10001101110101
$\lightning_2(x)$	0000000110111	10101101110101
Decodierung der einfachen Störung	aatetee	mmtetee
$(\lightning_2 \circ \lightning_6)(x)$	0000001110111	10101111110101
Decodierung der doppelten Störung	aaeeetee	mmttee

g) i) **Szenario:** Datenspeicherung
Ein Wort (mit optimaler Eigenschaft für Codierung h^{**} soll auf einer Festplatte mit beschränkter größe gespeichert werden. Hier hat die die Huffman-Codierung h^{**} einen Vorteil, da die kleinstmögliche Codierung eben durch diese gegeben ist.

ii) **Szenario:** Datenübertragung in Form einer Email
Ein Wort soll so übertragen werden, dass eine andere Person diese nach dem Erhalt des Datums und anschließender Decodierung lesen kann. Hier hat die Codierung c^{**} einen Vorteil, da im Falle einer Störung nur das Zeichen, welches das gestörte Bit enthielt, fehlerbehaftet decodiert wird. Der Rest der Nachricht bleibt fehlerfrei.

Aufgabe 5.2 (0.5 + 1.5 + 1 + 1 + 2 = 6 Punkte)
Auf Blatt 2 haben Sie bereits den Operator $\oslash_n\colon Z_2^n \times Z_2^n \to Z_2^n$ sowie in der Vorlesung die Funktionen Repr_k, Num_k, und die Modulooperation $a \textbf{ mod } b$ kennen gelernt. Zur Erinnerung:

$$u \oslash_n v = w \text{ mit } w(i) = \begin{cases} \mathtt{1}, & \text{wenn } u(i) = \mathtt{1} \text{ und } v(i) = \mathtt{1} \\ \mathtt{0}, & \text{sonst} \end{cases}$$

a) Berechnen Sie 28 **mod** 16
b) Berechnen Sie $\text{Repr}_2(28)$ schrittweise.
c) Berechnen Sie $\text{Num}_2(\text{Repr}_2(28) \oslash_5 \mathtt{01111})$.
d) Sei $\text{shl}\colon Z_2^* \to Z_2^*, w \mapsto w \cdot \mathtt{0}$ der „shift left" Operator. Beweisen Sie, dass $\text{Num}_2(\text{shl}(w)) = 2 \cdot \text{Num}_2(w)$ für alle $w \in Z_2^*$ gilt.
e) Beweisen Sie durch vollständige Induktion über k, dass $2^k - 1 = \text{Num}_2(\mathtt{1}^k)$ für alle $k \in \mathbb{N}_0$ gilt.

Lösung 5.2

a) $28 \textbf{ mod } 16 = 12$

b)

$$\begin{aligned}
\text{Repr}_2(28) &= \text{Repr}_2(28 \textbf{ div } 2) \cdot \text{repr}_2(28 \textbf{ mod } 2) \\
&= \text{Repr}_2(14 \textbf{ div } 2) \cdot \text{repr}_2(14 \textbf{ mod } 2) \cdot \mathtt{0} \\
&= \text{Repr}_2(7 \textbf{ div } 2) \cdot \text{repr}_2(7 \textbf{ mod } 2) \cdot \mathtt{00} \\
&= \text{Repr}_2(3 \textbf{ div } 2) \cdot \text{repr}_2(3 \textbf{ mod } 2) \cdot \mathtt{100} \\
&= \text{repr}_2(1) \cdot \mathtt{1100} \\
&= \mathtt{11100}
\end{aligned}$$

c)

$$\begin{aligned}
\text{Num}_2(\text{Repr}_2(28) \oslash_5 \mathtt{01111}) &= \text{Num}_2(\mathtt{11100} \oslash_5 \mathtt{01111}) \\
&= \text{Num}_2(\mathtt{01100}) \\
&= 12
\end{aligned}$$

d) Für alle $w \in Z_2^*$ gilt:

$$\text{Num}_2(\text{shl}(w)) = \text{Num}_2(w\mathtt{0}) = 2 \cdot \text{Num}_2(w) + \text{num}_2(\mathtt{0}) = 2 \cdot \text{Num}_2(w) + 0 = 2 \cdot \text{Num}_2(w)$$

e) **Beh.:** Für alle $k \in \mathbb{N}_0$ gilt: $2^k - 1 = \mathrm{Num}_2(\mathtt{1}^k)$.
Bew.:

Induktionsanfang: Sei $k = 0$, dann gilt:

$$2^k - 1 = 2^0 - 1 = 1 - 1 = 0 = \mathrm{Num}_2(\varepsilon) = \mathrm{Num}_2(\mathtt{1}^0) = \mathrm{Num}_2(\mathtt{1}^k)$$

Induktionsvoraussetzung: Für ein beliebiges, aber festes $k \in \mathbb{N}_0$ gelte $2^k - 1 = \mathrm{Num}_2(\mathtt{1}^k)$.

Induktionsschluss : Gilt die Beh. für k, dann gilt diese auch für $k + 1$.

$$\begin{aligned}
\mathrm{Num}_2(\mathtt{1}^{k+1}) &= \mathrm{Num}_2(\mathtt{1}^k \cdot \mathtt{1}) && (\text{ Konkatenation })\\
&= 2 \cdot \mathrm{Num}_2(\mathtt{1}^k) + \mathrm{num}_2(\mathtt{1}) && (\text{ Def. } \mathrm{Num}_2)\\
&= 2 \cdot \mathrm{Num}_2(\mathtt{1}^k) + 1 && (\text{ Def. } \mathrm{num}_2)\\
&= 2 \cdot (2^k - 1) + 1 && (\text{ Induktionsvoraussetzung })\\
&= 2^{k+1} - 2 + 1 &&\\
&= 2^{k+1} - 1 &&
\end{aligned}$$

Und daraus folgt die Behauptung.

Aufgabe 5.3 (1 + 6 = 7 Punkte)
In dieser Aufgabe befassen wir uns mit der Addition von Binärzahlen, die wir formalisieren wollen. Dabei bildet der Operator $\oplus_n^c \colon Z_2^n \times Z_2^n \to Z_2^{n+1}$ zwei Wörter der Länge $n \in \mathbb{N}_0$, die jeweils eine Binärzahl repräsentieren auf ein Wort der Länge $n + 1$ ab, das die Summe beider Zahlen repräsentieren soll. Im Superskript wird dabei der Übertrag $c \in Z_2$ aus vorangegangenen Addtionsschritten mitgeführt, so dass sich hinter $\oplus_n^c$ eigentlich zwei Operationen $\oplus_n^{\mathtt{0}}$ und $\oplus_n^{\mathtt{1}}$ verbergen.

Wir betrachten nun die folgende induktive Definition für die durch $\oplus_n^c$ beschriebenen Funktionen für beliebige $c \in Z_2, n \in \mathbb{N}_0$:

$$\varepsilon \oplus_0^c \varepsilon = c$$
$$\forall v, w \in Z_2^n\ \forall \alpha, \beta \in Z_2 : v\alpha \oplus_{n+1}^c w\beta = (v \oplus_n^{\gamma(\alpha,\beta,c)} w) \cdot \delta(\alpha, \beta, c)$$

Dabei beschreibt

$$\delta \colon Z_2 \times Z_2 \times Z_2 \to Z_2, (x, y, z) \mapsto \begin{cases} \mathtt{1}, & \text{wenn } N_{\mathtt{1}}(xyz) \in \{1, 3\} \\ \mathtt{0}, & \text{wenn } N_{\mathtt{1}}(xyz) \in \{0, 2\} \end{cases}$$

die binäre Ergebnisziffer der niedrigsten Stelle, die aus Addition der drei Argumente hervorgeht, während der binäre Übertrag in die nächste Stelle durch die Funktion

$$\gamma \colon Z_2 \times Z_2 \times Z_2 \to Z_2, (x, y, z) \mapsto \begin{cases} \mathtt{1}, & \text{wenn } N_{\mathtt{1}}(xyz) > 1 \\ \mathtt{0}, & \text{wenn } N_{\mathtt{1}}(xyz) \le 1 \end{cases}$$

berechnet wird.

a) Berechnen Sie $\mathtt{0110} \oplus_n^{\mathtt{1}} \mathtt{1101}$ schrittweise

b) Wir wollen nun die Kommutativität von $\oplus_n^0$ beweisen. Dabei gehen wir strukturiert vor. Zeigen Sie zuerst die beiden folgenden hilfreichen Lemmata:

   i) $\forall x, y, z \in Z_2 : \delta(x, y, z) = \delta(y, x, z)$
   (d.h., δ ist kommutativ in den ersten beiden Argumenten).

   ii) $\forall x, y, z \in Z_2 : \gamma(x, y, z) = \gamma(y, x, z)$
   (d.h. γ ist kommutativ in den ersten beiden Argumenten).

   Der Beweis der Kommutativität von $\oplus_n^0$ hängt auch von der Kommutativität von $\oplus_n^1$ ab. Daher wollen wir eine stärkere Behauptung beweisen, die auch die Kommutativität von $\oplus_n^0$ impliziert.

   iii) Beweisen Sie: $\forall n \in \mathbb{N}_0, \forall c \in Z_2, \forall w, v \in Z_2^n : w \oplus_n^c v = v \oplus_n^c w$. Verwenden Sie dazu vollständige Induktion über die Wortlänge $|w| = |v| = n$.

Lösung 5.3

a)

$$\begin{aligned}
\texttt{0110} \oplus_4^1 \texttt{1101} &= (\texttt{011} \oplus_3^1 \texttt{110})\texttt{0} \\
&= (\texttt{01} \oplus_2^1 \texttt{11})\texttt{00} \\
&= (\texttt{0} \oplus_1^1 \texttt{1})\texttt{100} \\
&= (\varepsilon \oplus_0^1 \varepsilon)\texttt{0100} \\
&= \texttt{10100}
\end{aligned}$$

b) Wir bemerken zuerst, dass für beliebige Zeichen $a, x_1, x_2, x_3 \in A$ aus einem Alphabet A gilt, dass $N_a(x_1x_2x_3) = N_a(x_2x_1x_3)$. Das folgt daraus, dass die Anzahl der Zeichen a in einem Wort $x_1x_2x_3$ nicht von der Reihenfolge der Zeichen $x_i, i \in \{1, 2, 3\}$, abhängt, sondern nur davon, wie viele der $x_i, i \in \{1, 2, 3\}$ gleich a sind.

   i.) **Lemma 1**: $\forall x, y, z \in Z_2 : \delta(x, y, z) = \delta(y, x, z)$.
   Der Beweis folgt mit der Bemerkung angewendet auf $N_\texttt{1}(xyz)$:

$$\delta(x, y, z) = \begin{cases} 1, & \text{wenn } N_\texttt{1}(xyz) \in \{1, 3\} \\ 0, & \text{wenn } N_\texttt{1}(xyz) \in \{0, 2\} \end{cases} = \begin{cases} 1, & \text{wenn } N_\texttt{1}(yxz) \in \{1, 3\} \\ 0, & \text{wenn } N_\texttt{1}(yxz) \in \{0, 2\} \end{cases} = \delta(y, x, z)$$

   ii.) **Lemma 2**: $\forall x, y, z \in Z_2 : \gamma(x, y, z) = \gamma(y, x, z)$.
   Der Beweis folgt wieder mit der Bemerkung angewendet auf $N_\texttt{1}(xyz)$:

$$\gamma(x, y, z) = \begin{cases} 1, & \text{wenn } N_\texttt{1}(xyz) > 1 \\ 0, & \text{wenn } N_\texttt{1}(xyz) \leq 1 \end{cases} = \begin{cases} 1, & \text{wenn } N_\texttt{1}(yxz) > 1 \\ 0, & \text{wenn } N_\texttt{1}(yxz) \leq 1 \end{cases} = \gamma(y, x, z)$$

   iii.) Den Beweis für die Behauptung $\forall n \in \mathbb{N}_0, \forall c \in Z_2, \forall w, v \in Z_2^n : w \oplus_n^c v = v \oplus_n^c w$ führen wir nun mittels vollständiger Induktion über die Wortlänge $|w| = |v| = n \in \mathbb{N}_0$.

   Induktionsanfang: Sei $n = 0$, dann ist $w = v = \varepsilon$ und es gilt:

$$v \oplus_0^c w = \varepsilon \oplus_0^c \varepsilon = w \oplus_0^c v \tag{1}$$

Induktionsschluss: Sei $n \in \mathbb{N}_0$ beliebig aber fest.
Es gelte die **Induktionsvoraussetzung**: $\forall c \in Z_2, \forall w, v \in Z_2^n : w \oplus_n^c v = v \oplus_n^c w$.

Wir betrachten nun Wörter $v = v'\alpha, w = w'\beta$ der Länge $n+1$ mit $|v'|, |w'| = n$ und $\alpha, \beta \in Z_2$. Es gilt

$$\begin{aligned}
v \oplus_{n+1}^c w &= v'\alpha \oplus_{n+1}^c w'\beta \\
&= (v' \oplus_n^{\gamma(\alpha,\beta,c)} w') \cdot \delta(\alpha,\beta,c) \quad (\text{ Definition von } \oplus_{n+1}^c) \\
&= (v' \oplus_n^{\gamma(\alpha,\beta,c)} w') \cdot \delta(\beta,\alpha,c) \quad (\text{ Lemma 1 }) \\
&= (v' \oplus_n^{\gamma(\beta,\alpha,c)} w') \cdot \delta(\beta,\alpha,c) \quad (\text{ Lemma 2 }) \\
&= (w' \oplus_n^{\gamma(\beta,\alpha,c)} v') \cdot \delta(\beta,\alpha,c) \quad (\text{ Induktionsvoraussetzung }) \\
&= w'\beta \oplus_{n+1}^c v'\alpha \quad (\text{ Definition von } \oplus_{n+1}^c) \\
&= w \oplus_{n+1}^c v
\end{aligned}$$

Und damit folgt die Behauptung.