Grundbegriffe der Informatik — Aufgabenblatt 6
Lösungsvorschläge

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

Ausgabe: 9. Dezember 2020

Abgabe: 22. Dezember 2020, 12:00 Uhr
durch Hochladen in den Ilias-Kurs

Lösungen werden nur korrigiert, wenn sie
- handschriftlich erstellt sind und
- rechtzeitig
- mit dieser Seite als Deckblatt
- gescannt oder fotografiert mit allen Seiten in *einer* Pdf-Datei

ins Ilias-System hochgeladen werden.

Vom Tutor auszufüllen: erreichte Punkte

Blatt 6:	/ 23	Blätter 1 – 6, Stud. 1:	/ 120
		Blätter 1 – 6, Stud. 2:	/ 120

Aufgabe 6.1 (2 + 2 + 2 = 6 Punkte)

Es sei $A = \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}\}$ und $B = \{\texttt{0}, \texttt{1}\}$.

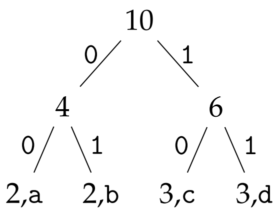
a) Erstellen Sie einen Huffman-Baum zum Wort $w = \texttt{ccdbdaacdb} \in A^*$. Geben Sie anschließend die Codierung $H(w)$ von w anhand des zu Ihrem Baum zugehörigen Huffman-Codes H.

b) Das Codewort $c = H(w)$ wird auf einem störungsbehafteten Kanal übertragen, sodass genau ein Bit des Wortes c an einer zufälligen Position i gekippt ist. Der Empfänger erhält also ein Wort $c' \in B^*$ mit $|c'| = |c|$ und so, dass es ein $i \in \mathbb{Z}_{|c|}$ derart gibt, dass $c(i) \neq c'(i)$ und $\forall j \in \mathbb{Z}_{|c|} \setminus \{i\} : c'(j) = c(j)$ ist.
Kann der Empfänger nur anhand von H und c' stets eindeutig bestimmen, welches Wort w ursprünglich versendet worden ist? Falls ja, begründen Sie, warum das sein muss. Andernfalls geben Sie ein solches c' sowie unterschiedliche Wörter $w_1, w_2 \in A^*$ derart an, dass c' sowohl $H(w_1)$ als auch $H(w_2)$ entsprechen kann.

c) Man nehme jetzt an, dass wir $c = \pi(H(w))$ statt $H(w)$ auf dem Kanal übertragen, wobei $\pi\colon \{\texttt{0}, \texttt{1}\}^* \to \{\texttt{0}, \texttt{1}\}^*$ der Homomorphismus ist, der durch $\pi(\texttt{0}) = \texttt{000}$ und $\pi(\texttt{1}) = \texttt{111}$ induziert wird. Dazu sei c' wie in Teilaufgabe b).
Kann der Empfänger nur anhand von H, π, und c' stets eindeutig bestimmen, welches Wort w ursprünglich versendet worden ist? Falls ja, begründen Sie, warum das sein muss. Andernfalls geben Sie ein solches c' sowie unterschiedliche Wörter $w_1, w_2 \in A^*$ derart an, dass c' sowohl $H(w_1)$ als auch $H(w_2)$ entsprechen kann.

Lösung 6.1

a) Huffman-Baum:



also $H(\texttt{a}) = \texttt{00}$, $H(\texttt{b}) = \texttt{01}$, $H(\texttt{c}) = \texttt{10}$, $H(\texttt{d}) = \texttt{11}$

also $H(\texttt{ccdbdaacdb}) = \texttt{10 10 11 01 11 00 00 10 11 01}$

b) Nein. Z. B. zu $c' = \texttt{01}$ kann sowohl $c = \texttt{00}$ oder $c = \texttt{11}$ gehören, das heißt, der Empfänger kann nicht herausfinden, ob ein a oder ein d übertragen worden ist.

c) Ja.
Wenn man $c = \pi(H(w))$ in Blöcke der Länge 3 aufteilt, können nur die Blöcke 000 und 111 entstehen.
Nun teilt man c' in Blöcke der Länge 3. Wenn genau ein Bitfehler aufgetreten ist, sind noch immer alle Blöcke von einer der genannten Formen und genau ein Block enthält entweder zwei 0 und eine 1 oder zwei 1 und eine 0.
In allen Fällen ist das Bit, das in einem Block in der Mehrheit ist, das „richtige“. D. h., indem man jeden Block das entsprechende Bit umwandelt, erhält man $H(w)$, was man wie in der Vorlesung beschrieben decodieren kann.
Wer es formaler mag:
Es sei $K = \{\texttt{000}, \texttt{111}\}$ die Menge der Blöcke ohne Fehler, $F_0 = \{\texttt{001}, \texttt{010}, \texttt{100}\}$, die Menge der Blöcke, in denen eine 0 verfälscht wurde, $F_1 = \{\texttt{110}, \texttt{101}, \texttt{011}\}$ die Menge der Blöcke, in denen eine 1 verfälscht wurde, und $F = F_0 \cup F_1$.
Dann ist c' in der Sprache K^*FK^*. Da jedes Symbol durch zwei Bits codiert wird, weiß man sogar genauer:

$$c' \in (K^2)^*(F_0K \cup KF_0 \cup F_1K \cup F_1K)(K^2)^*.$$

Jedes Teilwort in c' aus K^2 lässt sich also eindeutig mittels H einem Zeichen aus A zuzuordnen, weil da kein Fehler entstanden ist. Zudem ist jedes Teilwort aus F_xK bzw. KF_x für $x \in B$ ebenfalls einem eindeutigen Zeichen aus A zuzuordnen, weil F_0 und F_1 disjunkt sind.
Anmerkung: Die Strategie, die durch den Homomorphismus π realisiert wird, heißt *Padding* und kommt in mehreren Gebieten der Informatik vor. In diesem Kontext sorgt π für mehr Redundanz im übertragenen Wort. Wie gesehen ist das eine „Abwehr" gegen zufällig aufgetretene Fehler.

Aufgabe 6.2 (1 + 1 + 2 = 4 Punkte)
Es sei $A = \{\mathtt{0}, \mathtt{1}\}$ und $\mathrm{and}, \mathrm{or}, \mathrm{xor}\colon A \times A \to A$ die in der Übung vorgestellten binären Operationen. Zudem sei für $c \in A$ die Abbildung $P_c\colon A^* \times A^* \to A^*$ wie folgt induktiv definiert:

$$\begin{aligned} \forall w \in A^* : &\qquad P_c(w, \varepsilon) = P_c(\varepsilon, w) = \varepsilon \\ \forall w_1, w_2 \in A^*\ \forall x_1, x_2 \in A : &\qquad P_c(w_1x_1, w_2x_2) = P_{c'}(w_1, w_2) \cdot \mathrm{xor}(c, \mathrm{xor}(x_1, x_2)), \end{aligned}$$

wobei $c' = \mathrm{or}(\mathrm{and}(x_1, x_2), \mathrm{and}(c, \mathrm{or}(x_1, x_2)))$ ist.

a) Rechnen Sie $P_0(\mathtt{101}, \mathtt{001})$ schrittweise aus. Wenden Sie dabei bei jedem Schritt die Definition von P_c höchstens einmal an.
b) Es sei $n \in \mathbb{N}_+$ beliebig. Zeigen Sie, dass genau dann $P_0(\mathtt{1}^n, \mathtt{0}^{n-1}\mathtt{1}) = \mathtt{0}^n$ ist, wenn $P_1(\mathtt{1}^{n-1}, \mathtt{0}^{n-1}) = \mathtt{0}^{n-1}$ gilt.
c) Zeigen Sie durch vollständige Induktion über m, dass Folgendes gilt:

$$\forall m \in \mathbb{N}_0 : \ P_1(\mathtt{1}^m, \mathtt{0}^m) = \mathtt{0}^m.$$

Lösung 6.2
Anmerkung: Die Abbildung P wird in der technischen Informatik *Volladdierer* genannt. Man kann zeigen, dass P die Addition (mit Übertrag) von Zahlen in Zweierkomplementdarstellung realisiert.

a)

$$P_0(\mathtt{101}, \mathtt{001}) = P_1(\mathtt{10}, \mathtt{00})\mathtt{0} = P_0(\mathtt{1}, \mathtt{0})\mathtt{10} = P_0(\varepsilon, \varepsilon)\mathtt{110} = \mathtt{110}$$

b) Wir wenden die Definition von P_c einmal an:

$$\begin{aligned} P_0(\mathtt{1}^{n-1}\mathtt{1}, \mathtt{0}^{n-1}\mathtt{1}) = \mathtt{0}^n \qquad &\text{gdw.} \qquad P_1(\mathtt{1}^{n-1}, \mathtt{0}^{n-1})\mathtt{0} = \mathtt{0}^n \\ &\text{gdw.} \qquad P_1(\mathtt{1}^{n-1}, \mathtt{0}^{n-1}) = \mathtt{0}^{n-1} \end{aligned}$$

c) Zu zeigen:

$$\forall m \in \mathbb{N}_0 : \ P_1(\mathtt{1}^m, \mathtt{0}^m) = \mathtt{0}^m.$$

Vollständige Induktion über m:

IA: Für $m = 0$ ist $P_1(\mathtt{1}^m, \mathtt{0}^m) = P_1(\varepsilon, \varepsilon) = \varepsilon = \mathtt{0}^m$.

IS: Es sei $m \in \mathbb{N}_0$ beliebig und es gelte die

IV: $P_1(\mathtt{1}^m, \mathtt{0}^m) = \mathtt{0}^m$.

Zu zeigen ist die IB: $P_1(1^{m+1}, 0^{m+1}) = 0^{m+1}$

$$\begin{aligned} P_1(1^{m+1}, 0^{m+1}) &= P_1(1^m, 0^m)0 && \text{nach Def. von } P \\ &\overset{\text{IV}}{=} 0^m0 = 0^{m+1}. \end{aligned}$$

Aufgabe 6.3 (0.5 + 0.5 + 2 = 3 Punkte)

a) Geben Sie $\mathrm{Zkpl}_\ell(3)$ für jedes $\ell \in \{3, 4, 5\}$ an.

b) Geben Sie $\mathrm{Zkpl}_\ell(-4)$ für jedes $\ell \in \{3, 4, 5\}$ an.

c) Es sei $A = \{0, 1\}$ und $\ell \in \mathbb{N}_+$, $\ell \geq 1$. Geben Sie für jedes $k \in \mathbb{N}_0$ eine Abbildung $\ddot{U}_k \colon A^\ell \to A^{\ell+k}$ an, sodass für jedes $x \in \mathbb{K}_\ell$ gilt: $\ddot{U}_k(\mathrm{Zkpl}_\ell(x)) = \mathrm{Zkpl}_{\ell+k}(x)$.
Bei Ihrer Definition von $\ddot{U}_k$ darf dabei keine der Abbildungen Zkpl, bin, Repr, und Num vorkommen.
Begründen Sie anschließend, warum Ihr $\ddot{U}_k$ das Geforderte leistet. Einen Beweis müssen Sie nicht erbringen; eine plausible Begründung genügt.

Lösung 6.3

a) $\mathrm{Zkpl}_3(3) = 011$, $\mathrm{Zkpl}_4(3) = 0011$, $\mathrm{Zkpl}_5(3) = 00011$

b) $\mathrm{Zkpl}_3(-4) = 100$, $\mathrm{Zkpl}_4(-4) = 1100$, $\mathrm{Zkpl}_5(-4) = 11100$

c) Für $w \in A^\ell$ sei

$$\ddot{U}_k(w) = \begin{cases} 0^k w, & \text{falls } w(0) = 0 \\ 1^k w, & \text{falls } w(0) = 1. \end{cases}$$

oder kurz $\ddot{U}_k(w) = w(0)^k w$.
Man kann das verschieden begründen. Hier zwei Vorschläge:

Begründung 1. In der großen Übung wurde gesagt:

Für $b \in Z_2$ und $w \in Z_2^*$ ist
bw Zweierkomplementdarstellung von $-2^{|w|} \cdot \mathrm{num}_2(b) + \mathrm{Num}_2(w)$

Angewendet auf $\ddot{U}_k(bw) = b^k bw = bb^k w$ ist das die Zweierkomplementdarstellung von

$$\begin{aligned} &-2^{k+|w|}\mathrm{num}_2(b) + \mathrm{Num}_2(b^k w) \\ =\ &-2^{k+|w|}\mathrm{num}_2(b) + (2^k - 1)\mathrm{num}_2(b) \cdot 2^{|w|} + \mathrm{Num}_2(w) \\ =\ &(-2^{k+|w|} + (2^k - 1) \cdot 2^{|w|})\mathrm{num}_2(b) + \mathrm{Num}_2(w) \\ =\ &-2^{|w|}\mathrm{num}_2(b) + \mathrm{Num}_2(w) \end{aligned}$$

und das ist die Zahl mit Zweierkomplementdarstellung bw.

Begründung 2. Rechnen

- Für $k = 0$ ist $\ddot{U}_k$ die Identität, was offensichtlich das Geforderte leistet.
- Sei daher nun $k \in \mathbb{N}_+$: Laut Vorlesung ist $\mathrm{Zkpl}_\ell(x) = \begin{cases} 0\mathrm{bin}_{\ell-1}(x) & \text{falls } x \geq 0 \\ 1\mathrm{bin}_{\ell-1}(2^{\ell-1} + x) & \text{falls } x < 0 \end{cases}$

  Unterscheide zwei Fälle nach Vorzeichen von x:

  1. Fall $x \geq 0$: Dann ist $\mathrm{Zkpl}_{\ell+k}(x) = 0\mathrm{bin}_{\ell+k-1}(x) = 00^{\ell+k-1-|\mathrm{Repr}_2(x)|}\mathrm{Repr}_2(x)$
  $$\begin{aligned} &= 0^k 00^{\ell-1-|\mathrm{Repr}_2(x)|}\mathrm{Repr}_2(x) \\ &= 0^k 0\mathrm{bin}_{\ell-1}(x) = \ddot{U}_k(\mathrm{Zkpl}_\ell(x)) \end{aligned}$$

2. Fall $x < 0$: Wegen $x \in \mathbb{K}_\ell$ ist insbesondere $-2^{\ell-1} \leq x < 0$ und folglich $2^{\ell+k-1} + x \geq 2^{\ell+k-1} - 2^{\ell-1} = (2^{\ell+k-1} - 1) + (1 - 2^{\ell-1})$
Also ist $\mathrm{bin}_{\ell+k-1}(2^{\ell+k-1} + x) = 1^k \mathrm{bin}_{\ell-1}(2^{\ell-1} + x)$. *(An dieser Stelle sind wir genau aber übergehen vielleicht den ein oder anderen erläuternden Schritt.)*
Folglich ist

$$\begin{aligned}
\mathrm{Zkpl}_{\ell+k}(x) &= \mathtt{1}\mathrm{bin}_{\ell+k-1}(2^{\ell+k-1} + x) \\
&= \mathtt{1}\mathtt{1}^k \mathrm{bin}_{\ell-1}(2^{\ell-1} + x) \\
&= \mathtt{1}^k \mathtt{1}\mathrm{bin}_{\ell-1}(2^{\ell-1} + x) = \ddot{U}_k(\mathrm{Zkpl}_\ell(x)).
\end{aligned}$$

Aufgabe 6.4 (1.5 + 1 + 1 + 2 + 1 + 0.5 = 7 Punkte)

In der Vorlesung haben Sie kennengelernt, dass die MIMA ausschließlich mit Zahlen in Zweierkomplementdarstellung arbeitet. Nichtsdestotrotz werden wir in dieser Aufgabe sehen, wie man eine geschickte Übersetzung definieren kann, sodass sich auch Zeichenketten im MIMA-Speicher repräsentieren und von der MIMA verarbeiten lassen.

Zu diesem Zweck seien $A = \{\mathtt{a}, \mathtt{b}\}$ und $B = \{\mathtt{0}, \mathtt{1}\}$ Alphabete, $\perp = \mathtt{0}^{24}$ das sogenannte *Nullzeichen*, und $z_\mathtt{a}, z_\mathtt{b} \in B^{24} \setminus \{\perp\}$ zwei beliebige Zweierkomplementdarstellungen, die unterschiedlich sind (also $z_\mathtt{a} \neq z_\mathtt{b}$). Damit definieren wir eine Übersetzung $\ddot{U}\colon A^* \to \bigcup_{m \in \mathbb{N}_+} (B^{24})^{\mathbb{Z}_m}$, die durch folgende Eigenschaft gegeben ist: Für $w \in A^*$ ist $f_w = \ddot{U}(w)$ genau die Abbildung, für die gilt:

1. $f_w \in (B^{24})^{\mathbb{Z}_{|w|+1}}$, das heißt, f_w ist eine Abbildung von $\mathbb{Z}_{|w|+1}$ nach B^{24}.
2. Für jedes $i \in \mathbb{Z}_{|w|}$ gilt $f_w(i) = z_x$ genau dann, wenn $w(i) = x$ ist.
3. $f(|w|) = \perp$.

Gehen Sie davon aus, dass der Inhalt aller Speicherstellen unbekannt ist, es sei denn, dieser wird unten explizit angegeben.

a) Geben Sie $\ddot{U}(w)$ für jedes $w \in \{\mathtt{a}, \mathtt{abb}, \varepsilon\}$ explizit an.

b) Es seien $w_1, w_2 \in A^*$. Geben Sie den Wert $\ddot{U}(w_1 w_2)(i)$ für jedes $i \in \mathbb{Z}_{|w_1|+|w_2|+1}$ an.

Sei $s \in \mathbb{N}_0$ und $s \leq 2^{20} - |w|$. Wir sagen, ein Wort $w \in A^*$ ist an Adresse $a = \mathrm{bin}_{24}(s)$ *gespeichert*, wenn es für jedes $i \in \mathbb{Z}_{|w|+1}$ gilt, dass der Wert an Adresse $a_i = \mathrm{bin}_{24}(s + i)$ des Speichers gleich $f_w(i)$ ist.

c) Das Wort $w = \mathtt{ababa}$ sei an Adresse $\mathrm{bin}_{24}(2)$ gespeichert. Füllen Sie die folgende Tabelle mit dem Inhalt des Speichers für jede der Adressen aus, die unten aufgeführt sind. Sollte der Inhalt unbekannt sein, tragen Sie dann ein Fragezeichen „?“ ein.

Adresse	0	1	10	11	100	101	110	111	1000	1001
Inhalt										

Hinweis. Bei der Tabelle haben wir bei den Adressen führende Nullen weggelassen. Sie dürfen das auch.

d) Das Wort $w = \mathtt{ababa}$ sei weiterhin an Adresse $a = \mathrm{bin}_{24}(2)$ gespeichert. Nehmen wir jetzt an, durch Ausführung eines MIMA-Programms werde der Inhalt des Speichers an Adresse

(i) $\mathrm{bin}_{24}(5)$ (ii) $\mathrm{bin}_{24}(6)$ (iii) $\mathrm{bin}_{24}(7)$ (iv) $\mathrm{bin}_{24}(8)$

durch den Wert $z_\mathtt{b}$ überschrieben. Der Inhalt aller anderen Speicherzellen bliebe unverändert.

Geben Sie für jeden der obigen Fälle an, ob das Wort w weiterhin an Adresse a gespeichert ist. Falls das nicht zutrifft, geben Sie explizit das Wort an, das an Adresse

a gespeichert ist; sollte dieses Wort nicht eindeutig sein bzw. nach Ausführung des Programms an Adresse a ggf. nicht unbedingt ein Wort stehen, erklären Sie, warum das der Fall ist.

e) Es sei $w \in A^+$ wieder beliebig sowie $s \in \mathbb{N}_0$, $s \leq 2^{20}$, und $a = \text{bin}_{24}(s)$. Wenn w an Adresse a gespeichert ist, ist dann an Adresse $a' = \text{bin}_{24}(s+1)$ auch ein Wort gespeichert? Wenn ja, geben Sie dieses Wort an; andernfalls begründen Sie, warum an Adresse a' nicht unbedingt ein Wort steht.

f) Lautet Ihre Antwort zu e) immer noch dieselbe, wenn $w \in A^*$ ist? Begründen Sie.

Lösung 6.4

a) • $\ddot{U}(\mathtt{a}) = z_\mathtt{a}\bot$ • $\ddot{U}(\mathtt{abb}) = z_\mathtt{a}z_\mathtt{b}z_\mathtt{b}\bot$ • $\ddot{U}(\varepsilon) = \bot$

b)
- für $0 \leq i < |w_1|$ ist $\ddot{U}(w_1w_2)(i) = \ddot{U}(w_1)(i) = w_1(i)$
- für $|w_1| \leq i \leq |w_1| + |w_2|$ ist $\ddot{U}(w_1w_2)(i) = \ddot{U}(w_2)(i - |w_1|)$
  dieser Fall beinhaltet auch das $\bot$ „am Ende“

c)

Adresse	0	1	10	11	100	101	110	111	1000	1001
Inhalt	?	?	$z_\mathtt{a}$	$z_\mathtt{b}$	$z_\mathtt{a}$	$z_\mathtt{b}$	$z_\mathtt{a}$	$\bot$	?	?

d) (i) w

(ii) `ababb`

(iii) Nicht unbedingt. Z. B. kann in keiner der höheren Adressen ein $\bot$ vorkommen.

(iv) w

e) Ja, nämlich das Wort w' der Länge $|w'| = |w| - 1$ und für das $w'(i) = w(i+1)$ für jedes $i \in \mathbb{Z}_{|w'|}$ gilt.

f) Nicht mehr. Z. B. wenn $w = \varepsilon$ und wie in der Situation oben bei d) (iii) in keiner höheren Adresse ein $\bot$ vorkommt.

Aufgabe 6.5 (1.5 + 1.5 = 3 Punkte)

In dieser Aufgabe geht es um MIMA-Befehle, die den Kontrollfluss der Maschine steuern. Den Inhalt des Speichers dürfen Sie dabei nicht verändern.

a) Der MIMA-Befehlssatz soll um den Sprung-Befehl `JGEZ` a erweitert werden. Dabei wird der Sprung zu Adresse a genommen falls der Wert, der im Akkumulator steht, die Zweierkomplementdarstellung einer Zahl ≥ 0 ist.
Geben Sie eine Folge von höchstens 3 Befehlen an, die den Befehl `JGEZ` a realisiert. Ihre Befehlsfolge muss dabei mit dem Befehl `JMN` a enden. Sie dürfen keine andere Sprungbefehle verwenden. Erklären Sie anschließend, warum Ihre Befehlsfolge das Geforderte leistet.

b) Angenommen, der Wert im Akkumulator ist die Zweierkomplementdarstellung einer positiven Zahl x. Zudem ist an Adresse c_1 der Wert $\text{Zkpl}_{24}(1)$ gespeichert.
Es sei eine Adresse a gegeben, für die $a \neq c_1$. Geben Sie eine Folge von höchstens 4 Befehlen an, sodass genau dann einen Sprung zu a genommen wird, wenn $x = 0$ ist. Sie dürfen hierbei höchstens einen Sprungbefehl verwenden, wobei der eben definierte Befehl `JGEZ` a auch zulässig ist und als ein Befehl zählt. Erklären Sie anschließend, warum Ihre Befehlsfolge das Geforderte leistet.

Hinweis. Lösungen, die mit keiner Erklärung versehen sind, werden mit 0 Punkten bewertet.

Lösung 6.5

a) `NOT`
`JMN` a

Begründung: Der Sprung `JMN` a wird genau dann genommen, wenn das höchstwertige Bit des Wertes im Akkumulator gleich `1` ist. Wenn man also den Akkumulator davor negiert, wird der Sprung genau dann genommen, wenn dieses Bit gleich `0`, das heißt, wenn der Wert im Akkumulator positiv ist.

b) *Anmerkung:* Die Aufgabe enthielt leider einen Fehler. Statt „positiv" war für die Zahl x „nicht negativ" gemeint. Wenn das so wäre, dann würde unserer Lösungsvorschlag folgendermaßen aussehen:
`NOT`
`ADD` c_1
`JGEZ` a

Begründung: Da x nicht negativ ist, ist das höchstwertige Bit von $w = \mathrm{Zkpl}_{24}(x)$ gleich `0`. Damit hat die Negierung w' von w als höchstwertige Bit `1`. Zudem wird 1 auf diesen Wert addiert genau dann einen nicht-negativen Wert ergeben, wenn ein Overflow auftritt, das heißt, wenn $w' = \mathtt{1}^{24}$ ist. Das ist genau dann der Fall, wenn $w = \mathtt{0}^{24}$, sprich $x = 0$ ist.