

Grundbegriffe der Informatik — Aufgabenblatt 6

Lösungsvorschläge

Tutorium Nr.: Tutor*in:

Matr.nr. 1:

Nach-,Vorname 1: ,

Matr.nr. 2:

Nach-,Vorname 2: ,

Ausgabe: 22. November 2019

Abgabe: 3. Dezember 2019, 12:30 Uhr
im GBI-Briefkasten im Untergeschoss
von Gebäude 50.34

Lösungen werden nur korrigiert, wenn sie

- rechtzeitig
- handschriftlich
- mit dieser Seite als Deckblatt und
- in der oberen **linken** Ecke zusammengeheftet abgegeben werden.

Vom Tutor auszufüllen: erreichte Punkte

Blatt 6: / 18

Aufgabe 6.1 (3 Punkte)

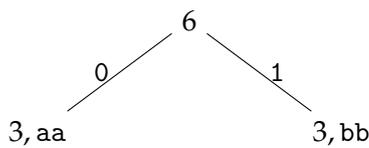
Es sei $A = \{a, b\}$. Geben Sie ein Wort $w \in A^*$ der Länge ≤ 18 an, sodass es eine Block-Codierung von w mit Blöcken der Länge 2 gibt, die echt kürzer als alle Block-Codierungen von w mit Blöcken der Länge 3 ist. Insbesondere muss $|w|$ dabei ein Vielfaches von 6 sein.

Erstellen Sie anschließend Block-Codierungen für Ihre Wahl von w mit Blöcken der Länge 2 bzw. 3 und geben Sie die jeweilige Codierung von w an.

Lösung 6.1

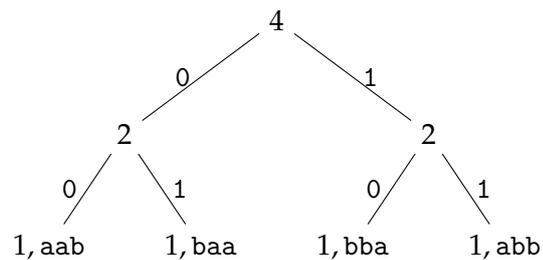
Z. B. $w = aabbaabbaabb = aa \cdot bb \cdot aa \cdot bb \cdot aa \cdot bb = aab \cdot baa \cdot bba \cdot abb$

Huffman-Bäume und -Codierungen:



aa	0
bb	1

Codierung von w : 010101



aab	00
baa	01
bba	10
abb	11

Codierung von w : 00011011

Aufgabe 6.2 (2 + 2 = 4 Punkte)

Der MIMA-Befehlssatz soll um die folgenden zwei Befehle erweitert werden:

- INC a addiert 1 zu dem Wert, der an Adresse a in Zweierkomplementdarstellung gespeichert ist.
- DEC a zieht 1 vom Wert ab, der an Adresse a in Zweierkomplementdarstellung gespeichert ist.

Die Addition bzw. Subtraktion erfolgt dabei gemäß den Vorschriften der 24-Bit Zweierkomplementdarstellung.

- Geben Sie jeweils zu INC a und DEC a eine Folge von MIMA-Befehlen an, die den Befehl realisiert. Verwenden Sie dabei jeweils höchstens fünf MIMA-Befehle. Sie dürfen ausschließlich die Befehle verwenden, die in der Vorlesung vorgestellt wurden. Sprungbefehle sind nicht zugelassen.
- Es seien jetzt $n_x, n_y \in \mathbb{K}_{24}$ nicht-negative Werte, die kleiner gleich 2^{10} sind, und die jeweils in Zweierkomplementdarstellung an den 20-Bit Adressen x und y gespeichert sind, wobei $x \neq y$. Betrachten Sie folgendes MIMA-Programm:

```

loop:  LDC 0
       EQL y
       JMN end
       INC x
       DEC y
       JMP loop
end:   HALT

```

Geben Sie die Werte an, die nach Ausführung des Programms an den Adressen x und y gespeichert sind.

Lösung 6.2

a)	INC a :	LDC 1	DEC a :	LDC 0
		ADD a		NOT
		STV a		ADD a
				STV a

b) Adresse x : $n_x + n_y$
 Adresse y : 0

Aufgabe 6.3 (2 Punkte)

Es sei a eine (feste) 20-Bit Adresse. Geben Sie eine Folge von höchstens acht MIMA-Befehlen an, die die Inhalte der von a und b adressierten Speicherzellen vertauscht, wobei b eine unbekannte Adresse ist, die vor Ausführung Ihrer Befehlsfolge im Akkumulator steht. Sie dürfen dabei annehmen, dass $a, b \notin \{100, 101, 102\}$ ist und die Speicherzellen 100, 101, und 102 als Zwischenspeicher benutzen. Außerdem dürfen Sie dabei keine Sprungbefehle verwenden. Ihre Befehlsfolge muss keinen HALT Befehl enthalten.

Lösung 6.3

```

STV 100  «M(100) ← b»
LDIV 100
STV 101  «M(101) ← M(M(100)) = M(b)»
LDV a
STIV 100 «M(M(100)) = M(b) ← M(a)»
LDV 101
STV a    «M(a) ← M(101)»

```

Aufgabe 6.4 (2 + 2 = 4 Punkte)

Es seien x und y zwei verschiedene 20-Bit Adressen.

Für $w \in \{0, 1\}^*$ sei $N_1(w)$ die Anzahl der 1 in w . Es sei P ein Programm, das $N_1(b_x)$ berechnet und in Zweierkomplementdarstellung an Adresse y ablegt, wenn $b_x \in \{0, 1\}^{24}$ zu Beginn an Adresse x steht.

- a) Es mögen nun an den Adressen x und y zwei Bitfolgen b_x und b_y aus $\{0, 1\}^{24}$ stehen. Betrachten Sie folgendes MIMA-Programm P_1 :

```
LDV x
XOR y
STV x
```

Angenommen, P_1 und P werden (in dieser Reihenfolge) ausgeführt, was ist der kleinste Wert (über alle mögliche b_x und b_y), der danach an der Adresse y stehen kann? Begründen Sie Ihre Antwort. Geben Sie anschließend alle Paare $(b_x, b_y) \in \{0, 1\}^{24} \times \{0, 1\}^{24}$ an, die zu diesem minimalen Wert führen.

- b) Geben Sie ein MIMA-Programm P_2 mit höchstens vier Anweisungen an, sodass wenn P_2 und P (in dieser Reihenfolge) ausgeführt werden, am Ende der Wert $\text{Zkpl}_{24}(24 - N_1(b_x))$ an der Adresse y steht. Ihr Programm darf dabei keine Sprungbefehle enthalten. Begründen Sie anschließend, warum Ihr Programm das Geforderte tut.

Lösung 6.4

- a) Null. Das ist genau dann der Fall, wenn $b_x = b_y$ ist (für $b_x \in \{0, 1\}^{24}$ beliebig); denn $N_1(b_x)$ ist minimal bei $b_x = 0^{24}$ und es gilt $\text{XOR}(b_x, b_y) = 0^{24}$ genau dann, wenn $b_x = b_y$ ist.
- b) Es gilt $N_1(\text{NOT}(b_x)) = 24 - N_1(b_x)$, weil $|b_x| = 24$ ist und NOT alle Bits von b_x umkippt. Also:

```
LDV x
NOT
STV x
```

Aufgabe 6.5 (1.5 + 2.5 + 1 = 5 Punkte)

Es seien x und y (verschiedene) 20-Bit Adressen. Betrachten Sie folgendes MIMA-Programm:

```
LDC 0
STV x
STV y
A: LDC 1
ADD x
STV x
LDC 1
NOT
AND x
B: RAR
XOR x
STV y
JMP A
```

- a) Betrachten Sie den Codeblock zwischen den Beschriftungen A und B (beides einschließlich). Angenommen, es wäre vor dessen Ausführung der Wert $Zkpl_{24}(n)$ an Adresse x gespeichert, wobei $n \in \mathbb{N}_0$ und $n \leq 2^{10}$ ist. Geben Sie den Wert an, der nach Ausführung des Blocks im Akkumulator steht.
- b) Es seien x_i und y_i die Werte (in Binärdarstellung), die direkt vor der i -ten Ausführung der mit A beschrifteten Anweisung an den Adressen x und y stehen, wobei $i \in \mathbb{N}_+$ ist. Geben Sie x_i und y_i für $i \leq 8$ tabellarisch an. Es reicht, wenn Sie dabei nur die vier niedrigstwertigen Stellen angeben.
- c) An wie vielen Stellen unterscheiden sich y_i und y_{i+1} ? Wie kann man die aus der Binärdarstellung von i ablesen?

Lösung 6.5

Die Folge binärer Wörter, die von diesem Programm erzeugt werden, heißt (*gespiegelter*) *Gray-Code* (nach dem amerikanischen Physiker Frank Gray).

a) $(n + 1) \text{ div } 2$

b)

i	1	2	3	4	5	6	7	8
x_i	0000	0001	0010	0011	0100	0101	0110	0111
y_i	0000	0001	0011	0010	0110	0111	0101	0100

c) An genau eine Stelle, nämlich (für $i > 0$) an der kleinsten Stelle j in der Binärdarstellung von i , an der eine 1 steht, und für die alle Bits in den Stellen weiter rechts von j gleich 0 sind. (Für $i = 0$ handelt es sich um die niedrigstwertige Stelle.)