

**Universität Karlsruhe**  
Institut für Betriebs- und Dialogsysteme  
Prof. Dr. Alfred Schmitt  
Dipl.-Inform. B. Klimmek  
Samstag, den 26. Juli 2003

## Klausur Informatik IV

Sommersemester 2003

Name: \_\_\_\_\_  
Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

**Beachten Sie:**

- Schreiben Sie Ihren **Namen**, Ihren **Vornamen** und Ihre **Matrikelnummer** gut lesbar (Druckschrift!) sowohl auf dieses Deckblatt als auch auf jedes bearbeitete Aufgabenblatt.
- Schreiben Sie Ihre Lösungen auf die Aufgabenblätter. Benutzen Sie gegebenenfalls die Rückseiten. Notfalls können Sie weiteres Papier anfordern.
- Zum **Bestehen** der Klausur sind **20** der möglichen **60 Punkte** hinreichend.
- Es sind **keine Hilfsmittel** zugelassen.

Aufgabe	1	2	3	4	5	6	7	8	Gesamt
Erreichte Punkte									
Mögliche Punkte	6	5	5	6	11	10	6	11	<b>60</b>

Note	Bonus	Endnote

### Aufgabe 1

Beweisen Sie für zwei Funktionen  $f : \mathbb{N} \rightarrow \mathbb{N}$  und  $g : \mathbb{N} \rightarrow \mathbb{N}$  anhand der Definition des O-Kalküls:

$$\max\{f, g\} \in \Theta(f + g)$$

Offenbar gilt für alle nichtnegativen Zahlen  $a, b$

Mittelwert  $\leq$  Maximum  $\leq$  Maximum + Minimum

Damit gilt insbesondere für die Funktionswerte von  $f$  und  $g$ :

$$\forall n \in \mathbb{N} : \frac{f(n) + g(n)}{2} \leq \max\{f(n), g(n)\} \leq f(n) + g(n)$$

Für  $C_1 = 1/2$  und  $C_2 = 1$  und  $n_0 = 1$  gilt dann also

$$\exists C_1, C_2 > 0 : \exists n_0 \in \mathbb{N} : \forall n \geq n_0 :$$

$$C_1 \cdot (f(n) + g(n)) \leq \max\{f(n), g(n)\} \leq C_2 \cdot (f(n) + g(n))$$

Und dies ist gleichbedeutend mit

$$\max\{f, g\} \in \Theta(f + g)$$

q.e.d.

## Aufgabe 2

Beweisen Sie, daß der Tetraeder (Pyramide mit dreieckiger Grundfläche) mit den Eckpunkten

$$\mathbf{p}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{p}_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{p}_3 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{p}_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

eine rationale Menge im  $\mathbb{R}^3$  ist, indem Sie ihn in die Standardform für rationale Mengen überführen.

Je drei Punkte geben die Lage einer der vier Begrenzungsflächen an. Jede dieser vier Ebenen teilt den  $\mathbb{R}^3$  in zwei Halbräume. Der Schnitt der richtigen vier Halbräume ist schließlich das Volumen des Tetraeders.

Für den Schnitt durch die Grundfläche (die ja die  $xy$ -Ebene ist) läßt sich die einfache Bedingung

$$B_1(x, y, z) = z \geq 0$$

aufstellen.

Analog gilt für die Schnitte durch zwei der Seitenflächen ( $yz$ -Ebene bzw.  $xz$ -Ebene) offenbar

$$B_2(x, y, z) = x \geq 0$$

$$B_3(x, y, z) = y \geq 0$$

Für den Schnitt durch die letzte (diagonale) Seitenfläche muß gelten

$$x + y + z \leq 1$$

Also:

$$B_4(x, y, z) = 1 - x - y - z \geq 0$$

Damit sind alle vier Bedingungen definiert (die  $B_i$  sind rationale Funktionen!) und die folgende Standardform für rationale Mengen beschreibt den Tetraeder:

$$M = \{ (x, y, z) \in \mathbb{R}^3 \mid B_1(x, y, z) \geq 0 \wedge B_2(x, y, z) \geq 0 \\ \wedge B_3(x, y, z) \geq 0 \wedge B_4(x, y, z) \geq 0 \}$$

### Aufgabe 3

Gegeben seien  $2n$  Fächer (mit den Indizes  $1, 2, \dots, 2n-1, 2n$ ) und  $2n$  Kugeln. Die Kugeln werden zufällig, und zwar gleichverteilt, in die Fächer geworfen.

Wie groß ist die Wahrscheinlichkeit, daß alle Fächer mit geradem Index (also die Fächer  $2, 4, \dots, 2n-2, 2n$ ) leer bleiben?

Beschreiben Sie kurz den Rechenweg.

Für jede einzelne geworfene Kugel ist die Wahrscheinlichkeit, daß ein Fach mit ungeradem Index getroffen wird,  $= 1/2$ .

Dieser Vorgang wiederholt sich  $2n$  mal. Da die  $2n$  Würfe aber unabhängig voneinander geschehen, dürfen die Wahrscheinlichkeiten multipliziert werden; die gesuchte Wahrscheinlichkeit ist das Produkt:

$$p = \prod_{i=1}^{2n} \frac{1}{2} = \left(\frac{1}{2}\right)^{2n} = \frac{1}{4^n}$$

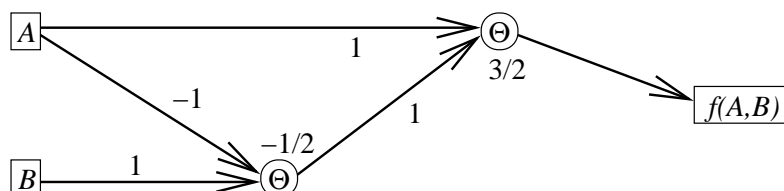
Bemerkung:

Der Ansatz mit Indexfolgen führt zum selben Ziel:

$$p = \frac{\text{Anz. günstiger Fälle}}{\text{Anz. möglicher Fälle}} = \frac{n^{2n}}{(2n)^{2n}} = \left(\frac{n}{2n}\right)^{2n} = \left(\frac{1}{2}\right)^{2n} = \frac{1}{4^n}$$

#### Aufgabe 4

Welche boolesche Funktion  $f$  wird durch das folgende neuronale Netz modelliert?



Läßt sich  $f$  auch mit nur einem Neuron modellieren? (Wenn ja: Skizze! Wenn nein: Begründung!)

Sowohl für die Eingaben als auch für die Ausgabe des Netzes gelte:

$$\text{TRUE} \equiv 1 \quad \text{und} \quad \text{FALSE} \equiv 0$$

(Ein Neuron sei folgendermaßen beschaffen: ist die gewichtete Summe der Eingaben größer als sein Schwellwert  $\Theta$ , so sei die Ausgabe des Neurons Eins; ist die gewichtete Summe  $< \Theta$ , so sei die Ausgabe Null; andernfalls ist die Ausgabe als undefiniert anzusehen!)

- Das “untere” Neuron mit dem Schwellwert  $\Theta = -1/2$  berechnet offenbar die logische Implikation  $A \rightarrow B$ :

$A$	$B$	$Ausgabe$
0	0	1
0	1	1
1	0	0
1	1	1

- Das “obere” Neuron mit dem Schwellwert  $\Theta = 3/2$  berechnet die logische Konjunktion (“und”-Verknüpfung): beide Eingänge müssen den Wert = 1 annehmen, damit es aktiviert wird.

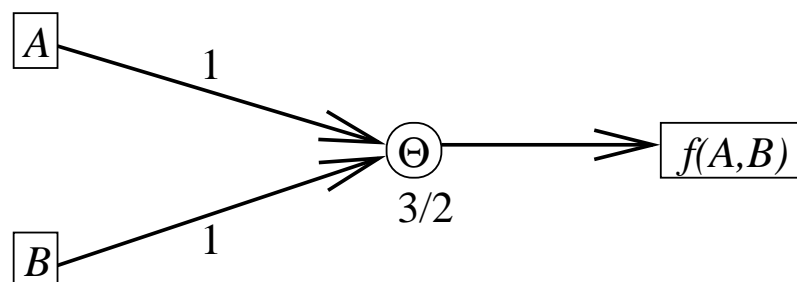
Insgesamt berechnet das obige Netz also die Funktion

$$f(A, B) = (A \rightarrow B) \wedge A$$

Sie läßt sich umschreiben als

$$\begin{aligned} &= (B \vee \neg A) \wedge A \stackrel{\text{distrib.}}{=} (B \wedge A) \vee (\neg A \wedge A) \\ &= (B \wedge A) \vee \text{FALSE} = B \wedge A \end{aligned}$$

Diese einfache Konjunktion läßt sich aber bekanntlich mit nur einem einzigen Neuron darstellen:



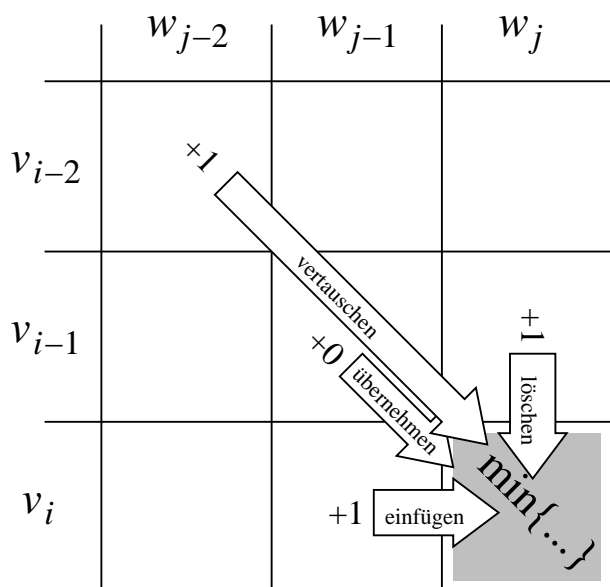
### Aufgabe 5

Betrachten Sie den Editierabstand von Zeichenketten, wenn als einzige Elementaroperationen das Zeichen-Einfügen, das Zeichen-Löschen und die schwächere Variante des Zeichen-Vertauschens erlaubt sind.

“Schwächere Variante” heißt hier, daß die zu vertauschenden Zeichen sowohl in  $v$  als auch in  $w$  direkt benachbart vorkommen müssen.

- a) Geben Sie ein Schema an, nach dem ein Matrixelement  $(i, j)$  aus benachbarten Matrixelementen ermittelt werden kann. Es sei Wort  $v$  in Wort  $w$  zu überführen.

$$A(i, j) = \min \left\{ \begin{array}{ll} A(i-1, j) + 1 & \text{in } w \text{ einfügen} \\ A(i, j-1) + 1 & \text{in } v \text{ löschen} \\ A(i-1, j-1) & \Leftrightarrow \left\{ \begin{array}{l} v_i = w_j \\ \text{übernehmen} \end{array} \right. \\ A(i-2, j-2) + 1 & \Leftrightarrow \left\{ \begin{array}{l} v_{i-1} = w_j \\ v_i = w_{j-1} \\ \text{vertauschen} \end{array} \right. \end{array} \right.$$



(6 Punkte)

b) Bestimmen Sie den Editierabstand zwischen ANANAS und BANANEN:

		B	A	N	A	N	E	N
	0	1	2	3	4	5	6	7
A	1	2	1	2	3	4	5	6
N	2	3	2	1	2	3	4	5
A	3	4	3	2	1	2	3	4
N	4	5	4	3	2	1	2	3
A	5	6	5	4	3	2	3	4
S	6	7	6	5	4	3	4	<b>5</b>

Das Endergebnis lautet also: 5 Editieroperationen.<sup>1</sup>

(5 Punkte)

11 Punkte

---

<sup>1</sup>Die "vertauschen"-Regel kommt in der obigen Matrix kein einziges Mal erfolgreich zur Anwendung.

## Aufgabe 6

Die Zeichenkette BLABLABLABLA soll binär kodiert werden.

- a) Wie viele Bits enthält eine (zeichenweise) Huffman-Kodierung der Zeichenkette BLABLABLABLA?

Auftrittswahrscheinlichkeiten:

Zeichen $x$	A	B	L
$p(x)$	1/3	1/3	1/3

Da alle drei Zeichen gleich oft auftreten, ist es egal, welches von ihnen das “privilegierte” 1-stellige Kodewort bekommt. Jedenfalls bekommen die beiden anderen Zeichen 2-stellige Kodewörter zugeordnet.

Der Erwartungswert an Kodelänge ist pro Zeichen also  $\frac{1}{3} \cdot 1 \text{ bit} + \frac{1}{3} \cdot 2 \text{ bit} + \frac{1}{3} \cdot 2 \text{ bit} = \frac{5}{3} \text{ bit}$ . Da die Zeichenkette aus 12 Zeichen besteht, wird die Länge der Huffman-Kodierung also in jedem Falle

$$12 \cdot \frac{5}{3} \text{ bit} = 20 \text{ bit}$$

sein.

(3 Punkte)

- b) Kodieren Sie die Zeichenkette BLABLABLABLA nach dem Lempel-Ziv-Welch-Verfahren (LZW). Die Kodierung des Eingabe-Alphabets sei durch

Eingabe	A	B	L
Ausgabe	0	1	2

gegeben. Lassen Sie also weitere Kodierungen im Wörterbuch bei Kode-Nr. 3 anfangen:

Eingabe	Ausgabe	Wörterbuch
B	1	3=BL
L	2	4=LA
A	0	5=AB
B	3	6=BLA
L		
A	5	7=ABL
⋮	⋮	⋮

⋮	⋮	⋮
B		
L	4	8=LAB
A		
B	6	
L		
A		
Eingabe	Ausgabe	Wörterbuch

Wie viele Bits benötigen Sie bei dieser LZW-Kodierung pro Zeichen des Ausgabealphabets, wenn die Ausgabe – wie üblich – im Blockcode (d. h. mit konstanter Kodelänge) kodiert wird?

Da die höchste ausgegebene Kode-Nr. 6 ist (Ausgabezeichen also  $\in \{0, \dots, 6\}$ ), reichen 3 Bits pro Ausgabe-Kodewort.

Wie lang ist also die Kodierung der gesamten Zeichenkette?

Die Ausgabe “1203546” umfaßt 7 Zeichen; die Länge der Bit-Kodierung ist also  $7 \cdot 3 \text{ bit} = 21 \text{ bit}$ .

(4 Punkte)

- c) Bestimmen Sie die absolute Redundanz der Kodierungen aus (a) und (b), indem Sie (nach Shannon) die Realinformation  $H$  der ursprünglichen Zeichenkette mit der Nominalinformation (Kodelänge)  $\tilde{H}$  ihrer jeweiligen binären Kodierung verrechnen.

Eine Tabelle für die numerischen Werte von  $-p \cdot \log p$  finden Sie im Anhang.

Die Realinformation pro Zeichen beläuft sich nach Shannon auf:

$$\begin{aligned} H(X) &= H\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) = -3 \cdot \frac{1}{3} \log \frac{1}{3} \\ &\approx 3 \cdot 0,528 \text{ bit/Zeichen} = 1,584 \text{ bit/Zeichen} \end{aligned}$$

Damit ist für die Zeichenkette

$$\begin{aligned} H &= 12 \cdot H(X) \approx 12 \text{ Zeichen} \cdot 1,584 \text{ bit/Zeichen} \\ &= 19,008 \text{ bit} \approx 19,0 \text{ bit} \end{aligned}$$

Die absolute Redundanz der Huffman-Kodierung (a) der Zeichenkette ist also

$$R = \tilde{H} - H = 20 \text{ bit} - 19,0 \text{ bit} = 1,0 \text{ bit}$$

und die absolute Redundanz der LZW-Kodierung (b) der Zeichenkette ist

$$R = \tilde{H} - H = 21 \text{ bit} - 19,0 \text{ bit} = 2,0 \text{ bit}$$

(3 Punkte)

10 Punkte

### Aufgabe 7

Mit  $x^*$  sei die komplex Konjugierte einer komplexen Zahl  $x$  bezeichnet.

Zeigen Sie, daß nur für Zahlen  $x$  auf dem Einheitskreis in der komplexen Zahlenebene die Beziehung

$$x^{-1} = x^*$$

gilt. (Tip: Euler-Darstellung!)

Achten Sie auf die Beweisrichtung!

Im folgenden seien  $a, b, r, \phi \in \mathbb{R}$  mit  $r = \sqrt{a^2 + b^2} \neq 0$  und  $\tan \phi = b/a$  (für  $a \neq 0$ , sonst  $\cot \phi = a/b$ ). Dann muß gelten

$$x^{-1} = (a + bi)^{-1} \stackrel{!}{=} a - bi = x^*$$

$$(r \cdot e^{\phi i})^{-1} \stackrel{!}{=} a - bi$$

$$r^{-1} \cdot e^{(-\phi)i} \stackrel{!}{=} a - bi$$

$$r^{-1}(\cos(-\phi) + i \sin(-\phi)) \stackrel{!}{=} a - bi$$

$$r^{-1}(\cos \phi - i \sin \phi) \stackrel{!}{=} a - bi$$

$$\underbrace{r^{-1} \cos \phi}_a - i \underbrace{r^{-1} \sin \phi}_b \stackrel{!}{=} \underbrace{r \cos \phi}_a - i \underbrace{r \sin \phi}_b$$

Zwei komplexe Zahlen sind aber genau dann gleich, wenn sie gleiche Real- und Imaginärteile haben. Daher muß  $r^{-1} = r$  sein. Dies ist offenbar nur für  $r = 1$  der Fall. q.e.d.

## Aufgabe 8

Kreuzen Sie für folgende Aussagen an, ob diese wahr oder falsch sind.

**Hinweis:** Für jede richtige Antwort gibt es einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Die gesamte Aufgabe wird mit einer nichtnegativen Punktezahl gewertet.

<i>Aussage</i>	<b>wahr</b>	<b>falsch</b>
Für dünne rationale Mengen $M$ gilt stets $ M  > 0$ .		falsch
Wenn die Höhe eines B-Baumes bei einer Löschoperation um Eins abnimmt, so hat der Wurzelknoten zuvor höchstens 1 Schlüssel enthalten.	wahr	
Invertierte Dateien werden zwar rückwärts gelesen, aber vorwärts geschrieben.		falsch
Udi Manbers IR-Technik <i>Glimpse</i> ist für große Datenbestände (ab ca. 1/2 GByte) nicht mehr geeignet, weil das Durchsuchen der Text-Zonen zu lange dauern würde.		falsch
Der Zelleraster-Algorithmus erlaubt es, die Suche eines nächsten Nachbarn beim Post-Office-Problem in der <i>worst-case</i> -Zeit $T_{\max}(n) \in O(\log n)$ zu realisieren.		falsch
Bei einem binären symmetrischen Kanal mit Fehlerwahrscheinlichkeit $\beta$ hat die Wahrscheinlichkeit $w(10010110   10111001)$ den Wert $\sum_{i=0}^5 \binom{8}{i} (1 - \beta)^{8-i} \beta^i$ .		falsch
Um das Problem der Elementeindeutigkeit zu lösen, genügt eine einzige Vergleichsoperation.	wahr	
Die schnelle Fouriertransformation (FFT) schafft es in $\Theta(n)$ Schritten, von der Stützstellen- in die Koeffizientendarstellung zu transformieren.		falsch
Für B-Bäume gilt: Knotenüberläufe kommen nur bei Insert-Operationen vor; Knotenunterläufe kommen nur bei Delete-Operationen vor.	wahr	
Ein deterministischer Kanal mit $ X  = n$ hat die Kanalkapazität $C = \log_2 n$ .		falsch
In der Optimierung dienen Penalty-Terme dazu, den Suchraum einzuschränken.	wahr	