

Klausur über den Stoff der Vorlesung „Grundlagen der Informatik II“ (90 Minuten)

Name: _____ Vorname: _____

Matr.-Nr.: _____ Semester: _____ (SS 2016)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-17).

Unterschrift des o. g. Klausurteilnehmers
bzw. der o. g. Klausurteilnehmerin

Anmerkungen:

1. Legen Sie bitte Ihren Studierendenausweis bereit.
2. Bitte tragen Sie **Name**, **Vorname** und **Matr.-Nr.** deutlich lesbar ein.
3. Die folgenden **10 Aufgaben** sind vollständig zu bearbeiten.
4. Folgende Hilfsmittel sind zugelassen: **keine**.
5. Täuschungsversuche führen zum Ausschluss von der Klausur.
6. Unleserliche oder mit Bleistift geschriebene Lösungen können von der Klausur bzw. Wertung ausgeschlossen werden.
7. Die Bearbeitungszeit beträgt 90 Minuten.

Nur für den Prüfer :

1	2	3	4	5	6	7	8	9	10	-	-	-	-	-	-	gesamt
(8)	(8)	(9)	(8)	(9)	(10)	(10)	(12)	(8)	(8)							(90)

Aufgabenübersicht

1) Chomsky-Normalform (8 Punkte)	2
2) Reguläre Sprachen (8 Punkte)	4
3) Nichtdeterministischer Kellerautomat (9 Punkte)	5
4) Komplexität (8 Punkte)	6
5) Binary Decision Diagram (9 Punkte)	7
6) Fehlererkenn- und ~korrigierbarkeit (10 Punkte)	9
7) Zahlendarstellung (10 Punkte)	11
8) Assembler und Cache (12 Punkte)	13
9) Betriebssysteme (8 Punkte)	15
10) Hashing (8 Punkte)	17

Aufgabe 1**8 Punkte****2016-N-01****Chomsky-Normalform**

Gegeben sei die folgende kontextfreie Grammatik:

$$G = (\{A, B, C, D, S\}, \{a, b\}, P, S)$$

$$P = \{S \rightarrow a \mid Db \mid aD \mid BAa,$$

$$A \rightarrow a \mid b \mid Ca \mid Sb,$$

$$B \rightarrow b \mid Bb,$$

$$C \rightarrow \lambda \mid Cb,$$

$$D \rightarrow a \mid b \mid Ca \mid SbBS\}$$

(a) Wandeln Sie die Grammatik um in die **Chomsky-Normalform (CNF)**.

(b) Geben Sie **einen Ableitungsbaum** für das Wort $aabbab \in L(G)$ an:

- Entweder mit der Grammatik G ;
- Oder mit Ihrer selbsterstellten CNF-Grammatik aus Aufgabenteil (a).

/ 3

Aufgabe 2**8 Punkte**

2016-N-02

Reguläre Sprachen

/ 8

Der Shuffle zweier Worte $u = u_1 \dots u_m$ und $v = v_1 \dots v_n$ besteht aus allen Wörtern, die sich durch Mischen der Zeichen von u und v erzeugen lassen, wobei die relative Ordnung der Zeichen in den Wörtern erhalten bleibt:

$$\lambda \odot v = \{v\}$$

$$u \odot \lambda = \{u\}$$

$$u \odot v = u_1(u_2 \dots u_m \odot v) \cup v_1(u \odot v_2 \dots v_n)$$

Sei beispielsweise $u = ab$ und $v = xy$, dann gilt:

$$u \odot v = \{abxy, axby, xaby, axyb, xayb, xyab\}.$$

Der Shuffle zweier Sprachen L_1 bzw. L_2 besteht aus allen Wörtern, die sich durch Shuffle aus Wörtern jeweils aus L_1 und L_2 erzeugen lassen:

$$L_1 \odot L_2 = \bigcup_{u \in L_1, v \in L_2} u \odot v.$$

Begründen Sie, dass wenn L_1 und L_2 **regulär** sind, $L_1 \odot L_2$ ebenfalls **regulär** ist.

Hinweis: Nehmen Sie an, Sie kennen zwei deterministische endliche Automaten A_1 und A_2 , die jeweils L_1 und L_2 erkennen. Nutzen Sie A_1 und A_2 , um einen nichtdeterministischen endlichen Automaten zu beschreiben, der $L_1 \odot L_2$ erkennt.

Aufgabe 3 **9 Punkte**

2016-N-03

Nichtdeterministischer Kellerautomat

/ 9

Gegeben sei die Sprache aller Wörter $w \in \{0, 1, \$\}^*$, bei denen **auf jede 0 oder auf jede 1 ein \$ folgt**.

Das heißt, dass in einem Wort der Sprache auf mindestens eines der Zeichen 0, 1 **immer** ein \$ folgen muss; hinter dem jeweiligen anderen Zeichen darf \$ folgen, muss aber nicht. Ansonsten ist die Verteilung von 0, 1 und \$ beliebig. Formal:

$$L = \{w \in \{0, 1, \$\}^* \mid \exists a \in \{0, 1\} : \forall u, v, b \in \{0, 1, \$\}^*, |b| = 1 : w = uabv \implies b = \$\}$$

Es gilt beispielsweise:

$$\begin{aligned} \lambda, 0, 1, 0$, 1$, 0$$$, 0$1, 01, 0$111$0$111, 0$1$0$$, $$$ \in L, \\ 01, 10, 0011, 01, 10$11$00 \notin L \end{aligned}$$

Entwerfen Sie eine **nichtdeterministischen Kellerautomaten**

$$A = \left(\{0, 1, \$\}, \underbrace{\hspace{10em}}_S, \delta, \underbrace{\hspace{2em}}_{s_0}, \underbrace{\hspace{10em}}_F \right)$$

welcher die Sprache L erkennt.

Hinweis: Sie müssen den Keller nicht unbedingt benutzen, um die Sprache zu erkennen. Geben Sie aber auf jeden Fall einen Kellerautomaten an (obwohl prinzipiell auch ein endlicher Automat ausreichend wäre).

Aufgabe 4**8 Punkte****2016-N-04****Komplexität** / 8

Eine **eindeutige Turingmaschine** ist eine Turingmaschine, die in jeder Kombination aus Zustand und Eingabesymbol potentiell mehrere Folgekonfigurationen besitzt, aber eine Eingabe genau dann akzeptiert, wenn nur **ein einziger Pfad** im Berechnungsbaum in einen zulässigen Endzustand mündet.

Bezeichne UP (Unique Polynomial Time) die Klasse der Entscheidungsprobleme, die von einer eindeutigen Turingmaschine in **Polynomialzeit** gelöst werden können.

Hinweis: Überlegen Sie sich, wie sich eindeutige Turingmaschinen von deterministischen und nichtdeterministischen Turingmaschinen unterscheiden.

(a) Begründen Sie, warum $P \subseteq UP$ gilt.

 / 4

(b) Begründen Sie, warum $UP \subseteq NP$ gilt.

 / 4

Aufgabe 5**9 Punkte****2016-N-05****Binary Decision Diagram**

/ 9

Gegeben sei die Funktion $f : \mathbb{B}^3 \rightarrow \mathbb{B}$ durch folgende Wahrheitstabelle:

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

(a) Erstellen Sie das zu f gehörende BDD bei Variablenreihenfolge $a \rightarrow b \rightarrow c$.

/ 5

- (b) Wie könnte eine Funktion $f' : \mathbb{B}^3 \rightarrow \mathbb{B}$ aussehen, die sich als BDD **möglichst stark** komprimieren lässt?

/ 1

a	b	c	f'
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Tragen Sie Ihren Vorschlag für f' in die Wahrheitsabelle ein.

- (c) Wie viele Vereinfachungsschritte (beider Typen laut Algorithmus aus der Vorlesung) werden benötigt, um das BDD zu Ihrer Funktion f' zu erzeugen? Begründen Sie kurz und geben Sie **nur das fertige BDD** an.

/ 3

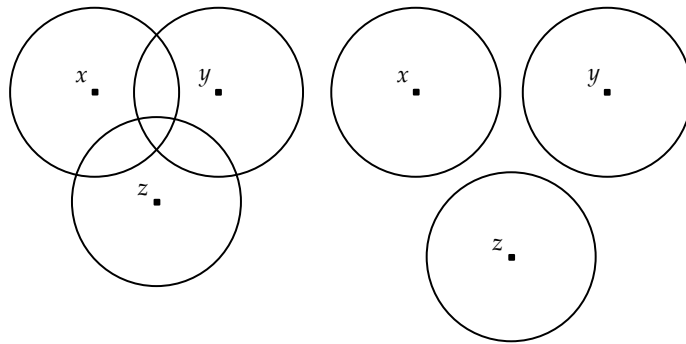
Aufgabe 6 **10 Punkte**

2016-H-06

Fehlererkenn- und ~korrigierbarkeit

/ 10

Gegeben seien folgende zwei schematische Darstellungen der drei Codewörter $x, y, z \in \{0, 1\}^8$ und ihres „Abstands“ voneinander im Rahmen eines Codes. Erklären Sie den Zusammenhang zwischen den euklidischen Abständen in der Abbildung und den Hammingabständen zwischen den Codewörtern in Bezug auf die Behandelbarkeit von Fehlern anhand der folgenden Teilaufgaben.



Erkennbarkeit		
Korrigierbarkeit		

- (a) Kreuzen Sie zunächst in der vorgegebenen Tabelle unterhalb der beiden Darstellungen an, ob diese jeweils für die Erkennbarkeit oder die Korrigierbarkeit von Fehlern steht.

/ 1

- (b) Zeichnen Sie in die Abbildung die für die Fehlerbehandlung relevanten Abstände ein (es reicht jeweils ein Beispiel zur Verdeutlichung). Beschriften Sie deutlich.

/ 3

- (c) Wie lassen sich aus den Abbildungen die Formeln für die k -Fehlererkennbarkeit bzw. die k -Fehlerkorrigierbarkeit von Codes herleiten? Geben Sie die Formeln an und erklären Sie kurz.

/ 4

(d) Die Codewörter seien nun gegeben als

x	0 0 0 0 1 1 1 1
y	0 0 1 1 1 1 0 0
z	1 1 1 1 0 0 0 0

Zeichnen Sie ein neues Diagramm, das die **maximale Korrigierbarkeit** dieser Codewörter darstellt. Wie groß sind die entsprechenden Abstände?

/ 2

Aufgabe 7 **10 Punkte**

2016-N-07

Zahlendarstellung

/ 10

(a) Subtrahieren Sie $B = (33)_{10}$ von $A = (132)_{10}$ **bitweise** in 12-Bit Dualdarstellung.

/ 2

A_2	-----
B_2	-----
$(A - B)_2$	-----

(b) Subtrahieren Sie $C = (50)_{10}$ von $A = (132)_{10}$ **bitweise** in 12-Bit 2-Komplementdarstellung.

/ 2

A_{2K}	-----
C_{2K}	-----
$(A - C)_{2K}$	-----

(c) Wie lautet, dezimal, die betragsmäßig größte Zahl, die mittels einer binären 8-Bit Festkommazahl mit einem Vorzeichenbit, 5 Vorkommastellen und 2 Nachkommastellen dargestellt werden kann?

/ 2

- (d) Die Zahl $N = 1,5 \cdot 2^5$ kann im Gleitkommaformat mit einem Vorzeichenbit, 5 Bit Charakteristik und 3 Bit Mantisse folgendermaßen dargestellt werden:

$$V = 0; c = 10100; m' = 100$$

Multiplizieren Sie N mit sich selbst, ohne Umweg über die Dezimaldarstellung. Unter Berücksichtigung der Formel, mittels derer normierte Gleitpunktzahlen in Dezimalzahlen umgerechnet werden, führen Sie dafür die folgenden Schritte aus:

- Multiplizieren sie die Mantisse (nicht nur m' , sondern m) **bitweise** mit sich selbst.
- Errechnen Sie die Charakteristik aus c und q der ursprünglichen Zahl.
- Bestimmen Sie das neue Vorzeichenbit.
- Geben Sie das Ergebnis N^2 im selben Format an, wie ursprünglich N .

/ 4

Aufgabe 8**12 Punkte**

2016-N-08

Assembler und Cache

/ 12

Die Befehle einer Assembler-Sprache seien folgendermaßen aufgebaut, wobei Q für Quelle steht und Z für Ziel:

OpCode Q1, (Q2,) Z

unmittelbare Adressierung wird durch das Präfix # gekennzeichnet, **indirekte Adressierung** durch *. Der Sprungbefehl JNZ Q label springt genau dann zu label, wenn der Inhalt von Q $\neq 0$ ist.

Das Rechenwerk enthält die Register R1 und R2, die Adressen der Hauptspeicherzellen sind natürliche Zahlen k ($k \in \mathbb{N}_0$). Zwischen Registern und Hauptspeicher befindet sich ein **2-zeiliger Direct-Mapped-Cache** mit **Write-Back-Policy** zur Zwischenspeicherung der geladenen Operanden.

Gegeben sei das folgende Assemblerprogramm.

I		LOAD	#1	R1	
II		LOAD	20	21	
III	loop1	LOAD	#1	R2	
IV		LOAD	21	24	
V	loop2	MULT	24	R2	R2
VI		SUB	24	#1	24
VII		JNZ	24	loop2	
VIII		MULT	R2	R1	R1
IX		STORE	R1	*20	
X		SUB	21	#1	21
XI		JNZ	21	loop1	
XII	end				

- (a) Dokumentieren Sie in der gegebenen Tabelle den Programmablauf, bis zum ersten Mal der Befehl XI erreicht wird. Kennzeichnen Sie für jeden ausgeführten Befehl, ob es dabei zu einem Cache-Hit ($H\langle addr \rangle$) oder Cache-Miss ($M\langle addr \rangle$) kommt. Sie finden sowohl die Tabelle als auch eine Kopie des Programms auf der nächsten Seite.

Hinweis: Die ersten beiden Schritte (Befehlsausführungen) sind bereits beispielhaft eingetragen. Den Inhalt der Register R1 und R2, des Cache und der verwendeten Hauptspeicherzellen können sie als Hilfe notieren, sie werden jedoch bei der Punktevergabe nicht berücksichtigt.

/ 10

```

I      LOAD  #1 R1
II     LOAD  20 21
III    loop1 LOAD #1 R2
IV     LOAD  21 24
V      loop2 MULT 24 R2    R2
VI     SUB   24 #1 24
VII    JNZ   24 loop2
VIII   MULT  R2 R1    R1
IX     STORE R1 *20
X      SUB   21 #1 21
XI     JNZ   21 loop1
XII    end
    
```

Programmablauf

Schritt	Befehl	Hit / Miss
1	I	-
2	II	M 20
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

Hilfstabellen

Rechenwerk

R1	1,
R2	

Cache (direct mapped, write back)

Zeile	Tag-Feld (Hauptspeicheradresse)	Datum
0	20,	2,
1		

Hauptspeicher

Adresse	Datum
20	2,
21	2,

(b) In welchem Schritt, bzw. in welchen Schritten wird der Wert in Speicherzelle 24 verändert?

/ 2

(c) **Bonusaufgabe für drei Zusatzpunkte:** Welche Funktion berechnet das Programm für allgemeine Werte $n \in \mathbb{N}$ an Hauptspeicheradresse 20?

Aufgabe 9 **8 Punkte**

2016-N-09

Betriebssysteme

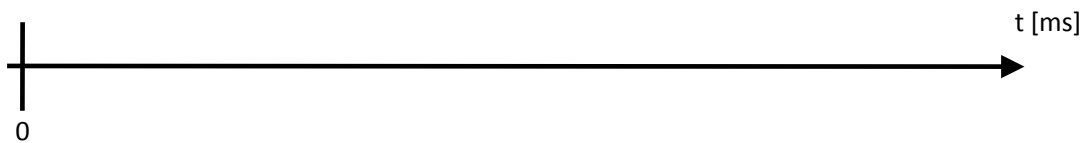
/ 8

In der Warteschlange eines Prozessors befinden sich die folgenden Prozesse P1 bis P4, die in dieser Reihenfolge beim Prozessor ankommen:

Prozesse	CPU-Zeit in ms	Priorität
P1	25	1
P2	30	2
P3	55	2
P4	20	1

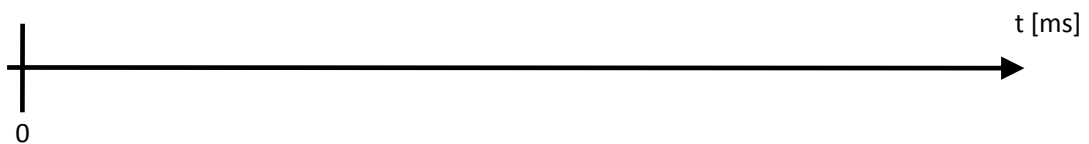
- (a) Teilen Sie den Prozessen Rechenzeit gemäß dem Round Robin Verfahren zu. Die Zeitscheibe sei dabei in feste Zeitspannen der Länge $Z = 15\text{ms}$ unterteilt. Veranschaulichen Sie Ihr Ergebnis auf dem gegebenen Zeitstrahl.

/ 2



- (b) Verwenden Sie nun ein prioritätsbasiertes Round Robin Verfahren, in dem zunächst immer nur den Prozessen der aktuell höchsten Prioritätsstufe Rechenzeit zugewiesen wird. Erst wenn alle Prozesse der höchsten Prioritätsstufe abgearbeitet wurden, wird Prozessen der nächsten Prioritätsstufe Rechenzeit zugeteilt. Die Zeitspannen der Zeitscheibe betragen weiterhin $Z = 15\text{ms}$. Veranschaulichen Sie Ihr Ergebnis auf dem gegebenen Zeitstrahl.

/ 2



- (c) Erklären Sie den Unterschied zwischen **preemptive** und **nonpreemptive** Scheduling. Zu welchem Typ zählt das Round-Robin Verfahren?

/ 2

- (d) Erklären Sie, was ein Deadlock ist und geben Sie eine Möglichkeit an, wie Deadlocks verhindert werden können.

/ 2

Aufgabe 10**8 Punkte****2016-N-10****Hashing**

- (a) Verwenden Sie die Divisionsmethode mit dem Divisor 11 um die folgenden Zahlen in die untenstehende Hashtabelle einzutragen. Behandeln Sie auftretende Kollisionen durch lineares Austesten.

23, 505, 66, 75, 64, 13, 21, 95

RSN	0	1	2	3	4	5	6	7	8	9	10
Schlüssel											

- (b) Nennen und beschreiben Sie kurz eine weitere Hashfunktion, die zur Dateioorganisation geeignet ist.

- (c) Geben Sie eine weitere Möglichkeit zur Kollisionsbehandlung neben dem linearen Austesten an.