

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

Name: _____ Vorname: _____

Matr.-Nr.: _____ Semester: _____ (SS 2019)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-18).

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
(11)	(9)	(9)	(12)	(9)	(9)	(8)	(11)	(9)	(3)							(90)

Aufgabenübersicht

1) Minimierung endlicher Automaten (11 Punkte)	2
2) Kellerautomat (9 Punkte)	5
3) Pumping Lemma (9 Punkte)	7
4) Turingmaschine (12 Punkte)	8
5) Komplexität (9 Punkte)	9
6) Mealy-/Moore-Automat (9 Punkte)	11
7) Huffman (8 Punkte)	13
8) Binary Decision Diagram (11 Punkte)	14
9) Programmiersprachen (9 Punkte)	16
10) Betriebssysteme (3 Punkte)	18

Aufgabe 1 **11 Punkte**

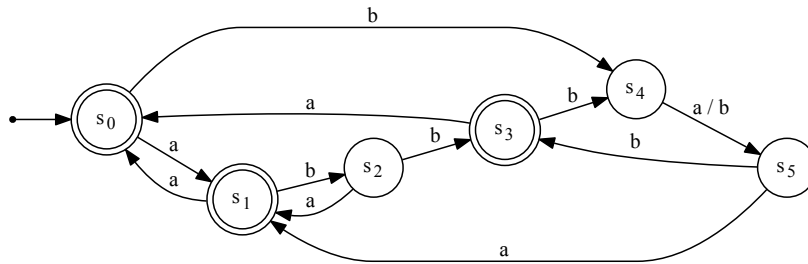
2019-N-01

Minimierung endlicher Automaten

/ 11

Gegeben sei folgender deterministischer endlicher Automat $A = (E, S, \delta, s_0, F)$. Durch das abgebildete Zustandsüberföhrungsdiagramm sei δ definiert.

δ :



- (a) Minimieren Sie A und geben Sie den minimierten Automaten A' vollständig an. Geben Sie insbesondere die Übergangstabellen für A , A' und ein Zustandsüberföhrungsdiagramm δ' an.

/ 7

A	<i>a</i>	<i>b</i>
s_0		
s_1		
s_2		
s_3		
s_4		
s_5		

s_1					
s_2					
s_3					
s_4					
s_5					
	s_0	s_1	s_2	s_3	s_4

A'	a	b

$$A' = \left(\underbrace{\quad}_{E}, \underbrace{\quad}_{S'}, \delta', \underbrace{\quad}_{s_0}, \underbrace{\quad}_{F'} \right)$$

δ' :

- (b) Geben Sie die Mengen aller zueinander k -äquivalenten Zustände für $k \in \{0, 1, 2\}$ und die Mengen äquivalenter Zustände des endlichen Automaten A an. Verwenden Sie hierfür die nachfolgende Tabelle. Geben Sie auch einelementige Mengen an.

/ 4

0-Äquivalenz	
1-Äquivalenz	
2-Äquivalenz	
Äquivalenz	

Aufgabe 2**9 Punkte****2019-N-02****Kellerautomat**

/ 9

Die Sprache $L \subseteq \{ (,), [,] \}^*$ sei gegeben durch folgende Definition:

- (a) $\lambda \in L$,
- (b) falls $w, v \in L$, dann auch: wv , (w) und $[w] \in L$.

L ist also die Sprache aller wohlgeformten Klammerausdrücke über $(,), [$ und $]$.

Hinweis: Bspw. gilt:

$$\lambda, ((())), ()[](([])), [](), (>[]() \in L;$$

$$), ()[], ()[], (,], [[][] \notin L.$$

- (a) Geben Sie einen deterministischen Kellerautomaten $A = (E, S, K, \delta, s_0, k_0, F)$ an mit $L(A) = L$. Geben Sie diesen vollständig an.

/ 7

(b) Zeigen Sie, dass Ihr Kellerautomat das Testwort $() [()]$ akzeptiert.

Aufgabe 3**9 Punkte****2019-N-03****Pumping Lemma** / 9

Gegeben sei die Sprache $L \subset 0^*$ durch $000 \in L$ und $w \in L \rightarrow w^{|w|} \in L$. Zeigen Sie mit Hilfe des Pumping Lemmas, dass L keine Typ 3 Sprache ist.

Hinweis: Es gilt zum Beispiel $000, 000000000 \in L$

Aufgabe 5**9 Punkte****2019-N-05****Komplexität**

- (a) Wie verhält sich die Menge der Sprachen, die von einer deterministischen Turing-Maschine in polynomieller Laufzeit erkannt werden können, zu der Menge der Sprachen, die von einer nichtdeterministischen Turing-Maschine in polynomieller Laufzeit erkannt werden können? Benennen Sie die Mengen.

(b) Zeigen Sie durch Diagonalisierung, dass $p(E^*)$ für ein Alphabet E überabzählbar ist.

Hinweis: $p(E^*)$ entspricht der Menge aller Sprachen über den Wörtern aus E^* .

/ 7

Aufgabe 6**9 Punkte****2019-N-06****Mealy-/Moore-Automat**

/ 9

Im Folgenden soll ein Automat modelliert werden, der Golfbälle zum Trainieren ausgibt. Der zu modellierende Automat nimmt ausschließlich 1€ und 2€ Münzen an. Maximal wird vom Automaten eine Gesamteingabe von 3€ akzeptiert. Sofern dieser Betrag überschritten wird, wird die Differenz direkt zurückgegeben. Über eine Rückgabebetaste kann der Kunde sich das eingezahlte Geld direkt zurückgeben lassen und den Vorgang abbrechen. Kunden können zwischen drei verschiedenen Optionen wählen. Für jede dieser Optionen gibt es am Automaten eine Taste:

- 1) Die Ausgabe von 10 Golfbällen kostet 1€.
- 2) Die Ausgabe von 25 Golfbällen kostet 2€.
- 3) Die Ausgabe von 50 Golfbällen kostet 3€.

Konstruieren Sie den im folgenden Abschnitt beschriebenen Ballautomaten als Mealy-Automat M . Geben Sie M vollständig an. Benutzen Sie dafür das folgende Ein- und Ausgabealphabet (auf der nächsten Seite).

Hinweis: Bei einer Eingabe von beispielsweise 3€ kann der Kunde sich direkt 50 Golfbälle ausgeben lassen. Es ist jedoch auch möglich, sich erst 25 Golfbälle ausgeben und den Restbetrag von 1€ auszahlen zu lassen oder mit diesem Restbetrag 10 weitere Golfbälle zu ordern.

Eingabealphabet $E = \{M_1, M_2, M_U, 10, 25, 50, M_R\}$ mit der Bedeutung:

M_1	1 Euro Münze
M_2	2 Euro Münze
M_U	Ungültige Münze
10	Taste zur Ausgabe von 10 Golfbällen
25	Taste zur Ausgabe von 25 Golfbällen
50	Taste zur Ausgabe von 50 Golfbällen
M_R	Rückgabetaste

Ausgabealphabet $A = \{RU, R1, R2, R3, A10, A25, A50, -\}$ mit der Bedeutung:

RU	Rückgabe von ungültiger Münze
R1	Rückgabe von Einzahlungen in Höhe von 1 Euro
R2	Rückgabe von Einzahlungen in Höhe von 2 Euro
R3	Rückgabe von Einzahlungen in Höhe von 3 Euro
A10	Ausgabe von 10 Golfbällen
A25	Ausgabe von 25 Golfbällen
A50	Ausgabe von 50 Golfbällen
-	Keine Aktion

Aufgabe 7 **8 Punkte**

2019-N-07

Huffman

/ 8

Gegeben sei der folgende Text aus 23 Zeichen:

before-you-can-say-huff

Erzeugen Sie zu der durch den Text gegebenen Wahrscheinlichkeitsverteilung eine Huffman-Kodierung. Tragen Sie dazu die Häufigkeiten der Zeichen in die untere Zeile der ersten Tabelle ein, erstellen Sie einen Huffman-Baum mit Angabe der Häufigkeiten an den Knoten und geben Sie in der zweiten Tabelle für jedes Zeichen eine dem Baum entsprechende Kodierung an.

n	h	e	c	b	a	-	f	y	u	s	r	o

Zeichen	Kodierung
n	
h	
e	
c	
b	
a	
-	
f	
y	
u	
s	
r	
o	

Aufgabe 8

11 Punkte

2019-N-08

Binary Decision Diagram

/ 11

Gegeben sei die Boolesche Funktion $f : \mathbb{B}^3 \rightarrow \mathbb{B}$ mit

$$f(a, b, c) = \overline{(a \oplus b)} \wedge c$$

(a) Tragen Sie in die folgende Tabelle die Funktionswerte der Funktion f ein.

/ 3

a	b	c	$a \oplus b$	$\overline{(a \oplus b)}$	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			

(b) Berechnen Sie das zu f gehörende Binary Decision Diagram (BDD) mit der Variablenreihenfolge $a \rightarrow b \rightarrow c$, indem Sie mit einer Baumdarstellung der Wertetabelle beginnen und den Algorithmus aus der Vorlesung anwenden.

/ 7

(c) Geben Sie die disjunktive Normalform (DNF): $f(a, b, c)$ an.
 $f(a, b, c) =$

/ 1

Aufgabe 9**9 Punkte****2019-N-09****Programmiersprachen**

/ 9

In der Vorlesung haben wir gesehen, dass alle gängigen Programmiersprachen dieselbe Ausdrucksmächtigkeit haben, also einander simulieren können. Im Folgenden seien gegeben:

- eine **GOTO-Sprache** (das sei eine übliche Assemblersprache, die Sprünge in Abhängigkeit des aktuellen Akkumulator-Werts erlaubt: `JNZ label`) und
- eine **WHILE-Sprache** (das sei eine höhere Sprache vergleichbar zu Java, die allgemeine WHILE-Schleifen unterstützt: `while (b) { * do something * }`).

(a) Gegeben sei ein Programmausschnitt in der GOTO-Sprache:

/ 4

```
label:  LOAD R1
        ADD R2
        STORE R1
        LOAD R4
        ADD R1
        STORE R4
        LOAD R3
        SUB #1
        STORE R3
        JNZ label // „JUMP NOT ZERO“
```

Simulieren Sie das Programm durch die WHILE-Sprache, d. h. geben Sie ein äquivalentes WHILE-Programm an.

Hinweise:

- Nutzen Sie möglichst Java-Notation oder einen verständlichen Pseudocode.
- Der Akkumulator sei über ACC erreichbar, die Register über ihre Namen Ri.

- (b) Was müsste im gegebenen Programmausschnitt der **GOTO-Sprache** verändert werden, damit bei Terminierung die Summe $\sum_{x=0}^4 2^x$ in R4 steht?

/ 5

Hinweis: $\sum_{x=0}^4 2^x = 2^0 + 2^1 + 2^2 + 2^3 + 2^4$. Sie können die Lösung auch im gegebenen Code eintragen.

Aufgabe 10**3 Punkte****2019-N-10****Betriebssysteme**

Betrachten Sie die Prozesse P1 bis P4, die nacheinander in die Warteschlange eines Prozessors zur Bearbeitung eingereicht werden.

Prozesse	CPU-Zeit in ms
P1	20
P2	10
P3	25
P4	35

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

