

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

Name: _____ Vorname: _____

Matr.-Nr.: _____ Semester: _____ (SS 2014)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-12).

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 **9 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	-	-	-	-	-	-	-	gesamt
(13)	(8)	(10)	(12)	(10)	(10)	(12)	(8)	(7)								(90)

Aufgabenübersicht

1) Endliche Automaten (13 Punkte)	2
2) Rechtslineare Grammatiken (8 Punkte)	5
3) Pumping-Lemma für EA-Sprachen (10 Punkte)	6
4) Turingmaschine (Busy Beaver) (12 Punkte)	7
5) Binary Decision Diagram (BDD) (10 Punkte)	8
6) Zahlendarstellung (10 Punkte)	9
7) Fehlererkenn- und ~korrigierbarkeit (12 Punkte)	10
8) Assembler (8 Punkte)	11
9) Betriebssysteme (7 Punkte)	12

Aufgabe 1 **13 Punkte**

2014-N-01

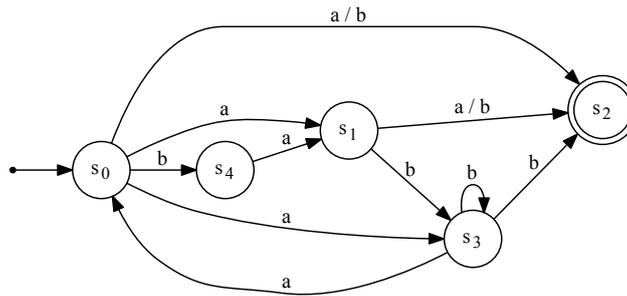
Endliche Automaten

/ 13

Gegeben sei der folgende nichtdeterministische endliche Automat:

$$A = (\{a, b\}, \{s_0, s_1, s_2, s_3, s_4\}, \delta, s_0, \{s_2\})$$

δ :



- (a) Wandeln Sie A durch den Algorithmus aus der Vorlesung in einen äquivalenten deterministischen endlichen Automaten A' um.

/ 8

Hinweis: Geben Sie A' vollständig an. Sie müssen jedoch kein Zustandsüberföhrungsdiagramm zeichnen, eine korrekte Angabe des Tupels und der Tabelle reichen aus.

$A' = (\quad)$

- (b) Je nach Benennung der Zustände kann sich für den deterministischen endlichen Automaten A' folgende Minimierungstabelle ergeben:

s_1	\times_2							
s_2	\times_0	\times_0						
s_3	\times_0	\times_0	\times_1					
s_4	\times_0	\times_0	\times_1	\times_1				
s_5	\times_0	\times_0	\times_2	\times_1	\times_1			
s_6	\times_0	\times_0	\times_1	\times_2	\times_1	\times_1		
s_7	\times_1	\times_1	\times_0	\times_0	\times_0	\times_0	\times_0	
	s_0	s_1	s_2	s_3	s_4	s_5	s_6	

Wie viele Zustände hat demnach ein äquivalenter minimaler endlicher Automat A'' ?

/ 1

- (c) Geben Sie die Mengen zueinander k -äquivalenter Zustände von A' für $k = 0, 1, 2, \dots$ an. Gehen Sie von der Minimierungstabelle in Aufgabenteil (b) aus, **nicht** von Ihrem Automaten in Aufgabenteil (a).

/ 4

Hinweis: Einelementige Mengen müssen Sie nicht angeben.

k	Mengen zueinander k -äquivalenter Zustände
0	
1	
2	
3	
4	
5	

Aufgabe 2 **8 Punkte**

2014-N-02

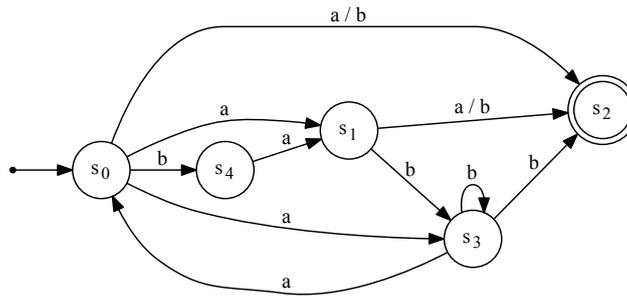
Rechtslineare Grammatiken

/ 8

Gegeben sei wieder der nichtdeterministische endliche Automat:

$$A = (\{a, b\}, \{s_0, s_1, s_2, s_3, s_4\}, \delta, s_0, \{s_2\})$$

δ :



Geben Sie eine rechtslineare Grammatik G an mit $L(G) = L(A)$. Geben Sie die Grammatik vollständig an.

Hinweis: Es könnte hilfreich sein, die Nonterminale ähnlich wie die Zustände des Automaten zu benennen, also etwa $N = \{S_0, \dots, S_4\}$.

$$G = (\underbrace{\{ \quad \}}_N, \underbrace{\{ \quad \}}_T, P, \underbrace{\{ \quad \}}_S)$$

$$P = \{$$

Aufgabe 3**10 Punkte****2014-N-03****Pumping-Lemma für EA-Sprachen**

/ 10

Gegeben sei die Sprache $L \subset \{1\}^*$ durch

$$111 \in L$$

und

$$w \in L \Rightarrow w^{|w|} \in L$$

Es gilt also:

$$111, \underbrace{111 \ 111 \ 111}_{3 \cdot 3 = 9}, \underbrace{1 \dots 1}_{9 \cdot 9 = 81}, \dots \in L$$

Zeigen Sie mithilfe des Pumping-Lemmas für EA-Sprachen, dass $L \notin L_3$.

Aufgabe 4 **12 Punkte**

2014-N-04

Turingmaschine (Busy Beaver)

--

 / 12

Diese Aufgabe kann zeitaufwändig sein, es könnte sinnvoll sein, sie am Ende zu bearbeiten.

Geben Sie eine Turingmaschine an, die

- genau fünf Zustände und das Bandalphabet $\{1, \star\}$ hat,
- bei leerer Eingabe **nach endlich vielen Schritten anhält** und
- dabei **möglichst viele Einsen auf das Band schreibt** (es zählt die Bandinschrift am Ende der Rechnung; Lücken sind erlaubt: „1 \star 11“).

Geben Sie die Turingmaschine vollständig an.

Hinweise:

- Damit die Turingmaschine nicht in eine Endlosschleife gerät, muss mindestens ein Zustand-Bandsymbol-Paar undefiniert bleiben.
- Die volle Punktzahl erhalten Sie, wenn die Turingmaschine **mindestens 7 Einsen** auf das Band schreibt; für 6 Einsen gibt es noch 10 Punkte.
- Sie erhalten **4 Zusatzpunkte**, wenn bei dieser Aufgabe kein Klausurteilnehmer mehr Einsen schafft als Sie.

$T = (\quad)$

δ	1	\star
s_0		
s_1		
s_2		
s_3		
s_4		

Aufgabe 5

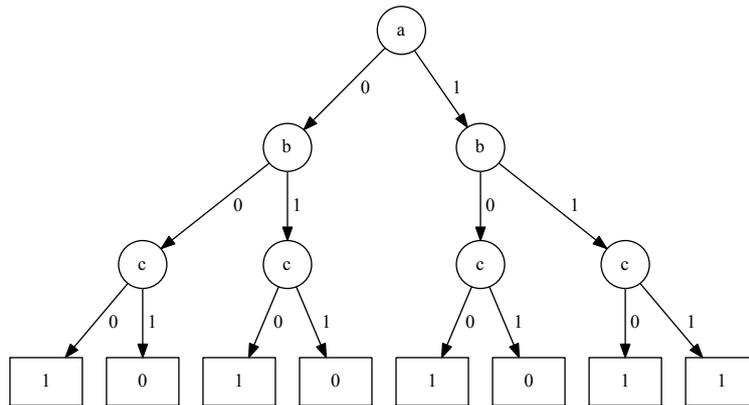
10 Punkte

2014-N-05

Binary Decision Diagram (BDD)

/ 10

Gegeben sei eine Boolesche Funktion $f : \mathbb{B}^3 \rightarrow \mathbb{B}$ durch folgenden Baum:



Generieren Sie mit dem Algorithmus aus der Vorlesung ein BDD zur Funktion f .

Aufgabe 6**10 Punkte****2014-N-06****Zahlendarstellung**

Gegeben sei der folgende Binärstring (Leerzeichen dienen der Lesbarkeit):

0000 0010 1001 0000 0000 0000 1000 0011

Welcher Zahlenwert ergibt sich, wenn man den String interpretiert als (es genügen jeweils Terme aus Summen von Zweierpotenzen etc.)

(a) Zahl in Einskomplement-Darstellung.

(b) Zahl in Zweikomplement-Darstellung.

(c) Zahl in Festkomma-Darstellung mit Komma in der Mitte.

(d) Zahl in Gleitpunkt-Darstellung nach IEEE-754.

(e) Nennen Sie jeweils einen Vorteil einer Gleitkommadarstellung gegenüber einer Festkommadarstellung und umgekehrt.

Aufgabe 7**12 Punkte**

2014-N-07

Fehlererkenn- und ~korrigierbarkeit

/ 12

Gegeben sei ein Code C von Wörtern der Länge 8 durch:

$$0101\ 0101 \in C$$

und für alle $s, t \in \{0, 1\}^*$, $a, b \in \{0, 1\}$:

$$sabt \in C \Rightarrow \begin{cases} sbat \in C, & \text{falls } |s| = |t|, \\ s00t \in C, & \text{falls } |s| < |t|, \\ s11t \in C, & \text{falls } |s| > |t| \end{cases}$$

(a) Geben Sie sechs verschiedene Codewörter $c_1, \dots, c_6 \in C$ an.

/ 3

(b) Wie groß ist die Hammingzahl des Codes C (begründen Sie kurz)?

/ 2

Gegeben sei ein zweiter Code C' durch

$$0101\ 0101 \in C'$$

und für alle $s, t \in \{0, 1\}^*$, $a, b \in \{0, 1\}$:

$$sabt \in C' \Rightarrow a^{|s|+1}b^{|t|+1} \in C', \text{ falls } a \neq b$$

(c) Geben Sie alle Codewörter aus C' an.

/ 3

(d) Geben Sie die Hammingzahl von C' an.

/ 2

(e) Geben Sie die Fehlererkenn- und ~korrigierbarkeit von C' an.

/ 2

Aufgabe 8**8 Punkte**

2014-N-08

Assembler

Gegeben sei das folgende Assemblerprogramm:

LOOP	LOAD	R1
	JUMPZERO	FERTIG
	ADD	R1 R2 R2
	SUBTRACT	R1 #1 R1
	JUMP	LOOP
FERTIG	HALT	

Hinweise:

- Für unmittelbare Adressierung wird das Präfix # genutzt, kein Präfix steht für direkte Adressierung.
- Der letzte von mehreren Operanden bezeichnet jeweils die Zieladresse.
- Sie können davon ausgehen, dass alle Register, außer $R1$, mit 0 initialisiert sind.

Das Register $R1$ ist zu Beginn mit einem Wert $n \in \mathbb{N}_0$ belegt. Der Wert $f(n)$ sei definiert als Ergebnis der Rechnung, das nach Terminierung des Programms in Register $R2$ steht.

- (a) Geben Sie für jedes $n \in \{1, \dots, 4\}$ jeweils den Wert $f(n)$ an.

- (b) Geben Sie eine allgemeine Definition für $f(n)$ an.

- (c) Wie viele Assembler-Befehle (Programmzeilen) werden bei der Berechnung von $f(n)$ abgearbeitet? (Geben Sie den genauen Wert an, nicht nur eine asymptotische Abschätzung.)

Aufgabe 9**7 Punkte****2014-N-09****Betriebssysteme**

- (a) Nennen Sie die zwei in der Vorlesung betrachteten Klassen von Verfahren, mit denen ein Betriebssystem Rechenzeit auf wartende Prozesse zuteilen kann.

- (b) Nennen Sie für jede der Klassen ein Beispiel und erklären Sie jeweils kurz das zugrunde liegende Prinzip und mindestens einen Vorteil.

- (c) Was versteht man in diesem Zusammenhang unter „preemptive Scheduling“ bzw. „non-preemptive Scheduling“?