

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

Name: _____ Vorname: _____

Matr.-Nr.: _____ Semester: _____ (SS 2017)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-24).

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 **11 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 | 11 | - | - | - | - | - | gesamt |
| (8) | (12) | (9) | (8) | (8) | (6) | (6) | (10) | (8) | (8) | (7) | | | | | | (90) |

Aufgabenübersicht

| | |
|--|----|
| 1) Endliche Automaten (8 Punkte) | 3 |
| 2) Kellerautomaten (12 Punkte) | 5 |
| 3) Grammatiken (9 Punkte) | 7 |
| 4) Pumping Lemma für kontextfreie Sprachen (8 Punkte) | 9 |
| 5) Berechenbarkeit (8 Punkte) | 10 |
| 6) Binary Decision Diagram (6 Punkte) | 13 |
| 7) CMOS (6 Punkte) | 15 |
| 8) Zahlendarstellung (10 Punkte) | 17 |
| 9) Rechnerarchitektur (8 Punkte) | 19 |
| 10) Adressierungsarten (8 Punkte) | 21 |
| 11) Betriebssysteme (7 Punkte) | 23 |

Aufgabe 1 **8 Punkte**

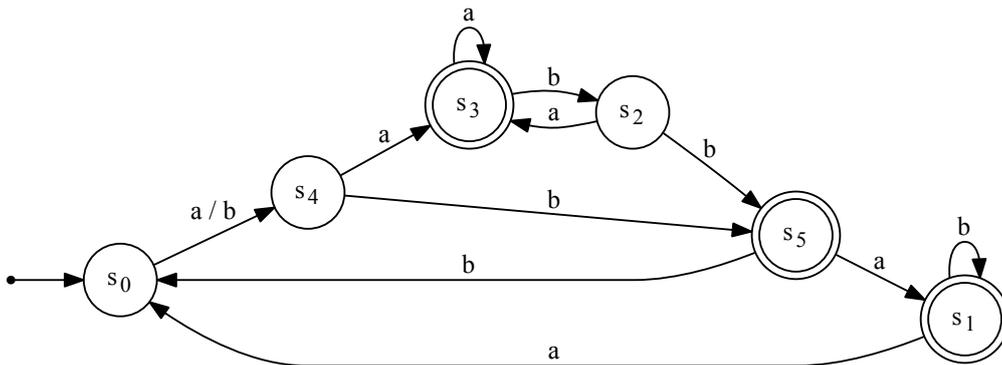
2017-N-01

Minimierung endlicher Automaten

/ 8

Gegeben sei folgender deterministischer endlicher Automaten $A = (E, S, \delta, s_0, F)$. 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 Minimierungstabelle und ein Zustandsüberföhrungsdiagramm δ' an.

/ 5

| | | |
|-------|-----|-----|
| | a | b |
| 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 |

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

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

12 Punkte

2017-N-02

Reguläre Ausdrücke

/ 12

(a) Gegeben sei die rechtslineare Grammatik

$$G_1 = (\{S, A, B, C\}, \{a, b, c, d\}, P, S)$$

$$P = \{S \rightarrow aA \mid \lambda,$$

$$A \rightarrow aA \mid bB \mid cC,$$

$$B \rightarrow bB \mid cC \mid \lambda,$$

$$C \rightarrow cC \mid dS\}$$

Geben Sie einen äquivalenten regulären Ausdruck α an, sodass $L(\alpha) = L(G_1)$.

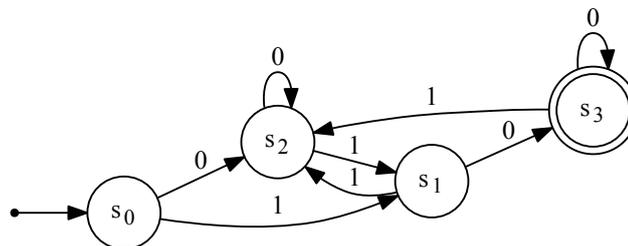
/ 4

α :

(b) Gegeben sei der endliche Automat

$$A = (E, S, \delta, s_0, F)$$

δ :



Geben Sie eine äquivalente rechtslineare Grammatik G_2 an, sodass $L(G_2) = L(A)$.

/ 3

Aufgabe 3 **9 Punkte**

2017-N-03

Turingmaschinen

/ 9

Gegeben sei für $E = \{a, b\}$ die Funktion:

$$f : E^* \rightarrow E^* : f(vx) = xv x \text{ für alle } x \in E, v \in E^*$$

Die Funktion erzeugt also aus einem Wort $w = vx$ (das mit einer beliebigen Zeichenfolge v beginnt und mit dem einzelnen Zeichen x endet) ein Wort, das mit x beginnt und danach w wiederholt. Bspw. gilt:

$$\begin{aligned} f(\lambda) &= \text{undef.} \\ f(b) &= bb \\ f(ab) &= bab \\ f(aaba) &= aaaba \\ f(abaabba) &= aabaabba \end{aligned}$$

(a) Geben Sie eine **deterministische** Turingmaschine T an mit $f_T = f$.

$$T = \left(E, \underbrace{\quad\quad\quad}_B, \underbrace{\quad\quad\quad}_S, \delta, s_0 \right)$$

| | | | |
|----------|--|--|--|
| δ | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

/ 5

- (b) Sei $S_T(w)$ die Anzahl an Schritten, die Ihre Turingmaschine T auf einer Eingabe w läuft.
- Geben Sie $S_T(w)$ an.
 - Geben Sie außerdem den **Durchschnittswert** $s_T(n)$ aller $S_T(w)$ für Wörter der Länge $n = |w|$ an.

Hinweis: Für einfache T gilt $s_T(|w|) = S_T(w)$ für alle w , aber nicht für die optimierte Variante aus (c).

| |
|-----|
| / 3 |
|-----|

- (c) Falls für Ihre Turingmaschine T der Durchschnittswert $s_T(n)$ **asymptotisch** mehr als $1,5n$ beträgt, falls also gilt:

$$\lim_{n \rightarrow \infty} \frac{s_T(n)}{1,5n} > 1$$

wie könnte man T optimieren, sodass asymptotisch nur $1,5n$ Schritte nötig sind? Erklären Sie kurz.

| |
|-----|
| / 3 |
|-----|

Hinweis: Falls Ihre Turingmaschine T bereits optimiert ist, erhalten Sie diese Punkte automatisch.

Aufgabe 4**8 Punkte**

2017-N-04

Grammatiken / 8

Gegeben sei die kontextfreie Grammatik:

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

$$P = \{S \rightarrow a \mid Bb \mid aS \mid aA,$$

$$A \rightarrow \lambda \mid Aa,$$

$$B \rightarrow a \mid S S b \mid B B b B\}$$

(a) Überführen Sie G in eine äquivalente Grammatik G_{CNF} in Chomsky-Normalform. / 6(b) Leiten Sie entweder bzgl. der Originalgrammatik G **oder** bzgl. G_{CNF} das Wort

$$aaabab \in L(G) = L(G_{CNF})$$

ab. Geben Sie eine Ableitungsfolge **oder** einen Ableitungsbaum an. Nutzen Sie die vorausgegangene Rückseite, falls der Platz nicht reicht. / 4

Aufgabe 5 **8 Punkte**

2017-N-05

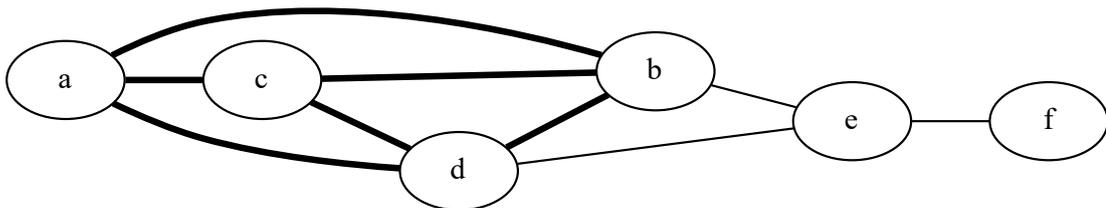
Komplexität

/ 8

Dem aus der Vorlesung bekannten *CLIQUE*-Problem liegt das Konzept vollständig verbundener Teilgraphen, sogenannter Cliques, in einem ungerichteten Graphen $G = (V, E)$ zugrunde.

(a) Sei $V = \{a, \dots, f\}$ und auf dieser Basis der folgende Graph G gegeben:

/ 3



Geben Sie in der Tabelle – **formal korrekt!** – für $k = 5, 3, 2, 1, 0$ jeweils **die Menge aller Cliques der Größe k** aus G an.

Hinweise:

- Orientieren Sie sich am Beispiel für $k = 4$ (fette Linien im Graphen).
- Bei mehr als 5 Cliques pro k dürfen Sie mit „...“ abkürzen.
- Bedenken Sie den Unterschied zwischen $k = 5$ und $k = 0$.

| k | Menge aller Cliques der Größe k |
|-----|-----------------------------------|
| 5 | |
| 4 | $\{\{a, b, c, d\}\}$ |
| 3 | |
| 2 | |
| 1 | |
| 0 | |

(b) Wie sind **Entscheidungs-**, **Optimierungs-** und **Konstruktionsvariante** von *CLIQUE* definiert? Formalisieren Sie oder beschreiben Sie umgangssprachlich präzise.

/ 3

- (c) Begründen Sie, dass, falls die **Entscheidungsvariante** von CLIQUE in det. Polynomialzeit lösbar wäre, die **Optimierungsvariante** ebenfalls in det. Polynomialzeit lösbar wäre.

/ 3

Aufgabe 6 **6 Punkte**

2017-N-06

Schaltnetze und Schaltwerke

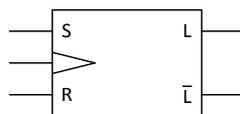
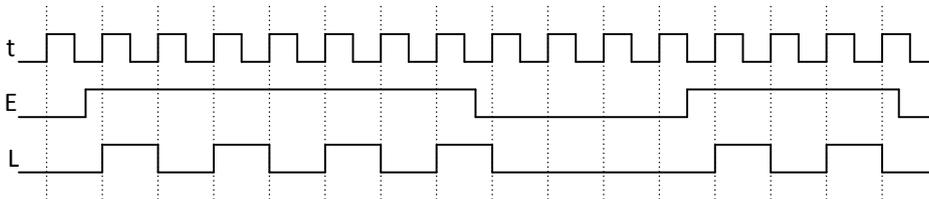
/ 6

Ein Detektor überwache ein Eingangssignal E und steuere mit einem Ausgangssignal L ein LED-Warmlämpchen an, das mit halber Taktfrequenz blinken soll, wenn das überwachte Signal auf 1 steht. Skizzieren Sie einen solchen Detektor als Schaltwerk aus einem synchronen RS-Flipflop, dessen positiver Ausgang direkt mit der LED verbunden sei. Bezeichnen bzw. verschalten Sie die Ein- und Ausgänge entsprechend.

Hinweise:

- Aus dem untenstehenden Signalverlauf ist ersichtlich, wie sich das Steuersignal L der LED bezüglich Takt t und Eingangssignal E verhalten soll.
- Synchrone Flipflops übernehmen einen neuen Wert exakt zum Zeitpunkt der steigenden Taktflanke und speichern ihn unverändert für den Rest des Taktes.
- Die Lösung ist nicht kompliziert, aber darauf zu kommen erfordert ein bisschen Knobelei oder die aus der Vorlesung bekannte strukturierte Vorgehensweise.

Signalverlauf:



| | |
|------------------|------------------|
| Aufgabe 7 | 6 Punkte |
| 2017-N-07 | Kodierung |
| | / 6 |

(a) Gegeben sei der folgende Text aus 35 Zeichen:

WENN-FLIEGEN-HINTER-FLIEGEN-FLIEGEN

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.

/ 7

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| N | G | F | I | W | T | R | H | E | - | L |
| | | | | | | | | | | |

| Zeichen | Kodierung |
|---------|-----------|
| N | |
| G | |
| F | |
| I | |
| W | |
| T | |
| R | |
| H | |
| E | |
| - | |
| L | |

- (b) Gegeben seien zwei gleich lange Texte A und B, die mit der in Teilaufgabe (a) erstellten Huffman-Kodierung kodiert werden sollen. Welcher der beiden Texte ist nach der Kodierung kürzer? Begründen Sie Ihre Antwort kurz.

Hinweis: Sie können diese Frage auch beantworten, ohne Teilaufgabe (a) gelöst zu haben.

| |
|-----|
| / 2 |
|-----|

| | | | | | | | | | | | | | | | | | | | |
|--------|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Text A | | G | E | H | I | R | N | - | L | E | R | N | T | - | G | E | R | N | E |
| Text B | | G | E | H | I | R | N | - | L | E | H | R | T | - | G | E | R | N | E |

Aufgabe 8**10 Punkte****2017-N-8****Rechnerarchitektur**

/ 10

- (a) Welche Bestandteile des Von-Neumann-Rechners sind in heutigen Rechnern noch vorhanden?

/ 2

- (b) Nennen Sie die Aufgaben des Steuerwerks und beschreiben Sie diese kurz.

/ 3

- (c) Datensammelwege, die sogenannten „Busse“, werden zur Übertragung von Daten, Befehlen oder Informationen benötigt. In der Vorlesung haben wir zentrale und dezentrale Verfahren zur Bussteuerung kennengelernt. Nennen Sie aus beiden Kategorien ein Beispiel und erklären Sie **ganz kurz**.

/ 2

- (d) Wenn es schon die Turingmaschine als abstraktes Modell für Rechner gibt, wofür wird dann die Architektur des Von-Neumann-Rechners noch benötigt?

/ 2

Aufgabe 9**8 Punkte**

2017-N-09

Programmiersprachen / 8

Gegeben seien jeweils als beispielhafte Vertreter für **imperative**, **objektorientierte**, **funktionale** und **prädikative** Programmiersprachen die Sprachen:

- C
- Java
- Lisp
- Prolog

Gegeben seien außerdem folgende Szenarios, für die Software geschrieben werden soll:

- (a) Hardwarenahe Programmierung von Mikrorobotern, begrenzte Ressourcen.
- (b) Analyse medizinischer Daten zur computergestützten Diagnose, Korrektheit wichtig.
- (c) Grammatikanalyse deutscher Texte, große Anzahl infrage kommender Regeln.
- (d) Großes Projekt zur Analyse von Verkehrsdaten, viele Programmierer und Teilprojekte.

Geben Sie für jedes Szenario eine der obigen Programmiersprachen an, die Sie zur Implementierung empfehlen würden, und begründen Sie kurz.

Hinweise: Begründen Sie anhand der in der Vorlesung genannten Vor- und Nachteile. Im Prinzip eignet sich jede Sprache für jedes Szenario, daher erhalten Sie **Punkte nur für die Begründungen**. Java enthält inzwischen auch Elemente funktionaler Programmierung, für diese Aufgabe seien diese aber nicht verfügbar.

(a) / 2

(b) / 2

(c) / 2

(d) / 2

Aufgabe 10**8 Punkte**

2017-N-10

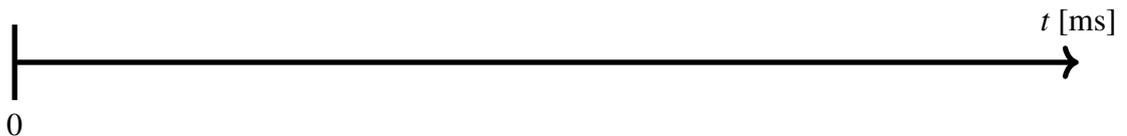
Betriebssysteme

 / 8

- (a) Betrachten Sie die Prozesse P1 bis P4, die in die Warteschlange eines Prozessors zur Bearbeitung eingereiht werden.

| Prozesse | CPU-Zeit in ms |
|----------|----------------|
| P1 | 25 |
| P2 | 35 |
| P3 | 5 |
| P4 | 15 |

Teilen Sie den Prozessen Rechenzeit gemäß dem Round Robin Verfahren zu. Die Zeitscheibe sei dabei in feste Zeitspannen der Länge $Z = 10$ ms unterteilt. Veranschaulichen Sie Ihr Ergebnis auf dem gegebenen Zeitstrahl.

 / 3

- (b) Nennen Sie für die folgenden Betriebsarten jeweils eine typische Anwendung.

 / 3

- Stapelbetrieb
- Echtzeitbetrieb
- Client/Server-Betrieb

Aufgabe 11 **7 Punkte**

2017-N-11

Dateiorganisation

| |
|-----|
| / 7 |
|-----|

- (a) Verwenden Sie die Divisionsmethode mit dem Divisor 7 um die folgenden Zahlen in die untenstehende Hashtabelle einzutragen. Behandeln Sie auftretende Kollisionen durch lineares Austesten.

11, 10, 19, 3, 13, 7

| | | | | | | | |
|-----------|---|---|---|---|---|---|---|
| RSN | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Schlüssel | | | | | | | |

| |
|-----|
| / 3 |
|-----|

- (b) Warum verwendet man bei der Divisionsmethode vornehmlich Primzahlen als Divisor?

| |
|-----|
| / 1 |
|-----|

- (c) Betrachten Sie folgende index-sequentielle Datei D mit zweistufigem Index: I_0 über D und I_1 über I_0 . Tragen Sie den Inhalt von D in den zugehörigen Suchbaum auf der nächsten Seite ein.

| |
|-----|
| / 2 |
|-----|

| | | |
|-----|-----------|-----|
| 000 | Kunigunde | 100 |
| 050 | Wolfhilde | 200 |
| 100 | Berta | 300 |
| 150 | Gertrud | |
| 200 | Ulf | 500 |
| 250 | | |
| 300 | Adelheid | |
| 350 | Bertel | |
| 400 | | |
| 450 | | |
| 500 | Siegfried | |
| 550 | | |
| 600 | | |
| 650 | | |

| | | |
|-----|-----------|------------|
| 000 | Kunigunde | 100 |
| 050 | Wolfhilde | 200 |
| 100 | Berta | 300 |
| 150 | Gertrud | |
| 200 | Ulf | 500 |
| 250 | | |
| 300 | Adelheid | |
| 350 | Bertel | 100 |
| 400 | | |
| 450 | | |
| 500 | Siegfried | |
| 550 | | |
| 600 | | |
| 650 | | |

