

01.08.2017

Lösung zur 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
(9)	(7)	(11)	(10)	(9)	(6)	(9)	(9)	(8)	(6)	(6)						(90)

Aufgabenübersicht

1) Minimierung endlicher Automaten (9 Punkte)	2
2) Reguläre Ausdrücke (7 Punkte)	5
3) Turingmaschinen (11 Punkte)	7
4) Grammatiken (10 Punkte)	10
5) Komplexität (9 Punkte)	12
6) Schaltnetze und Schaltwerke (6 Punkte)	14
7) Kodierung (9 Punkte)	15
8) Rechnerarchitektur (9 Punkte)	17
9) Programmiersprachen (8 Punkte)	19
10) Betriebssysteme (6 Punkte)	21
11) Dateiorganisation (6 Punkte)	23

Aufgabe 1 **9 Punkte**

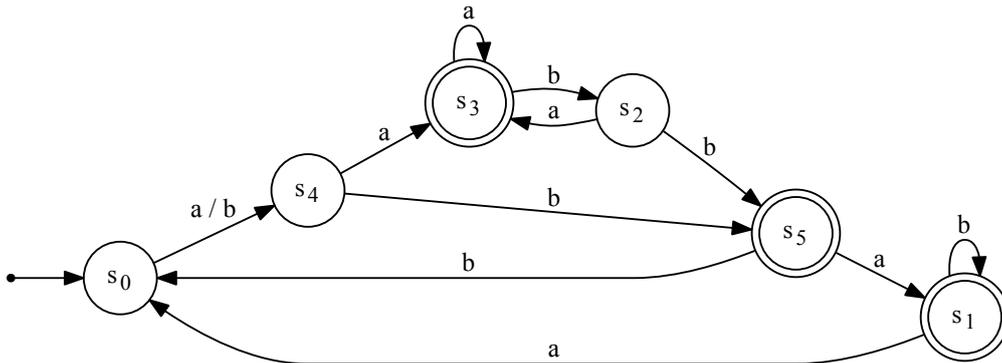
2017-N-01

Minimierung endlicher Automaten

/ 9

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

δ :



(a) Minimieren Sie A und geben Sie den minimierten Automaten A' vollständig an. Geben Sie insbesondere die Minimierungstabelle und ein Zustandsüberführungsdiagramm δ' an.

/ 5

Lösung:

- Übergangstabelle für A :

	a	b
s_0	s_4	s_4
s_1	s_0	s_1
s_2	s_3	s_5
s_3	s_3	s_2
s_4	s_3	s_5
s_5	s_1	s_0

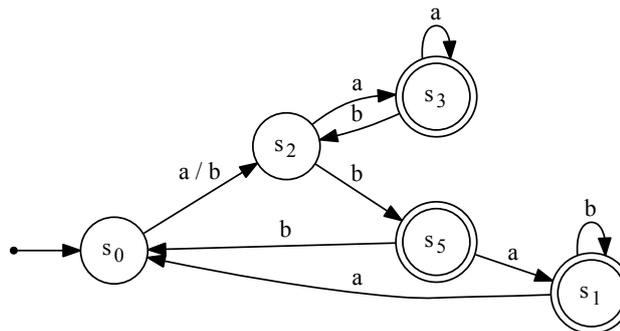
- Dreieckstabelle:

s_1	\times_0				
s_2	\times_1	\times_0			
s_3	\times_0	\times_1	\times_0		
s_4	\times_1	\times_0	–	\times_0	
s_5	\times_0	\times_1	\times_0	\times_2	\times_0
	s_0	s_1	s_2	s_3	s_4

- Übergangstabelle für A' :

	a	b
s_0	s_2	s_2
s_1	s_0	s_1
s_2	s_3	s_5
s_3	s_3	s_2
s_5	s_1	s_0

- Der minimierte Automat A' lautet wie folgt:
 $A' = (\{a, b\}, \{s_0, s_1, s_2, s_3, s_5\}, \delta', s_0, \{s_1, s_3, s_5\})$



- (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

Lösung:

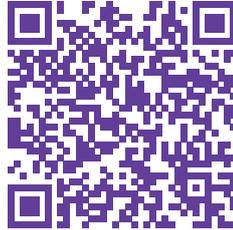
0-Äquivalenz	$\{s_0, s_2, s_4\}, \{s_1, s_3, s_5\}$
1-Äquivalenz	$\{s_3, s_5\}, \{s_2, s_4\}, \{s_0\}, \{s_1\}$
2-Äquivalenz	$\{s_2, s_4\}, \{s_0\}, \{s_1\}, \{s_3\}, \{s_5\}$
Äquivalenz	$\{s_2, s_4\}, \{s_0\}, \{s_1\}, \{s_3\}, \{s_5\}$

Skript für den Ausgangsautomaten:

```
fsm:
(s0, a) | (s0, b) => s4;
(s1, a) | (s5, b) => s0;
(s1, b) | (s5, a) => s1;
(s2, a) | (s3, a) | (s4, a) => s3;
(s2, b) | (s4, b) => s5;
(s3, b) => s2;
--declarations--
s0=s0;
```

```
F=s3,s5,s1;  
--declarations-end--
```

SKRIPT ID-24262



Skript für den minimierten Automaten:

```
fsm:  
(s0, a) | (s0, b) | (s3, b) => s2;  
(s1, a) | (s5, b) => s0;  
(s1, b) | (s5, a) => s1;  
(s2, a) | (s3, a) => s3;  
(s2, b) => s5;  
--declarations--  
s0=s0;  
F=s3,s5,s1;  
--declarations-end--
```

SKRIPT ID-24295



Aufgabe 2

7 Punkte

2017-N-02

Reguläre Ausdrücke

/ 7

(a) Gegeben sei die rechtslineare Grammatik

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

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

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

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

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

SKRIPT ID-24472



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

/ 4

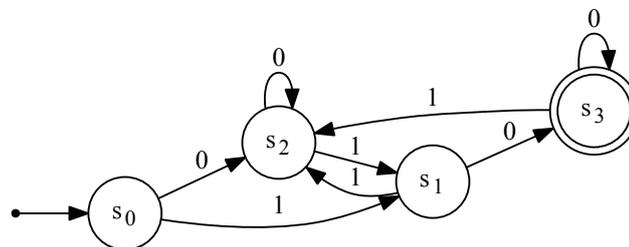
Lösung:

$$\alpha = (aa^*bb^*cc^*d + aa^*cc^*d)^*(\emptyset^* + aa^*bb^*) = (aa^*b^*cc^*d)^*(\emptyset^* + aa^*bb^*)$$

(b) Gegeben sei der endliche Automat

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

δ :



SKRIPT ID-24470



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

Lösung:

$$G = (\{s_0, s_1, s_2, s_3\}, \{0, 1\}, P, s_0)$$

$$P = \{s_0 \rightarrow 0s_2 \mid 1s_1,$$

$$s_1 \rightarrow 0s_3 \mid 1s_2,$$

$$s_2 \rightarrow 0s_2 \mid 1s_1,$$

$$s_3 \rightarrow \lambda \mid 0s_3 \mid 1s_2\}$$

SKRIPT ID-24475



RA (nicht verlangt): $\alpha = (1 + 00^*1)(10^*1 + 00^*10^*1)^*00^*$

Aufgabe 3 **11 Punkte**

2017-N-03

Turingmaschinen

/ 11

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 $f_T = f$.

/ 5

Lösung:

	a	b	\star
s_0	(s_0, a, L)	(s_0, b, L)	(s_1, a, R)
s_1	(s_1, a, R)	(s_1, b, R)	(s_2, \star, L)
s_2		(s_3, b, L)	
s_3	(s_3, a, L)	(s_3, b, L)	(s_4, \star, R)
s_4	(s_4, b, N)		

Auf Verdacht a links neben w schreiben.
 Nach rechts laufen.
 Falls a am Ende, sind wir schon fertig.
 Sonst müssen wir nochmal zurück...
 ... und b schreiben.

Tape	Transition
<i>âaaaabab</i>	$(s_0, a) \rightarrow (s_0, a, L)$
<i>âaaaaabab</i>	$(s_0, \star) \rightarrow (s_1, a, R)$
<i>ââaaaabab</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>âââaaaabab</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>ââââaaaabab</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>âââââaaaabab</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>ââââââaaaabab</i>	$(s_1, b) \rightarrow (s_1, b, R)$
<i>âââââââaaaabab</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>ââââââââaaaabab</i>	$(s_1, b) \rightarrow (s_1, b, R)$
<i>âââââââââaaaabab</i>	$(s_1, \star) \rightarrow (s_2, \star, L)$
<i>ââââââââââaaaabab</i>	$(s_2, b) \rightarrow (s_3, b, L)$
<i>âââââââââââaaaabab</i>	$(s_3, a) \rightarrow (s_3, a, L)$
<i>ââââââââââââaaaabab</i>	$(s_3, b) \rightarrow (s_3, b, L)$
<i>âââââââââââââaaaabab</i>	$(s_3, b) \rightarrow (s_3, b, L)$
<i>ââââââââââââââaaaabab</i>	$(s_3, a) \rightarrow (s_3, a, L)$
<i>âââââââââââââââaaaabab</i>	$(s_3, a) \rightarrow (s_3, a, L)$
<i>ââââââââââââââââaaaabab</i>	$(s_3, a) \rightarrow (s_3, a, L)$
<i>âââââââââââââââââaaaabab</i>	$(s_3, a) \rightarrow (s_3, a, L)$
<i>ââââââââââââââââââaaaabab</i>	$(s_3, \star) \rightarrow (s_4, \star, R)$
<i>âââââââââââââââââââaaaabab</i>	$(s_4, a) \rightarrow (s_4, b, N)$
<i>ââââââââââââââââââââaaaabab</i>	

Tape	Transition
<i>ââââââââ</i>	$(s_0, a) \rightarrow (s_0, a, L)$
<i>âââââââââ</i>	$(s_0, \star) \rightarrow (s_1, a, R)$
<i>ââââââââââ</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>âââââââââââ</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>ââââââââââââ</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>âââââââââââââ</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>ââââââââââââââ</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>âââââââââââââââ</i>	$(s_1, b) \rightarrow (s_1, b, R)$
<i>ââââââââââââââââ</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>âââââââââââââââââ</i>	$(s_1, a) \rightarrow (s_1, a, R)$
<i>ââââââââââââââââââ</i>	$(s_1, \star) \rightarrow (s_2, \star, L)$

SKRIPT ID-24101



(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

Lösung:

$$S(vx) = \begin{cases} |vx| + 3 & \text{falls } x = a, \\ 2|vx| + 6 & \text{sonst} \end{cases}$$

Durchschnittlich sind das also für jede Länge n (da es gleich viele Wörter mit a am Ende gibt wie mit b am Ende):

$$s_T(n) = \frac{n + 3 + 2n + 6}{2} = 1,5n + 4,5$$

- (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.

Lösung:

Siehe (a).

Aufgabe 4

10 Punkte

2017-N-04

Grammatiken

/ 10

Gegeben sei die kontextfreie Grammatik:

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

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

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

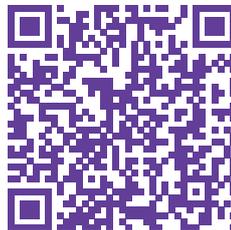
$$B \rightarrow a \mid SSb \mid BBbB\}$$

(a) Überführen Sie G in eine äquivalente Grammatik G_{CNF} in Chomsky-Normalform.

Lösung:

/ 6

SKRIPT ID-24468



*** Lambda-free grammar ***

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

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

$$A \rightarrow a \mid Aa,$$

$$B \rightarrow a \mid SSb \mid BBbB\}$$

*** Grammar without chains ***

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

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

$$A \rightarrow a \mid Aa,$$

$$B \rightarrow a \mid SSb \mid BBbB\}$$

*** Grammar with all terminals isolated ***

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

$$P = \{S \rightarrow a \mid BC_b \mid C_aS \mid C_aA,$$

$$A \rightarrow a \mid AC_a,$$

$$B \rightarrow a \mid SSC_b \mid BBC_bB,$$

$$C_a \rightarrow a,$$

$$C_b \rightarrow b\}$$

*** CNF grammar ***

$$G = (\{A, B, C_a, C_b, D_0, D_1, D_2, S\}, \{a, b\}, P, S)$$

$$P = \{S \rightarrow a \mid BC_b \mid C_aS \mid C_aA,$$

$$A \rightarrow a \mid AC_a,$$

$$B \rightarrow a \mid D_1B \mid D_2C_b,$$

$$C_a \rightarrow a,$$

$$C_b \rightarrow b,$$

$$D_0 \rightarrow BB,$$

$$D_1 \rightarrow D_0C_b,$$

$$D_2 \rightarrow SS\}$$

(b) Leiten Sie entweder bzgl. der Originalgrammatik G **oder** bzgl. G_{CNF} das Wort

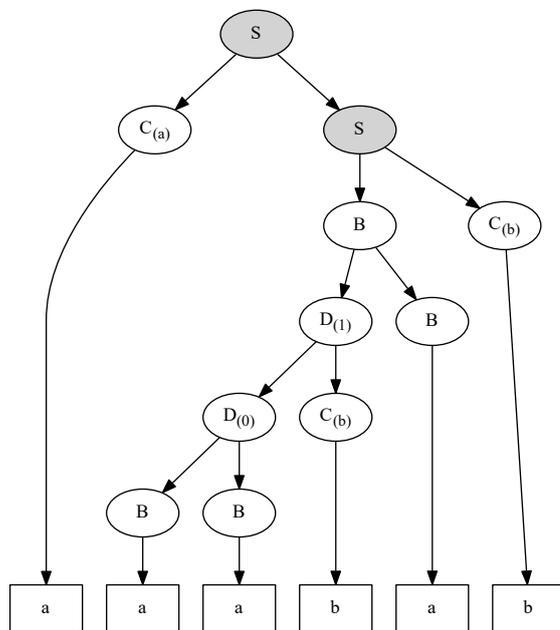
$$aaabab \in L(G) = L(G_{CNF})$$

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

/ 4

Lösung:

Ableitungsbaum nach G_{CNF} (alternativ nach $G : S \Rightarrow aS \Rightarrow aBb \Rightarrow aBBbBb \Rightarrow^* aaabab$):



SKRIPT ID-24467



Aufgabe 5 **9 Punkte**

2017-N-05

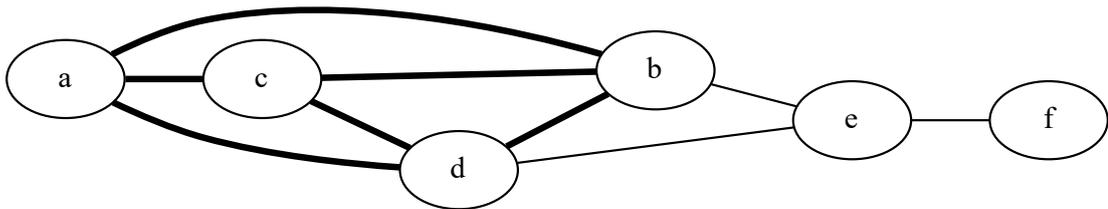
Komplexität

/ 9

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	\emptyset
4	$\{\{a, b, c, d\}\}$
3	$\{\{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{b, d, e\}\}$
2	$\{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \dots\}$
1	$\{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}\}$
0	$\{\emptyset\}$

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

/ 3

Lösung:

Gegeben sei ein Graph $G = (V, E)$.

- Entscheidungsvariante: Existiert eine Clique der Größe k in G , wobei $k \geq 0$ fix gegeben ist?
 - Optimierungsvariante: Wie lautet das größte k für das eine Clique der Größe k existiert?
 - Konstruktionsvariante: Welche Knoten gehören zur größten Clique in G ?
- (c) Begründen Sie, dass, falls die **Entscheidungsvariante** von **CLIQUE** in det. Polynomialzeit lösbar wäre, die **Optimierungsvariante** ebenfalls in det. Polynomialzeit lösbar wäre.

/ 3

Lösung:

Sei n die Anzahl der Knoten in V und $p(n)$ ein Polynom, das den Aufwand zur Lösung des Entscheidungsproblems nach oben abschätzt (nach heutigen Erkenntnissen kann p nicht existieren, das ist also nur ein Gedankenexperiment). Auf Basis der Entscheidungsvariante von **CLIQUE** kann iterativ überprüft werden, ob eine Clique der Größe $k \in \{1, \dots, n\}$ existiert, bspw. durch binäre Suche. Da jeder Graph eine Clique der Größe 0 enthält und kein Graph eine Clique der Größe $m > n$, muss das Verfahren nach $O(\log(n))$ Schritten terminieren. Die Entscheidungsvariante von **CLIQUE** wird damit nur logarithmisch oft ausgeführt und der polynomielle Aufwand $p(n)$ also nur (höchstens) auf $p(n) \log(n) \in P$ angehoben.

Aufgabe 6 **6 Punkte**

2017-N-06

Schaltnetze und Schaltwerke

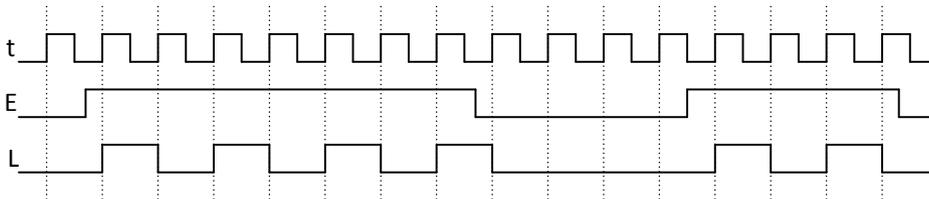
/ 6

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.

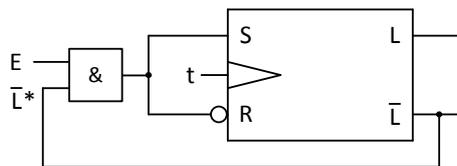
Hinweise:

- 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:



Lösung:



Aufgabe 7 **9 Punkte**

2017-N-07

Kodierung

/ 9

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

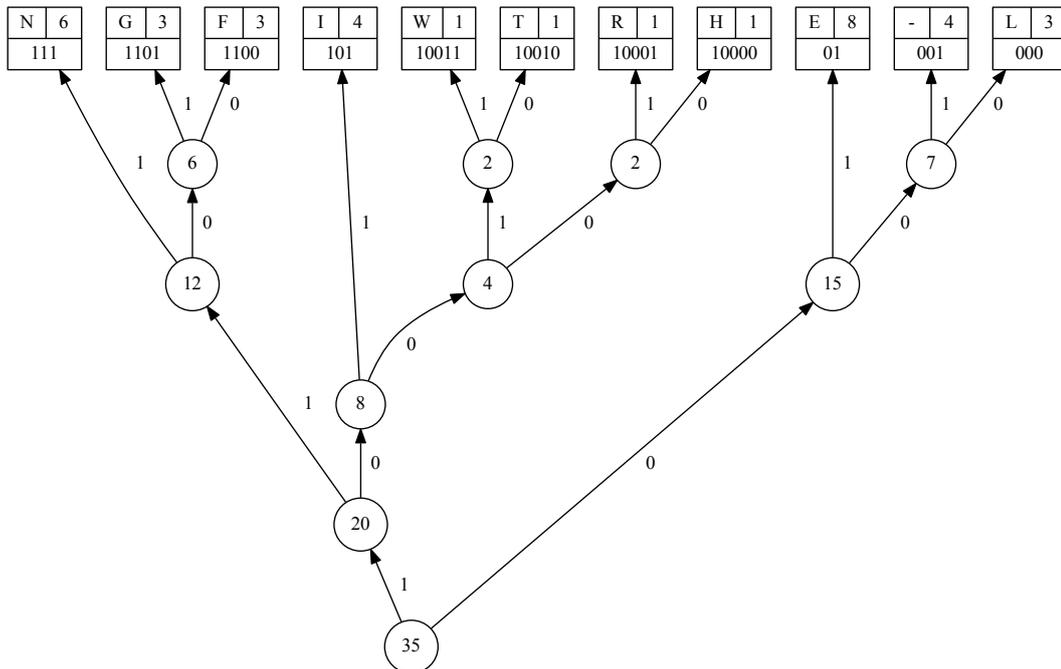
WENN-FLIEGEN-HINTER-FLIEGEN-FLIEGEN

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
6	3	3	4	1	1	1	1	8	4	3

Lösung (Beispiel):



- (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

Lösung:

Die Zeichen in Text A und B sind identisch bis auf eines: Text A enthält ein 'N' statt eines 'H'. Da das 'N' in dem der Kodierung zu Grunde liegenden Text aus Aufgabenteil (a) häufiger vorkommt als das 'H' (6 Mal statt 1 Mal), ist das 'N' kürzer kodiert als das 'H'. Damit ist auch Text A kürzer kodiert als Text B.

Aufgabe 8

9 Punkte

2017-N-8

Rechnerarchitektur

/ 9

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

/ 2

Lösung:

Arbeitsspeicher, Rechenwerk und Steuereinheit (CPU), Ein-/Ausgabeeinheit, interne Datenwege

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

/ 3

Lösung:

- **Holen von Befehlen (FETCH):** Befehl gemäß PC adressieren und aus dem Speicher laden
- **Entschlüsseln von Befehlen (DECODE):** Erkennen der Befehlsart, Zerlegung in seine Bestandteile (Operations-, Operandenteil)
- **Initiieren der Befehlsausführung (EXECUTE):** an der Befehlsausführung beteiligte Funktionseinheiten mit den notwendigen Steuersignalen versorgen: Operand(en) adressieren und (aus dem Speicher) laden, Operation ausführen, Ergebnis speichern, Befehlszähler PC verändern

- (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

Lösung:

- zentrale Steuerung: Übertragungswünsche von jedem Teilnehmer über eigene Meldeleitung, Bussteuereinheit fragt Teilnehmerwünsche ab (polling), zyklische Buszuteilung unabhängig von Übertragungswünschen
 - dezentrale Steuerung: Jeder Sender muss für sich entscheiden können, ob der Bus verfügbar ist: Daisy Chain, Parallele Abfrage, zyklische Buszuteilung, CSMA-Verfahren (carrier sense multiple access)
- (d) Wenn es schon die Turingmaschine als abstraktes Modell für Rechner gibt, wofür wird dann die Architektur des Von-Neumann-Rechners noch benötigt?

/ 2

Lösung:

Die Von-Neumann-Architektur ist ein Schaltungskonzept zur **Realisierung** universeller Rechner. Sie realisiert alle Komponenten einer Turingmaschine. Dabei ermöglicht ihre systematische Aufteilung in die entsprechenden Funktionsgruppen jedoch die Nutzung spezialisierter Schaltwerke und damit eine effizientere Strukturierung der Operationen. Die Realisierung einer Turingmaschine in Hardware und ihre Programmierung wären viel komplizierter und kaum durchführbar.

Im Prinzip bleibt es aber dabei, dass alles, was mit einer Turingmaschine berechenbar ist, auch auf einer Maschine mit Von-Neumann-Architektur berechenbar ist und umgekehrt.

(Es gibt weitere korrekte Antworten.)

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

Lösung:

Beispielsweise C, denn mit C lassen sich Abläufe auf der Hardware sehr präzise steuern (sogar Assembler-Code kann eingebettet werden), was eine hocheffiziente Programmierung ermöglicht. Mit fortschreitender Technik kann sich allerdings auch Java o. ä. für die Programmierung von Mikrorobotern eignen.

(b) / 2

Lösung:

Beispielsweise Lisp, denn funktionale Programmierung ermöglicht eine Abstraktion von den Kontrollstrukturen und eine stärkere Fokussierung auf die Semantik. PROLOG könnte hier aus demselben Grund auch eine gute Wahl sein.

(c) / 2

Lösung:

PROLOG kann und wird für solche Aufgaben eingesetzt, weil es konkurrenzlos einfach und intuitiv sowohl Syntax als auch Semantik natürlicher Sprachen abbilden kann. Allerdings ist das heute eine der wenigen echten Anwendungen für PROLOG, v. a. wegen der Schwierigkeit, effiziente und übersichtliche Programme zu implementieren. (Es geht, aber es ist nicht einfach.)

(d)

/ 2

Lösung:

Für große Projekte mit vielen Funktionen und vor allem auch vielen Programmierern sind objektorientierte Sprachen wie Java gut geeignet. Neben den Vorteilen, die sich durch Vererbung etc. ergeben, ist vor allem die Möglichkeit privater Variablen wichtig, damit man steuern kann, welche Teile des eigenen Codes von anderen benutzt werden können und welche nicht.

Aufgabe 10 **6 Punkte**

2017-N-10

Betriebssysteme

/ 6

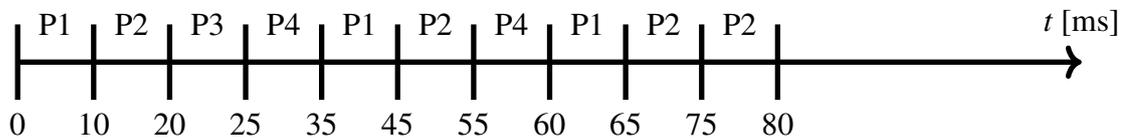
- (a) Betrachten Sie die Prozesse P1 bis P4, die in die Warteschlange eines Prozessors zur Bearbeitung eingereicht werden.

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

Lösung:



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

/ 3

- Stapelbetrieb

Lösung:

Statistische Auswertungen, technisch/wissenschaftliche Berechnungen, etc.

- Echtzeitbetrieb

Lösung:

Kfz-Elektronik, Telekommunikation, etc.

- Client/Server-Betrieb

Lösung:

Ilias-Plattform, Streaming von Medieninhalten, etc.

Aufgabe 11 **6 Punkte**

2017-N-11

Dateiorganisation

/ 6

- (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

/ 3

Lösung:

Schlüssel	11	10	19	3	13	7
Hash	4	3	5	3	6	0

RSN	0	1	2	3	4	5	6
Schlüssel	13	7		10	11	19	3

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

/ 1

Lösung:

Die Verwendung von Primzahlen führt zu weniger Kollisionen in der Hashtabelle.

- (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

I_0	000	Kunigunde	100
	050	Wolfhilde	200
	100	Berta	300
I_1	150	Gertrud	
	200	Ulf	500
	250		
	300	Adelheid	
D	350	Bertel	
	400		
	450		
	500	Siegfried	
	550		
	600		
	650		

Lösung:

