

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

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_

Matr.-Nr.: \_\_\_\_\_ Semester: \_\_\_\_\_ (WS 2011/12)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-14).

\_\_\_\_\_  
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
(11)	(9)	(10)	(7)	(9)	(9)	(7)	(9)	(8)	(7)	(4)						(90)

# Aufgabenübersicht

1) <b>Minimierung und reguläre Ausdrücke</b> (11 Punkte) . . . . .	2
2) <b>Kellerautomat</b> (9 Punkte) . . . . .	4
3) <b>Grammatiken und Chomsky-Hierarchie</b> (10 Punkte) . . . . .	5
4) <b>Cocke-Younger-Kasami-Algorithmus</b> (7 Punkte) . . . . .	6
5) <b>Komplexitäts- und Berechenbarkeitstheorie</b> (9 Punkte) . . . . .	7
6) <b>CMOS</b> (9 Punkte) . . . . .	8
7) <b>Zahldarstellung</b> (7 Punkte) . . . . .	10
8) <b>Huffman-Kodierung</b> (9 Punkte) . . . . .	11
9) <b>Assembler und Adressierung</b> (8 Punkte) . . . . .	12
10) <b>Speicherorganisation</b> (7 Punkte) . . . . .	13
11) <b>Betriebssysteme</b> (4 Punkte) . . . . .	14

**Aufgabe 1****11 Punkte****2012-H-01****Minimierung und reguläre Ausdrücke**

/ 11

Bei der Minimierung eines endlichen Automaten habe sich folgende Minimierungstabelle ergeben.

$s_1$	X1				
$s_2$	X0	X0			
$s_3$	X2	X1	X0		
$s_4$	X2	X1	X0	X2	
$s_5$	X1	X1	X0	X1	X1
	$s_0$	$s_1$	$s_2$	$s_3$	$s_4$

(a) Geben Sie für jedes  $k \in \{0, 1, 2\}$  jeweils die Aufteilung der Zustandsmenge auf die Mengen der Zustände des Automaten an, die zueinander  $k$ -äquivalent sind, sowie die Mengen der äquivalenten Zustände. Geben Sie auch einelementige Mengen an.

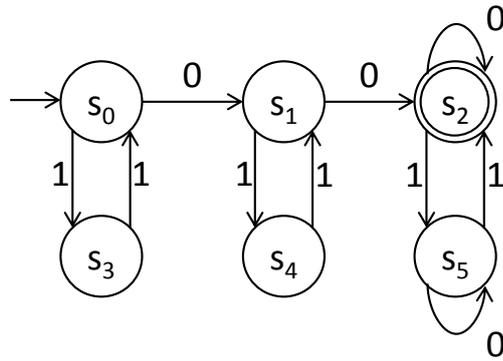
- Mengen 0-äquivalenter Zustände:
- Mengen 1-äquivalenter Zustände:
- Mengen 2-äquivalenter Zustände:
- Mengen äquivalenter Zustände:

(b) Der minimierte Automat  $A$  sei gegeben durch  $A = (\{0, 1\}, \{s_0, s_1, s_2, s_3, s_4, s_5\}, \delta, s_0, \{s_2\})$ .

Geben Sie eine alternative Menge der Endzustände  $\overline{F}$  an, die sich aus obiger Minimierungstabelle ergeben könnte.

$\overline{F} =$

(c) Gegeben sei der endliche Automat  $A' = (\{0, 1\}, \{s_0, s_1, s_2, s_3, s_4, s_5\}, \delta', s_0, \{s_2\})$  mit  $\delta'$ :



Beschreiben Sie die Sprache  $L(A')$  durch einen regulären Ausdruck  $\alpha$ .

**Aufgabe 2****9 Punkte****2012-H-02****Kellerautomat**

/ 9

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

- $\lambda \in L$ ,
- falls  $w, w' \in L$ , dann auch:  $ww'$ ,  $(w)$  und  $[w] \in L$ .

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

Es gilt beispielsweise:

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

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

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

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

**Aufgabe 3****10 Punkte****2012-H-03****Grammatiken und Chomsky-Hierarchie**

/ 10

Gegeben sei die Grammatik  $G$  mit

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

$$P = \{S \rightarrow YTb \mid a,$$

$$Y \rightarrow BY \mid aa,$$

$$Ba \rightarrow aaB,$$

$$BT \rightarrow Tb,$$

$$aTb \rightarrow ab\}$$

(a) Geben Sie vier verschiedene Wörter  $w_1, w_2, w_3, w_4 \in L(G)$  an.

$w_1$  :

$w_2$  :

$w_3$  :

$w_4$  :

(b) Geben Sie die Sprache  $L(G)$  (mathematisch oder umgangssprachlich präzise) an.

(c) Geben Sie das größte  $i$  an, sodass gilt: Die Grammatik  $G$  ist vom Chomsky-Typ  $i$ .

(d) Angenommen, jemand beauftragt Sie, die Grammatik  $G$  in Chomsky-Normalform zu bringen. Wie würden Sie vorgehen, um zu beweisen, dass das nicht möglich ist?

Begründen Sie kurz.

**Aufgabe 4**

**7 Punkte**

**2012-H-04**

**Cocke-Younger-Kasami-Algorithmus**

/ 7
-----

Gegeben sei die Grammatik  $G = (N, T, P, S)$  mit  $N = \{S, A, B, C\}$ ,  $T = \{a, b, c\}$  und

$$\begin{aligned}
 P = \{ & S \rightarrow AB \mid a, \\
 & A \rightarrow BC \mid a, \\
 & B \rightarrow CA \mid AS \mid b, \\
 & C \rightarrow SC \mid AB \mid c\}.
 \end{aligned}$$

Überprüfen Sie mithilfe des Algorithmus von Cocke, Younger und Kasami, ob

$$w = bacca \in L(G).$$

Geben Sie zusätzlich zum Ausfüllen der Tabelle explizit an, ob  $w \in L(G)$ .

	<i>b</i>	<i>a</i>	<i>c</i>	<i>c</i>	<i>a</i>
<i>m</i> = 1					
<i>m</i> = 2					
<i>m</i> = 3					
<i>m</i> = 4					
<i>m</i> = 5					

<b>Aufgabe 5</b>	<b>9 Punkte</b>
<b>2012-H-05</b>	<b>Komplexitäts- und Berechenbarkeitstheorie</b>
	/ 9

Gegeben seien die folgenden Informationen zu den Problemen  $A, B, C, D, E$  und  $F$ :

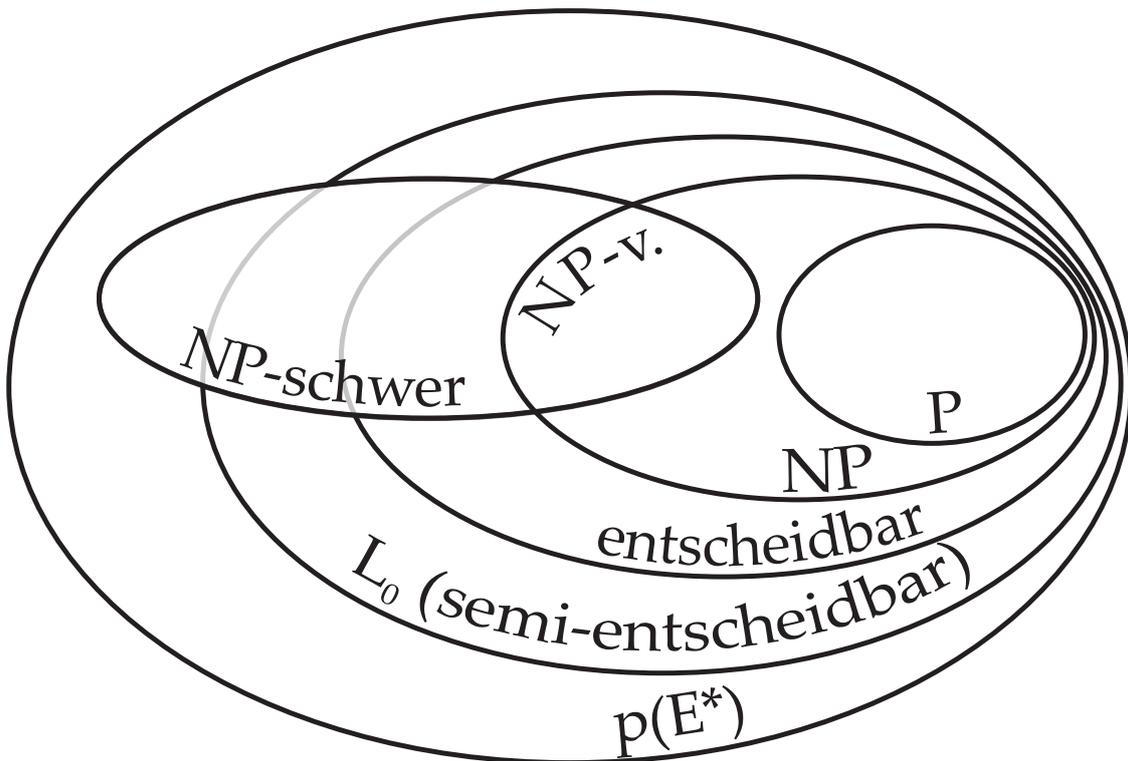
- $CLIQUE \leq_{pol} A \leq_{pol} B \leq_{pol} F$ ,
- $B \in NP$ ,
- $C \leq_{pol} PRIMES \leq_{pol} D$ ,
- $E$  ist äquivalent zum Halteproblem.

Darüber hinaus sei bekannt, dass

- $CLIQUE \in NP$ -vollständig,
- $PRIMES \in P$ ,

Zeichnen Sie  $A, B, C, D, E$  und  $F$  so genau wie möglich in das abgebildete Diagramm ein.

**Hinweis:** Sie erhalten 1,5 Punkte pro Problem, das an „genau der richtigen Stelle“ eingeordnet ist, also so tief wie möglich, aber nicht tiefer als aus den obigen Informationen ableitbar. Für  $NP$ -schwer müssen Sie nur „ $NP$ -vollständig“ und „nicht  $NP$ -vollständig“ unterscheiden.



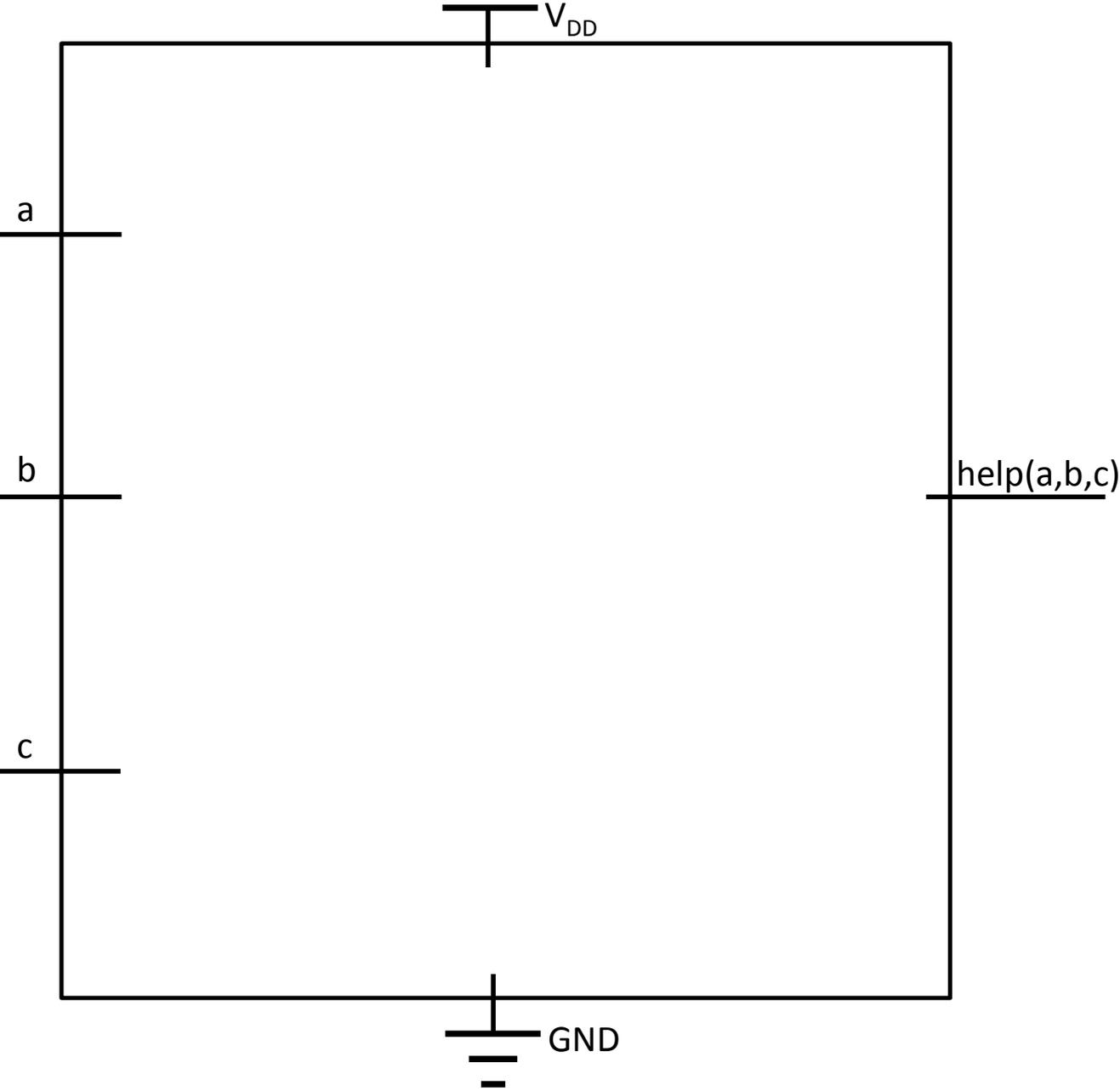
**Aufgabe 6****9 Punkte****2012-H-06****CMOS**

/ 9
-----

Gegeben sei folgende Schaltfunktion „help“:

$$\text{help}(a, b, c) = b \vee (a \wedge \neg c).$$

Geben Sie eine CMOS-Schaltung für diese Funktion an. Zeichnen Sie diese auf Transistorebene (verwenden Sie also keine elementaren CMOS-Bausteine als Blackboxen).



**Aufgabe 7****7 Punkte****2012-H-07****Zahlendarstellung**

	/7
--	----

Gegeben sei ein 4-stelliger Binärstring  $w \in \{0, 1\}^4$ .

Geben Sie jeweils die kleinste und die größte darstellbare Dezimalzahl an, wenn  $w$  als folgende Zahlendarstellung interpretiert wird:

(a) Vorzeichen-Betrag-Darstellung

(b) 1-Komplement-Darstellung

(c) 2-Komplement-Darstellung

(d) BCD-Darstellung

(e) Exzess-3-Darstellung

(f) Gleitpunktdarstellung mit 1-Bit-Charakteristik und 2-Bit-Mantisse

**Hinweis:** Gehen Sie davon aus, dass es – im Gegensatz zur IEEE-754-Darstellung – keine Sonderwerte oder denormalisierte Zahlen gibt.

**Aufgabe 8****9 Punkte****2012-H-08****Huffman-Kodierung**

/ 9

Folgende Zeichenkette sei repräsentativ für Daten, die noch kommen sollen:

**ABRAKADABRA-SIMSALABIM**

Erzeugen Sie anhand der durch die Zeichenkette gegebenen Häufigkeitsverteilung eine Huffman-Kodierung.

Tragen Sie dazu die Häufigkeiten der Zeichen in die erste Tabelle ein, erstellen Sie einen entsprechenden Baum mit Angabe der Häufigkeiten an den Knoten und tragen Sie die Kodierung der Zeichen in die zweite Tabelle ein.

Zeichen	A	B	R	S	I	M	K	D	-	L
Häufigkeit										

Zeichen	Code
A	
B	
R	
S	
I	
M	
K	
D	
-	
L	

<b>Aufgabe 9</b>	<b>8 Punkte</b>
<b>2012-H-09</b>	<b>Assembler und Adressierung</b>
	/ 8

Die Befehle einer Assembler-Sprache seien folgendermaßen aufgebaut:

OpCode Q1, (Q2,) Z

Dabei werde das Ergebnis der Operation *OpCode* basierend auf den durch Q1 (und bei Rechenoperationen Q2) bezeichneten Operanden an der durch Z bezeichneten Adresse abgelegt.

Gegeben sei folgender Ausschnitt aus einem Assemblerprogramm.

```

STORE 1001, 1002
STORE #1002, 1001
STORE *1001, 1003
MULTIPLY 1003, 1002, *1003
SUBTRACT 50, #2499, 1004
    
```

Dabei werden für die Adressierungsarten folgende Notationen verwendet:

- Unmittelbare Adressierung: Präfix '#'
- Direkte Adressierung: ohne Präfix
- Indirekte Adressierung: Präfix '\*'

Gegeben sei weiter ein 4-zeiliger Assoziativ-Cache, der nach dem Least-Recently-Used-Prinzip arbeite und zu Beginn die eingetragenen Werte enthalte:

Cache-Zeile	Tag-Feld (Hauptspeicheradresse)	Datum	Zugriffs- Zeitpunkte
0	1001	50	
1			
2			
3			

Bei jeder Assembler-Operation werden die beteiligten Operanden im Cache abgelegt.

Aktualisieren Sie die obige Tabelle mit den während des Programmlaufs anfallenden Werten. Schreiben Sie dabei die Zugriffszeitpunkte in die letzte Spalte.

**Aufgabe 10** **7 Punkte**

**2012-H-010**

**Speicherorganisation**

/ 7

- (a) Geben Sie für folgende Speicher die Energieabhängigkeit sowie die Zugriffsart an. Markieren Sie dies durch ein Kreuz in der entsprechenden Zelle der folgenden Tabelle.

	Energieabhängigkeit		Zugriffsarten		
	flüchtig	nicht flüchtig	wahlfrei	sequentiell	block-adressierbar
Register					
Optischer Speicher					
Hauptspeicher					
Cache					
Magnetplattenspeicher					
Solid-State-Disk					

- (b) Nennen Sie je einen Vor- und einen Nachteil für die Verwendung von Solid-State-Disks gegenüber herkömmlichen Magnetplattenspeichern.

