

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

Name: _____ Vorname: _____

Matr.-Nr.: _____ Semester: _____ (WS 2012/13)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-16).

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
(10)	(10)	(8)	(10)	(8)	(8)	(9)	(8)	(6)	(7)	(6)						(90)

Aufgabenübersicht

1) Umwandlung nichtdet. in det. endliche Automaten (10 Punkte)	2
2) Grammatiken (10 Punkte)	4
3) Cocke-Younger-Kasami-Algorithmus (8 Punkte)	6
4) Turingmaschine (10 Punkte)	7
5) Komplexitätstheorie (8 Punkte)	8
6) CMOS (8 Punkte)	10
7) Zahldarstellung (9 Punkte)	11
8) Huffman-Kodierung (8 Punkte)	13
9) Speicherorganisation und Cache (6 Punkte)	14
10) Maschinensprache (7 Punkte)	15
11) Betriebssystem (6 Punkte)	16

Aufgabe 1 **10 Punkte**

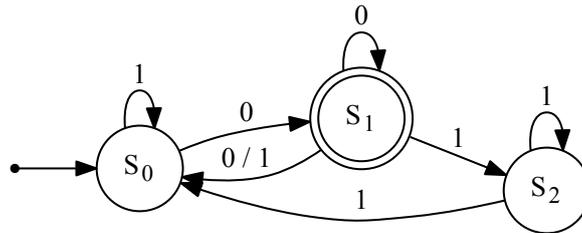
2013-H-01 **Umwandlung nichtdet. in det. endliche Automaten**

/ 10

Gegeben sei der nichtdeterministische endliche Automat

$$A = (\{0, 1\}, \{s_0, s_1, s_2\}, \delta, s_0, \{s_1\})$$

mit δ :



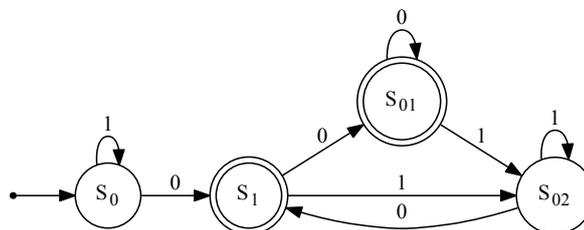
- (a) Erstellen Sie einen deterministischen endlichen Automaten $A' = (E, S', \delta', s'_0, F')$ mit $L(A') = L(A)$ mithilfe des aus der Vorlesung bekannten Algorithmus. Nutzen Sie dafür die vorgegebene Tabelle. Definieren Sie A' vollständig. Zeichnen Sie insbesondere ein Zustandsüberförungsdiagramm.

/ 5

δ'	0	1
$\{s_0\}$	$\{s_1\}$	$\{s_0\}$
$\{s_1\}$	$\{s_0, s_1\}$	$\{s_0, s_2\}$
$\{s_0, s_1\}$	$\{s_0, s_1\}$	$\{s_0, s_2\}$
$\{s_0, s_2\}$	$\{s_1\}$	$\{s_0, s_2\}$

Definition: $A' = (\{0, 1\}, \{s_0, s_1, s_{01}, s_{02}\}, \delta', s_0, \{s_1, s_{01}\})$

δ' :



- (b) Geben Sie einen regulären Ausdruck α an mit $L(\alpha) = L(A) = L(A')$.

/ 5

Lösung:

$$RA(\{0, 1\}) : \alpha = 1^*0(0^* + (0 + 1)1^*0 + 111^*0)^*$$

Aufgabe 2

10 Punkte

2013-H-02

Grammatiken

/ 10

Gegeben sei die monotone Grammatik

$$\begin{aligned}
 G &= (\{S, B, C\}, \{a, b, c\}, P, S), \\
 P &= \{S \rightarrow aSBBCCC \mid abbCCC \mid \lambda, \\
 &\quad CB \rightarrow BC, \\
 &\quad bB \rightarrow bb, \\
 &\quad bC \rightarrow bc, \\
 &\quad cC \rightarrow cc\}
 \end{aligned}$$

- (a) Welche Sprache $L(G)$ erzeugt die Grammatik? Geben Sie mindestens drei Wörter aus $L(G)$ an und geben Sie eine (möglichst formale) Definition der Sprache an.

/ 3

$$\lambda, abbccc, aabbbbcccccc \in L; L = \{w \in \{a, b, c\}^* \mid \exists n \in \mathbb{N} : w = a^n b^{2n} c^{3n}\}$$

- (b) Die Grammatik G ist nicht kontextsensitiv. Wandeln Sie die Grammatik G in eine kontextsensitive Grammatik G' um, sodass gilt $L(G') = L(G)$.

Hinweis: Sie brauchen G' nicht vollständig anzugeben. Es reicht, wenn Sie die Änderungen angeben, die notwendig sind, um von G zu G' zu kommen.

/ 5

1. Problem: $S \rightarrow \lambda$ und zusätzlich steht S auf einer rechten Seite

$$\begin{aligned}
 G'_1 &= (\{S, B, C, S'\}, \{a, b, c\}, P'_1, S), \\
 P'_1 &= \{S \rightarrow S' \mid \lambda, \\
 &\quad \underline{S' \rightarrow aS'BBCCC \mid abbCCC}, \\
 &\quad CB \rightarrow BC, \\
 &\quad bB \rightarrow bb, \\
 &\quad bC \rightarrow bc, \\
 &\quad cC \rightarrow cc\}
 \end{aligned}$$

2. Problem: $CB \rightarrow BC$ ist nicht kontextsensitiv

$$G' = (\{S, B, C, S', D\}, \{a, b, c\}, P', S),$$

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

$$S' \rightarrow aS'BBCCC \mid abbCCC,$$

$$\underline{CB \rightarrow CD},$$

$$\underline{CD \rightarrow BD},$$

$$\underline{BD \rightarrow BC},$$

$$bB \rightarrow bb,$$

$$bC \rightarrow bc,$$

$$cC \rightarrow cc\}$$

- (c) Zu welcher Sprachklasse der Chomsky-Hierarchie (welchen Sprachklassen der Chomsky-Hierarchie) gehört $L(G) = L(G')$, wenn Sie Ihr Wissen aus der Vorlesung zugrunde legen?

/ 2

	Wahr	Falsch
Typ 0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Typ 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Typ 2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Typ 3	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Aufgabe 3 **8 Punkte**

2013-H-03

Cocke-Younger-Kasami-Algorithmus

/ 8

Gegeben sei die Grammatik G mit

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

$$P = \{S \rightarrow AB \mid BC \mid a,$$

$$A \rightarrow BA \mid BB \mid b,$$

$$B \rightarrow CC \mid a,$$

$$C \rightarrow SS \mid b\}.$$

- (a) In welcher Normalform befindet sich G und zu welchem Typ der Chomsky-Hierarchie zählt G ?

/ 1

Lösung:

Kontextfreie Grammatik (Typ 2) in Chomsky-Normalform.

- (b) Überprüfen Sie mithilfe des Algorithmus von Cocke, Younger und Kasami, ob

$$babba \in L(G).$$

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

/ 7

Lösung:

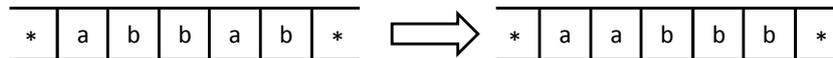
	b	a	b	b	a
$m = 1$	A, C	B, S	A, C	A, C	B, S
$m = 2$	S	A, S	B	S	
$m = 3$	\emptyset	A	A		
$m = 4$	\emptyset	A, C, S			
$m = 5$	B				

$\rightarrow w \notin L(G)$

Aufgabe 4	10 Punkte
2013-H-04	Turingmaschine
	/ 10

Geben Sie eine Turingmaschine $T = (E, B, S, \delta, s_0, F)$ an, die eine Eingabe über $E = \{a, b\}^*$ alphabetisch sortiert.

Beispiel:



Hinweise:

- T steht zu Beginn über dem linken Zeichen des Eingabewortes w .
- T soll über dem linken Zeichen des konvertierten Wortes halten.
- Geben Sie T vollständig an.
- Die Spaltenanzahl entspricht nicht der minimal erforderlichen Anzahl an Bandsymbolen und die Zeilenanzahl nicht der minimal erforderlichen Anzahl an Zuständen.

Möglichkeit 1:

$$T = (\{a, b\}, \{a, b, B, \star\}, \{s_0, s_1, s_2, s_3, s_e, s_{ee}\}, \delta, s_0, \{s_{ee}\})$$

δ	a	b	B	\star		
s_0	(s_0, a, R)	(s_1, B, R)		(s_e, \star, L)		
s_1	(s_1, a, R)	(s_1, b, R)		(s_2, \star, L)		
s_2	(s_3, b, L)	(s_2, b, L)	(s_e, b, L)			
s_3	(s_3, a, L)	(s_3, b, L)	(s_0, a, R)			
s_e	(s_e, a, L)			(s_{ee}, \star, R)		
s_{ee}						

Möglichkeit 2:

$$T = (\{a, b\}, \{a, b, \star\}, \{s_0, s_1, s_2, s_3, s_e, s_{ee}\}, \delta, s_0, \{s_{ee}\})$$

δ	a	b	\star			
s_0	(s_0, a, R)	(s_1, b, R)	(s_e, \star, L)			
s_1	(s_2, b, L)	(s_1, b, R)	(s_e, \star, L)			
s_2		(s_3, a, L)				
s_3	(s_0, a, R)	(s_1, b, R)				
s_e	(s_e, a, L)	(s_e, b, L)	(s_{ee}, \star, R)			
s_{ee}						

Aufgabe 5

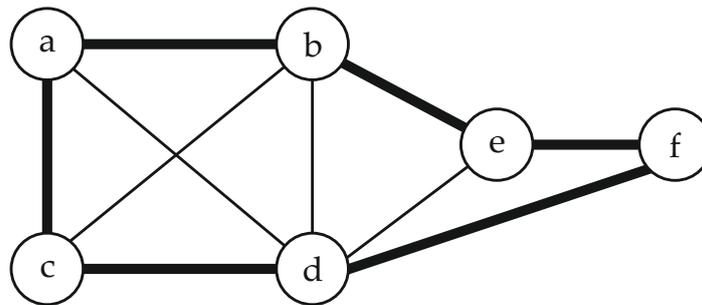
8 Punkte

2013-H-05

Komplexitätstheorie

/ 8

Das *ungerichtete Hamiltonkreisproblem (UHP)* ist die Frage, ob in einem ungerichteten Graphen $G = (V, E)$ ein geschlossener Pfad existiert, der jeden Knoten genau einmal enthält. Im abgebildeten Graphen kennzeichnet der dick gezeichnete Kantenzug einen Hamiltonkreis.



- (a) Wie könnte eine nichtdeterministische Turingmaschine vorgehen, um das UHP für einen gegebenen Graphen **effizient** zu lösen?

Hinweis: Argumentieren Sie umgangssprachlich; Sie müssen die Turingmaschine **nicht** angeben.

/ 2

Lösung: Die Turingmaschine könnte eine Teilmenge der Kanten $E' \subseteq E$ des Graphen raten, und in Polynomialzeit überprüfen, ob E' einen zusammenhängenden und geschlossenen Pfad darstellt, der jeden Knoten $v \in V$ genau einmal enthält. Alternativ könnte die TM auch eine Permutation der Knoten raten und überprüfen, ob sie sich alle auf einem geschlossenen Kantenzug befinden.

- (b) Wie könnte eine deterministische Turingmaschine vorgehen, um das Vorgehen der nichtdeterministischen Turingmaschine bei der Lösung des UHP zu simulieren?

Hinweis: Sie müssen auch hier die Turingmaschine **nicht** angeben.

/ 2

Lösung: Die det. Turingmaschine könnte nacheinander die verschiedenen Teilmengen $E' \subseteq E$ der Menge der Kanten des Graphen auf ihrem Band auflisten und jeweils den (deterministischen) Verifikationsprozess der nichtdeterministischen Turingmaschine durchführen.

- (c) Wie groß ist der Zeitaufwand Ihrer nichtdeterministischen Turingmaschine, bzw. Ihrer deterministischen Turingmaschine in Abhängigkeit von der Knotenanzahl des Graphen $n = |V|$ im schlimmsten Fall?

Lösung:

- Nichtdet: Das Raten läuft in konstanter Zeit c ab, das Verifizieren in Polynomialzeit n^k für ein konstantes k , also ergibt sich ein Gesamtaufwand von $O(c + n^k) = O(n^k)$. (Auch wenn der Verifikationsprozess polynomiell in der Anzahl der Kanten (anstatt der Knoten) ist, wäre der Gesamtaufwand $O((n^2)^k) = O(n^{2k}) = O(n^{k'})$ für eine andere Konstante k' .)
- Det: Der Auflistungsprozess beinhaltet alle Teilmengen der Kantenmenge E , also $p(E) = 2^{|E|}$ Elemente. Die Kantenmenge enthält maximal n^2 Elemente, also ist $p(E) \leq 2^{n^2}$. Für jedes dieser Elemente muss der Verifikationsprozess durchgeführt werden. Also ergibt sich ein Aufwand von $O(2^{n^2} \cdot n^k) = O(2^{n^2})$. Zu beachten ist, dass diese Zeitangabe zwar überexponentiell in der Anzahl der Knoten wächst, aber immer noch „nur“ exponentiell in der Gesamtlänge der Eingabe ist, da die Kanten auch Teil der Eingabe sind.

Aufgabe 6	8 Punkte
2013-H-06	CMOS
	/ 8

Gegeben sei folgenden Schaltfunktion „math“ mit

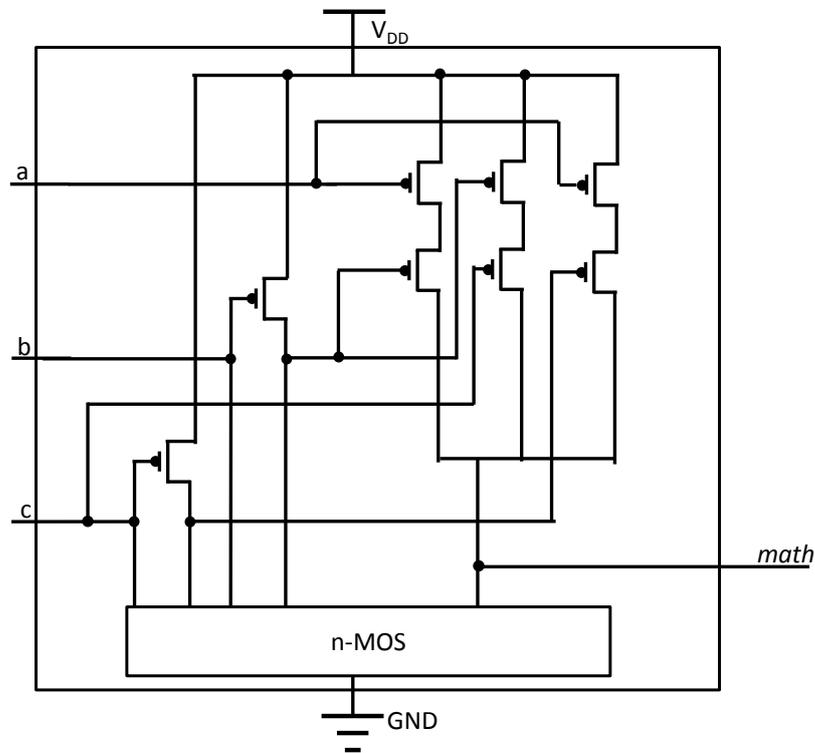
$$math(a, b, c) = (\neg a \wedge b) \vee (b \wedge \neg c) \vee (\neg a \wedge c).$$

(a) Geben Sie eine CMOS-Schaltung für die Funktion *math* an.

Hinweis: Zeichnen Sie nur den PMOS-Bereich auf Transistorebene. Verwenden Sie die vorgegebene Blackbox als NMOS-Bereich und stellen Sie durch geeignete Leitungen eine Verbindung zwischen NMOS- und PMOS-Bereich her.

/ 7

Lösung:



(b) Beschreiben Sie in einem Wort, wie Sie den NMOS-Bereich bei einer CMOS-Schaltung im Vergleich zum PMOS-Bereich aufbauen würden.

/ 1

Lösung:
Komplementär

Aufgabe 7

9 Punkte

2013-H-07

Zahlendarstellung

/ 9

Gegeben sei die folgende Binärsequenz

0100001101011001.

- (a) Geben Sie an, welcher Dezimalzahl der Bitstring entspricht, wenn dieser als Zahl in BCD-Darstellung interpretiert wird.

/ 1

Lösung:

0100 $\hat{=}$ 4; 0011 $\hat{=}$ 3; 0101 $\hat{=}$ 5; 1001 $\hat{=}$ 9;
Also ergibt sich als Zahl 4359.

- (b) Geben Sie an, welcher Dezimalzahl der Bitstring entspricht, wenn dieser als Zahl in 2-Komplement-Darstellung interpretiert wird.

Hinweis: Es reicht, das Ergebnis als Summe darzustellen.

/ 2

Lösung:

$$\begin{aligned} 0100001101011001 &: 2^0 + 2^3 + 2^4 + 2^6 + 2^8 + 2^9 + 2^{14} \\ &= 1 + 8 + 16 + 64 + 256 + 512 + 16384 \end{aligned}$$

Also ergibt sich als Zahl 17.241.

- (c) Geben Sie an, welcher Dezimalzahl der Bitstring entspricht, wenn dieser als Zahl in IEEE 754-Darstellung mit halber Genauigkeit (Mantisse: 10 Bits; Exponent: 5 Bits) interpretiert wird.

Hinweis: Es reicht, das Ergebnis als Summe darzustellen.

/ 4

Lösung:

$$\begin{aligned} &(-1)^0 \cdot (1 + m') \cdot 2^{(c-q)} \\ &= (-1)^0 \cdot (1 + (2^{-1} + 2^{-2} + 2^{-4} + 2^{-6} + 2^{-7} + 2^{-10})) \cdot 2^{16-15} \\ &= (-1)^0 \cdot (1 + (2^{-1} + 2^{-2} + 2^{-4} + 2^{-6} + 2^{-7} + 2^{-10})) \cdot 2 \\ &= 2 + 1 + 2^{-1} + 2^{-3} + 2^{-5} + 2^{-6} + 2^{-9} \\ &(\text{nicht gefordert: Also ergibt sich als Zahl } 3\frac{345}{512}.) \end{aligned}$$

- (d) Der Bitstring sei wieder als Zahl in IEEE-754-Darstellung mit halber Genauigkeit (Mantisse: 10 Bits; Exponent: 5 Bits) zu interpretieren. Addieren Sie zu diesem die folgende Zahl in IEEE-754-Darstellung mit halber Genauigkeit und geben Sie die Summe wieder als Zahl in IEEE-754-Darstellung mit halber Genauigkeit an.

0111110000000000

Hinweis: Beachten Sie, dass es bei der IEEE-754-Darstellung auch Sonderfälle gibt.

/ 2

Lösung:

Da 0111110000000000 die Darstellung für ∞ ist, ergibt sich als Summe wieder ∞ , also 0111110000000000.

Aufgabe 8	8 Punkte
2013-H-08	Huffman-Kodierung
	/ 8

Folgende Zeichenkette sei repräsentativ für Daten, die noch kommen sollen (Leerzeichen werden ignoriert):

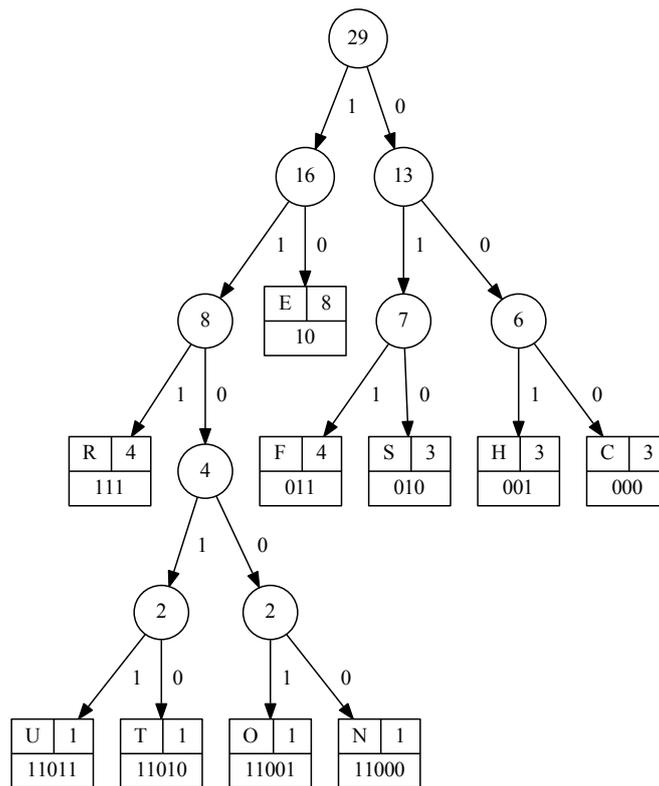
FRECHE FROESCHE FRESSEN FRUECHTE

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	F	R	E	C	H	O	S	N	U	T
Häufigkeit	4	4	8	3	3	1	3	1	1	1

Lösung (Beispiel):



Aufgabe 9 **6 Punkte**

2013-H-09

Speicherorganisation und Cache

/ 6

- (a) Nennen Sie zwei Verfahren, die beschreiben, an welche Stelle im Cache der Datenblock aus dem Hauptspeicher gelangt. Nennen Sie einen Vorteil für jedes der beiden Verfahren. Nutzen Sie hierfür die folgende Tabelle.

/ 3

	Verfahren 1	Verfahren 2
Name	directly mapped	assoziativ
Vorteil	CPU muss jeden Block nur an einer Stelle im Cache suchen; günstigere Hardware; schneller	die zu erwartende Anzahl der Cache-Fehler wird reduziert; wenn noch beliebiger Cacheblock frei, dann kann Cache weiter Daten aufnehmen ohne alte zu löschen; Ausnutzung des kompletten Caches möglich

- (b) Sortieren Sie folgende Speicherarten nach ihrer Zugriffszeit von schnell (1) nach langsam (7). Tragen Sie hierzu die entsprechenden Nummern in folgende Tabelle ein.

/ 3

Speicherart	Zugriffszeit
Magnetplattenspeicher	(5)
Level1-Cache	(2)
Prozessorregister	(1)
optischer Speicher	(6)
Level2-Cache	(3)
DRAM	(4)
Bandspeicher	(7)

Aufgabe 10 **7 Punkte**

2013-H-10

Maschinensprache

/ 7

Moderne Rechner verfügen über verschiedene Sprachebenen.

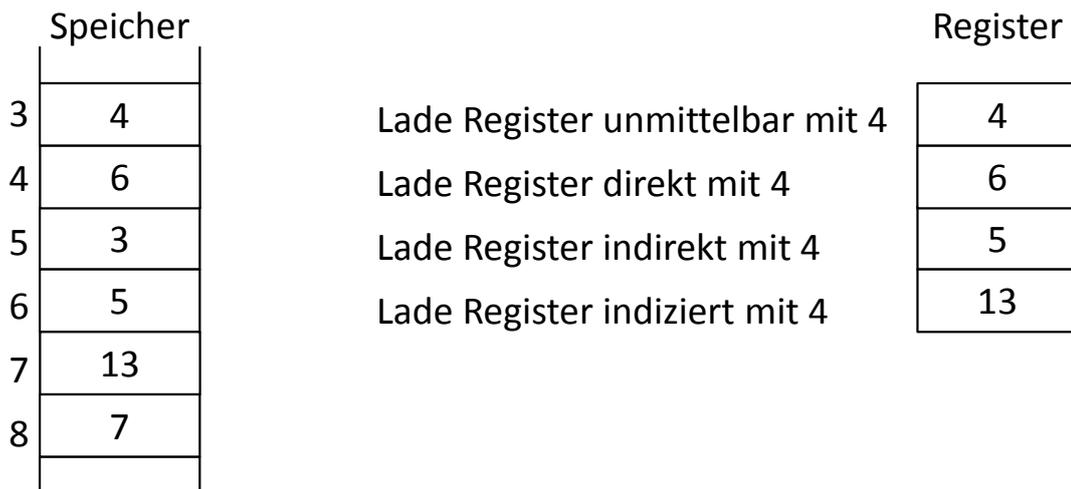
- (a) Nennen und erläutern Sie kurz drei wesentliche Unterschiede zwischen Maschinen- und Assemblersprachen.

/ 3

Maschinensprache	Assemblersprache
Binärcodierung	symbolische Notation
1 Mal kompilieren	2 Mal kompilieren
vom Programmierer kaum lesbar	vom Programmierer einfacher zu lesen
Maschinenbezogen	Anwenderbezogen
näher an der Hardware	nicht so nah an der Hardware
Maschinensprache ist die niederste zugängliche Ebene; Assemblersprache liegt eins drüber (zweitniedrigste Ebene)	
Maschinensprache ist direkt durch CPU ausführbar; Assemblersprache nicht (muss zuerst in Maschinensprache umgewandelt werden)	
Assemblersprache muss zuerst in Maschinensprache umgewandelt werden bevor sie ausgeführt werden kann	

- (b) Geben Sie in der nachfolgenden Abbildung an, welcher Wert bei der jeweiligen Adressierungsart in dem Register gespeichert wird. Tragen Sie dazu den entsprechenden Wert in das jeweils nebenstehende Registerfeld ein. Nehmen Sie ein Indexregister von 3 an.

/ 4



Aufgabe 11

6 Punkte

2013-H-11

Betriebssystem

/ 6

Für fünf bereite Prozesse P_1, \dots, P_5 seien folgende CPU-Zeiten zu erwarten:

$$Z(P_1) = 22ms$$

$$Z(P_2) = 12ms$$

$$Z(P_3) = 7ms$$

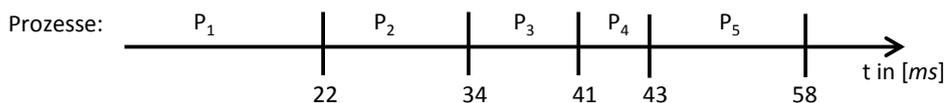
$$Z(P_4) = 2ms$$

$$Z(P_5) = 15ms$$

- (a) Nennen Sie ein einfaches Zuteilungsverfahren und veranschaulichen Sie die Zuteilung der Rechenzeit für $P_i, i \in \{1, 2, 3, 4, 5\}$ nach diesem Verfahren.

/ 1

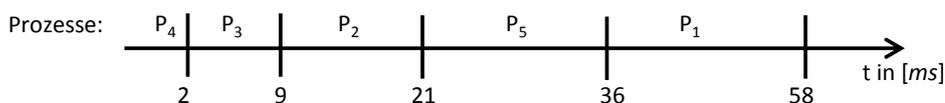
Verfahren: First-Come-First-Served



- (b) Teilen Sie die Rechenzeit gemäß eines Prioritätsgesteuerten Zuteilungsverfahrens ein. Geben Sie dafür zunächst ein Verfahren an, nachdem Sie die Priorität für die einzelnen Prozesse festlegen und veranschaulichen Sie auf dieser Basis die Zuteilung der Rechenzeit für $P_i, i \in \{1, 2, 3, 4, 5\}$.

/ 2

Verfahren: Shortest-Job-First, Longest-Job-First, $t_i > t_j$



- (c) Teilen Sie die Rechenzeit gemäß dem Round-Robin-Verfahren zu. Die Zeitscheibe sei dabei in feste Zeitspannen der Länge $Z = 10ms$ unterteilt. Veranschaulichen Sie die Zuteilung der Rechenzeit für $P_i, i \in \{1, 2, 3, 4, 5\}$ nach diesem Verfahren.

/ 3

