

16.02.2015

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

Name: _____ Vorname: _____

Matr.-Nr.: _____ Semester: _____ (WS 2014/15)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-19).

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)	(8)	(10)	(8)	(8)	(9)	(10)	(7)	(6)	(10)	(5)						(90)

Aufgabenübersicht

1) Endliche Automaten (9 Punkte)	2
2) Pumping-Lemma und Chomsky-Klassen (8 Punkte)	4
3) Grammatiken (10 Punkte)	5
4) Turingmaschine (8 Punkte)	7
5) Berechenbarkeit (8 Punkte)	9
6) CMOS und Schaltnetze (9 Punkte)	11
7) Zahldarstellung und Kodierung (10 Punkte)	13
8) Huffman-Kodierung (7 Punkte)	15
9) Von-Neumann-Architektur (6 Punkte)	16
10) Assembler, Adressierungsarten (10 Punkte)	17
11) Betriebssysteme (5 Punkte)	19

Aufgabe 1 **9 Punkte**

2015-H-01

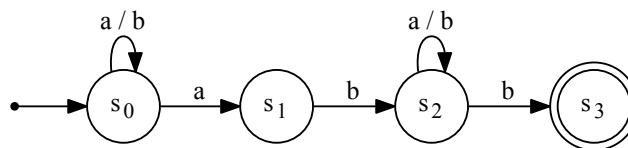
Endliche Automaten

/ 9

Gegeben sei der folgende nichtdeterministische endliche Automat A mit

$$A = (\{a, b\}, \{s_0, \dots, s_3\}, \delta, s_0, \{s_3\})$$

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**. Geben Sie insbesondere ein Zustandsüberföhrungsdiagramm an.

/ 4

$$A' = \left(\underbrace{\quad}_{E'}, \underbrace{\quad}_{S'}, \delta', \underbrace{\quad}_{s'_0}, \underbrace{\quad}_{F'} \right)$$

δ' :

- (b) Geben Sie einen zu A äquivalenten **deterministischen Kellerautomaten** A'' an, der lediglich **zwei Zustände** s''_0 und s_e verwendet. Geben Sie A'' vollständig an.

/ 5

$$A'' = \left(\underbrace{\quad}_{E''}, \underbrace{\{s''_0, s_e\}}_{S''}, \underbrace{\quad}_{K''}, \delta'', s''_0, k_0, \underbrace{s_e}_{F''} \right)$$

$\delta'' :$

Aufgabe 2**8 Punkte**

2015-H-02

Pumping-Lemma und Chomsky-Klassen

/ 8

Gegeben sei die Sprache

$$L = \{a^m b^n c^{mn} \mid m, n \in \mathbb{N}_0\}$$

Es gilt beispielsweise:

- $\lambda, a, aa, aaa, \dots; b, bb, bbb, \dots; abc, aabcc, abbcc, aabbcccc \in L,$
- $ab, ac, bc, abcc, aabccc \notin L.$

(a) Zeigen Sie mit dem Pumping-Lemma für reguläre Sprachen, dass L nicht regulär ist.

/ 6

(b) Skizzieren Sie kurz, wie Sie vorgehen würden, um zu zeigen, dass L vom Chomsky-Typ 1 ist.

/ 1

(c) Angenommen, das Negativergebnis aus (a) und das hypothetische positive Ergebnis aus (b) seien bekannt.

- In welchen Chomsky-Klassen könnte L dann liegen?
- In welchen liegt L gewiss?

/ 1

Aufgabe 3**10 Punkte**

2015-H-03

Grammatiken

/ 10

Gegeben sei eine kontextfreie Grammatik durch

$$G = (\{A, B, E, F, S\}, \{a, b, c, d, e, f\}, P, S) \quad \text{und}$$

$$P = \{S \rightarrow \lambda \mid A \mid EF \mid aSb,$$

$$A \rightarrow c \mid B,$$

$$B \rightarrow d,$$

$$E \rightarrow e \mid d \mid FF,$$

$$F \rightarrow f \mid SA\}$$

- (a) Geben Sie eine vollständige Ableitungsfolge **oder** einen Ableitungsbaum für das Wort $aadb\bar{b} \in L(G)$ an.

/ 1

- (b) Wandeln Sie G mit dem Algorithmus aus der Vorlesung in eine äquivalente Grammatik G_{CNF} in Chomsky-Normalform (CNF) um. Geben Sie für jeden der vier Schritte explizit an, welche Produktionen hinzukommen oder wegfallen. (Sie müssen nicht für jeden Schritt die vollständige Definition der Teilgrammatiken angeben.)

/ 4

- (c) Leiten Sie das Wort $aadb\bar{b}$ bezüglich G_{CNF} (aus Teil b) ab.

/ 1

(d) Gegeben sei eine zu G äquivalente Grammatik in Greibach-Normalform (GNF) durch

$$G_{GNF} = (\{A_1, A_4, A_5, A_7, B_7, S'\}, \{a, b, c, d, e, f\}, P_{GNF}, S')$$

und

$$P_{GNF} = \{S' \rightarrow \lambda \mid A_4,$$

$$A_1 \rightarrow c \mid d,$$

$$A_4 \rightarrow c \mid d \mid aA_4A_5A_1A_7A_7 \mid dB_7A_7A_7 \mid aA_4A_5A_1B_7A_7A_7 \mid aA_5A_1A_7A_7 \mid dA_1A_7A_7 \\ \mid aA_4A_5 \mid cA_7A_7 \mid eA_7A_1A_7A_7 \mid dA_7 \mid eA_7A_1B_7A_7A_7 \mid fB_7A_7A_7 \\ \mid aA_5A_1B_7A_7A_7 \mid dA_7A_1A_7A_7 \mid cA_1B_7A_7A_7 \mid aA_5 \mid dA_1B_7A_7A_7 \mid eA_7 \\ \mid dA_7A_7 \mid dA_7A_1B_7A_7A_7 \mid cB_7A_7A_7 \mid fA_7A_7 \mid cA_1A_7A_7,$$

$$A_5 \rightarrow b,$$

$$A_7 \rightarrow c \mid d \mid f \mid dA_1B_7 \mid aA_4A_5A_1 \mid eA_7A_1B_7 \mid aA_5A_1B_7 \mid fB_7 \mid aA_5A_1 \mid eA_7A_1 \\ \mid dA_7A_1B_7 \mid cA_1B_7 \mid cA_1 \mid dB_7 \mid aA_4A_5A_1B_7 \mid dA_7A_1 \mid cB_7 \mid dA_1,$$

$$B_7 \rightarrow aA_4A_5A_1A_7A_1 \mid cA_1A_7A_1 \mid dB_7A_7A_1 \mid aA_4A_5A_1B_7A_7A_1 \mid dA_7A_1A_7A_1B_7 \\ \mid aA_4A_5A_1B_7A_7A_1B_7 \mid aA_5A_1B_7A_7A_1B_7 \mid eA_7A_1A_7A_1 \mid cA_1B_7A_7A_1B_7 \\ \mid cB_7A_7A_1 \mid aA_5A_1A_7A_1B_7 \mid dA_7A_1B_7 \mid dA_7A_1B_7A_7A_1 \mid cB_7A_7A_1B_7 \\ \mid aA_4A_5A_1A_7A_1B_7 \mid dB_7A_7A_1B_7 \mid dA_7A_1A_7A_1 \mid dA_1A_7A_1B_7 \mid cA_7A_1 \\ \mid eA_7A_1A_7A_1B_7 \mid aA_5A_1B_7A_7A_1 \mid fB_7A_7A_1 \mid cA_1A_7A_1B_7 \mid cA_7A_1B_7 \\ \mid dA_7A_1B_7A_7A_1B_7 \mid fB_7A_7A_1B_7 \mid dA_1A_7A_1 \mid dA_1B_7A_7A_1 \mid fA_7A_1B_7 \\ \mid cA_1B_7A_7A_1 \mid eA_7A_1B_7A_7A_1B_7 \mid aA_5A_1A_7A_1 \mid dA_1B_7A_7A_1B_7 \mid fA_7A_1 \\ \mid dA_7A_1 \mid eA_7A_1B_7A_7A_1\}$$

/ 4

- Wie sind die Produktionen von GNF-Grammatiken allgemein definiert?

- Leiten Sie \boxed{aadb} bezüglich G_{GNF} ab.
Hinweis: Sie benötigen nur die Nonterminale S', A_4 und A_5 .

- Wie viele Ableitungsschritte werden **allgemein** bei GNF-Grammatiken benötigt, um ein Wort der Länge n abzuleiten? Wie viele sind es bei CNF-Grammatiken?

Aufgabe 4**8 Punkte**

2015-H-04

Turingmaschine

/ 8

Gegeben sei die Sprache

$$L = \left\{ w = b_1 \dots b_n \in \{0, 1\}^n \mid \sum_{i=1}^n u(b_i) = \sum_{k=1}^n g(b_k) \right\} \text{ und}$$

$$u(b_i) = \begin{cases} b_i & \text{falls } i \text{ ungerade,} \\ 0 & \text{sonst.} \end{cases}$$

$$g(b_k) = \begin{cases} b_k & \text{falls } k \text{ gerade,} \\ 0 & \text{sonst.} \end{cases}$$

L enthält somit alle Wörter, für die die Anzahl der Einsen an den geraden Stellen gleich der Anzahl der Einsen an den ungeraden Stellen ist. Es gilt also

- $\lambda, 0, 11, 011, 0110, 1001, 010100101 \in L$,
- $1, 01, 10, 101, 111, 1100001 \notin L$.

Geben Sie eine Turingmaschine $T = (E, B, S, \delta, s_0, F)$ an, für die gilt $L(T) = L$. T ist also ein Akzeptor, der in einem Endzustand anhält, falls $w \in L$. Geben Sie T vollständig an.

Hinweise:

- T darf die Bandinschrift w beliebig überschreiben.
- Der Schreib-/Lesekopf steht zu Beginn über dem linken Zeichen der Eingabe und terminiert, falls T akzeptiert, über dem \star -Feld rechts neben dem rechten beschriebenen Feld.
- Die Spaltenanzahl der vorgegebenen Tabelle (nächste Seite) entspricht nicht der minimal erforderlichen Anzahl an Bandsymbolen und die Zeilenanzahl nicht der minimal erforderlichen Anzahl an Zuständen.

Aufgabe 5**8 Punkte**

2015-H-05

Berechenbarkeit

Für eine Turingmaschine T sei $s(T)$ die Anzahl an Zustandsübergängen, die T **bei Eingabe des leeren Wortes** bis zum Halten durchführt. Es gelte $s(T) =_{def} 0$, falls die Turingmaschine nicht anhält.

Beispiel: Läuft eine Turingmaschine T zwei Schritte nach rechts und hält dann, gilt $s(T) = 2$. Welche Bandsymbole sie dabei schreibt, ist irrelevant.

Gegeben sei weiter durch \mathcal{T}_n für $n \in \mathbb{N}$ die Menge aller Turingmaschinen mit n Zuständen über dem Bandalphabet $\{1, \star\}$. Dann sei

$$S(n) =_{def} \max(\{s(T) \mid T \in \mathcal{T}_n\})$$

die maximale Anzahl an Zustandsübergängen $s(T)$, die irgendeine solche **nicht endlos laufende** Turingmaschine T mit n Zuständen durchführt.

- (a) Geben Sie $S(1)$ an und begründen Sie Ihre Wahl.

Hinweis: Denken Sie daran, dass Sie nur die Bandsymbole 1 und \star nutzen dürfen.

- (b) Wie müsste man (als Mensch) vorgehen, um für ein festes $k \in \mathbb{N}$ den Wert $S(k)$ zu bestimmen?

(c) Begründen Sie, dass $S(n)$ im Allgemeinen nicht berechenbar ist.

Hinweis: Das Halteproblem ist (auch bei leerer Eingabe) nicht berechenbar.

/ 3

Aufgabe 6 **9 Punkte**

2015-H-06

CMOS und Schaltnetze

/ 9

Die Funktion $f_n : \mathbb{B}^n \rightarrow \mathbb{B}$ sei bei Eingabe eines Binärstrings $B = b_1b_2 \dots b_{n-2}b_{n-1}b_n$ wie folgt definiert:

$$f_n(b_1, \dots, b_n) = \left(\sum_{i=1}^n b_i \right) \text{MOD } 2$$

f_n berechnet also den Rest bei Division durch 2 der Summe aller Einsen des Binärstrings B .

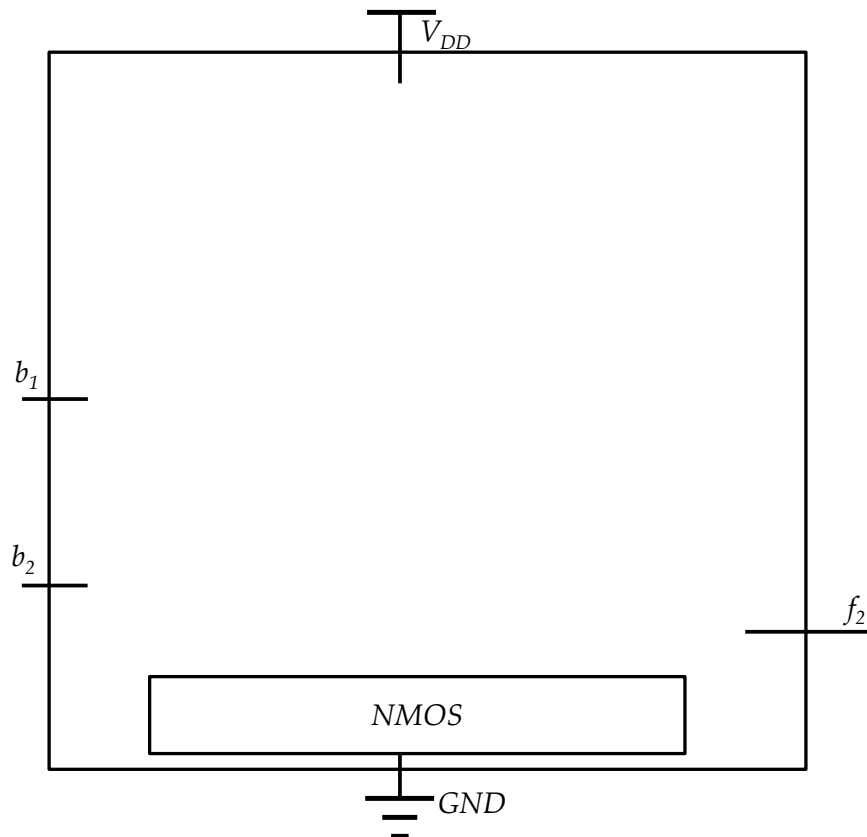
- (a) Geben Sie eine CMOS-Schaltung an, die zwei Eingaben $b_1, b_2 \in \mathbb{B}$ erhält, diese als Binärstring $b_1b_2 \in \mathbb{B}^2$ interpretiert und als Ausgabe den Wert $f_2(b_1b_2)$ berechnet.

Hinweise:

- Sie können sich zuerst überlegen, welchen Booleschen Ausdruck f_2 darstellt. Als Hilfsmittel können Sie die leere Tabelle verwenden, sie wird aber nicht gewertet.
- Zeichnen Sie nur den PMOS-Teil mit entsprechenden Verbindungen zur NMOS-Blackbox.

/ 7

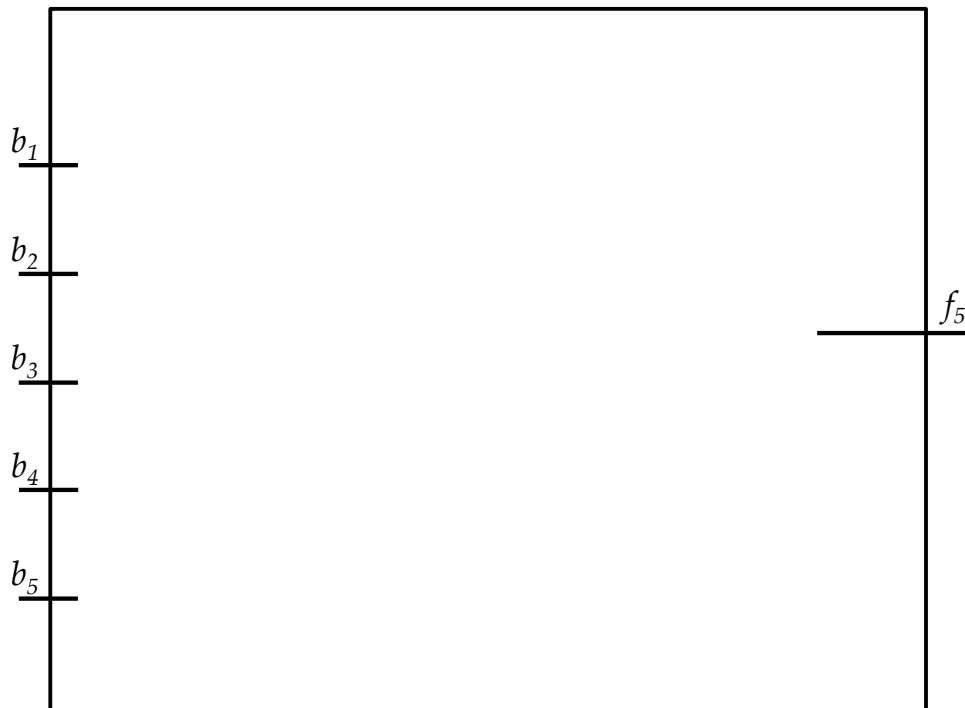
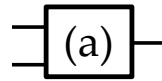
b_1	b_2	f_2



- (b) Geben Sie ein Schaltnetz an, das als Eingabe einen Bitstring $b_1 \dots b_5$ erhält und als Ausgabe $f_5(b_1, \dots, b_5)$ berechnet. Verwenden Sie hierfür als Blackbox die Schaltung für f_2 aus Teilaufgabe (a) (siehe unten; Sie dürfen die Blackbox auch verwenden, wenn Sie (a) nicht gelöst haben).

/ 2

Blackbox ($f_2: \mathbb{B}^2 \rightarrow \mathbb{B}$):



Aufgabe 7**10 Punkte**

2015-H-07

Zahldarstellung und Kodierung

/ 10

Gegeben seien die beiden Binärstrings $A, B \in \{0, 1\}^8$ durch

$$A = 10010110$$

$$B = 01011000$$

- (a) Interpretieren Sie A und B als Dualzahlen im 2er-Komplement. Addieren Sie A und B **ohne** sie vorher in Dezimalwerte umzuwandeln und geben Sie dann **das Ergebnis** im 2er-Komplement **und** als Dezimalwert an.

/ 2

- (b) Interpretieren Sie A und B als Gleitkommazahlen mit einem Vorzeichenbit, 4-Bit-Charakteristik und 3-Bit-Mantisse. Multiplizieren Sie A und B **im Gleitkommaformat**. Dokumentieren Sie Ihren Rechenweg in Stichworten und geben Sie dann **das Ergebnis** als Gleitkommazahl **und** Dezimalwert an.

/ 5

- (c) Interpretieren Sie A und B als die einzigen Codewörter eines Codes. Wie viele Bit-Fehler können maximal erkannt und wie viele maximal korrigiert werden?

/ 2

Fehler-erkennbar: _____ Bit

Fehler-korrigierbar: _____ Bit

- (d) Geben Sie ein weiteres Codewort C an, so dass der resultierende Code mindestens 2-Fehler-erkennbar ist.

/ 1

A	1 0 0 1 0 1 1 0
B	0 1 0 1 1 0 0 0
C	

Aufgabe 8 **7 Punkte**

2015-H-08

Huffman-Kodierung

/ 7

Folgende Zeichenkette der Länge 23 sei repräsentativ für Daten, die noch kommen sollen
(– wird explizit kodiert):

NARRI-NARRO-ALAAF-HELAU

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	N	A	R	I	-	O	L	F	H	E	U
Häufigkeit											

Zeichen	Code
N	
A	
R	
I	
-	
O	
L	
F	
H	
E	
U	

Aufgabe 9**6 Punkte**

2015-H-09

Von-Neumann-Architektur

Der Von-Neumann-Rechner enthält die wesentlichen Bestandteile

Eingabewerk, Speicherwerk, Ausgabewerk, Rechenwerk und Steuerwerk.

In diesem Zusammenhang treten die im Folgenden genannten Begriffe auf. Beschreiben Sie diese in kurzen Worten. Gehen Sie dabei insbesondere auf deren **Funktion** und deren **Zuordnung** zu den Bestandteilen des Von-Neumann-Rechners ein.

(a) Register

(b) Central Processing Unit (CPU)

(c) Maschinenbefehlszyklus

(d) Cache

Aufgabe 10**10 Punkte****2015-H-10****Assembler, Adressierungsarten**

/ 10

Die Befehle einer Assembler-Sprache seien folgendermaßen aufgebaut, wobei Q für Quelle steht und Z für Ziel:

$$\text{OpCode } Q1, (Q2,) Z$$

Für **unmittelbare Adressierung** wird das Präfix # verwendet, **direkte Adressierung** geben wir ohne Präfix an. Sprungbefehle sind unbedingt (JUMP L) oder bedingt (JUMPZERO L).

Zusätzlich verwenden wir den Befehl

$$\text{STORE_IND } Q, Z$$

um Q in einen von den Registern getrennten Hauptspeicherraum zu schreiben, wobei **indirekte Adressierung bezogen auf Z** verwendet wird. Adressen von Registern (im Rechenwerk) sind von der Form R_k ($k \in \mathbb{N}_0$), Adressen von Speicherzellen (im Hauptspeicher) sind natürliche Zahlen k ($k \in \mathbb{N}_0$).

Gegeben sei das folgende Assemblerprogramm.

	LOAD	R8 ACC	
outerloop	JUMPZERO	fertig	
	STORE	R8 R1	
	STORE	R8 R2	
	STORE	#0 R3	
innerloop	ADD	R2 R3 R3	
	SUBTRACT	R1 #1 R1	
	LOAD	R1 ACC	
	JUMPZERO	store	
	JUMP	innerloop	
store	STORE_IND	R3 R2	*STORE mit indirekter Adressierung*
	SUBTRACT	R8 #1 R8	
	LOAD	R8 ACC	
	JUMP	outerloop	
fertig	STORE_IND	R8 R8	*STORE mit indirekter Adressierung*
	HALT		

- (a) Welcher Wert wurde nach **dem ersten** Durchlauf der „**outerloop**“ an welche Adresse **im Hauptspeicher** geschrieben, wenn zu Beginn der Rechnung

$$R8 = n \in \mathbb{N}$$

gilt?

/ 6

- (b) Welche Werte stehen dann nach einem **vollständigen** Durchlauf des **ganzen** Programms an welchen Stellen **im Hauptspeicher**?

/ 3

- (c) Was würde passieren, wenn statt STORE_IND nur STORE verwendet würde?

/ 1

Aufgabe 11 **5 Punkte**

2015-H-11

Betriebssysteme

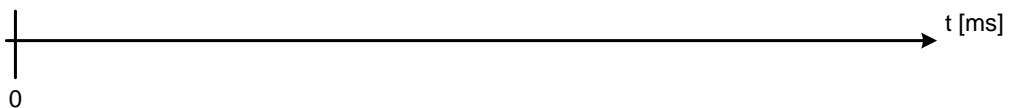
/ 5

In der Warteschlange eines Prozessors befinden sich die folgenden Prozesse $P1$ bis $P4$, die in dieser Reihenfolge beim Prozessor ankommen:

Prozesse	CPU-Zeit in ms	Priorität
P1	20	1
P2	35	2
P3	60	1
P4	15	1

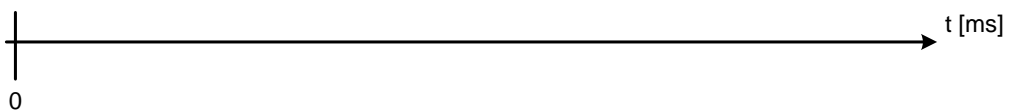
- (a) Teilen Sie den Prozessen Rechenzeit gemäß dem Round Robin Verfahren zu. Die Zeitscheibe sei dabei in feste Zeitspannen der Länge $Z = 15\text{ms}$ unterteilt. Veranschaulichen Sie Ihr Ergebnis auf dem gegebenen Zeitstrahl.

/ 2



- (b) Gehen Sie jetzt davon aus, dass $P1$, nachdem er 10ms bearbeitet wurde, auf das Ergebnis von $P2$ warten muss. Die Zeitspannen der Zeitscheibe betragen weiterhin $Z = 15\text{ms}$. Veranschaulichen Sie Ihr Ergebnis auf dem gegebenen Zeitstrahl.

/ 1



- (c) Betrachten Sie nun ein prioritätenbasiertes Zuteilungsverfahren (2 ist dabei die höchste Prioritätsstufe). Nennen Sie zwei Möglichkeiten, die CPU-Zeit zwischen Prozessen gleicher Priorität aufzuteilen und geben Sie für beide Möglichkeiten die Reihenfolge an, in der die Prozesse $P1$ bis $P4$ abgearbeitet werden.

/ 2