

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

Name: _____ Vorname: _____

Matr.-Nr.: _____ Semester: _____ (WS 2017/18)

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 |
| (7) | (9) | (12) | (12) | (8) | (10) | (7) | (6) | (6) | (7) | (6) | | | | | | (90) |

Aufgabenübersicht

| | |
|--|----|
| 1) Multiple-Choice (7 Punkte) | 2 |
| 2) Minimierung endlicher Automaten (9 Punkte) | 3 |
| 3) Monotone Grammatiken / Pumping-Lemma (12 Punkte) | 5 |
| 4) Turingmaschinen (12 Punkte) | 7 |
| 5) Berechenbarkeit (8 Punkte) | 9 |
| 6) CMOS / Schaltnetze (10 Punkte) | 10 |
| 7) Huffman (7 Punkte) | 12 |
| 8) Kodierung – Fano-Bedingung (6 Punkte) | 13 |
| 9) Rechnerarchitektur (6 Punkte) | 14 |
| 10) Programmiersprachen (7 Punkte) | 15 |
| 11) Betriebssysteme (6 Punkte) | 16 |

Aufgabe 1**7 Punkte****2018-H-01****Multiple-Choice**

/ 7

Kreuzen Sie für die folgenden Aussagen an, ob diese jeweils **wahr**, **falsch** oder nach heutigem Kenntnisstand nicht beantwortbar (**unbekannt**) sind. Für falsche Antworten werden **keine** Punkte abgezogen.

| | Wahr | Falsch | Unbekannt |
|---|--------------------------|--------------------------|--------------------------|
| Deterministische Kellerautomaten akzeptieren eine kleinere Sprachklasse als nichtdeterministische Kellerautomaten. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| NP -vollständige Probleme sind die schwierigsten Probleme, die es gibt. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Es gilt $NP \subseteq P$. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Mit irrationalen Zahlen wie $\sqrt{2}, e, \pi, i$ usw. kann man im Rechner nicht exakt rechnen. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Es gibt für jede beliebige Teilmenge $R \subseteq \mathbb{R}$ der reellen Zahlen eine Zahlendarstellung, die alle Zahlen aus R explizit und exakt im Rechner repräsentiert. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Assemblersprachen haben eine niedrigere Berechnungsmächtigkeit (in Bezug auf die Berechenbarkeit von Funktionen) als objektorientierte Sprachen. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Turing-vollständige Programmiersprachen besitzen die höchste Berechnungsmächtigkeit, die es gibt. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Aufgabe 2 **9 Punkte**

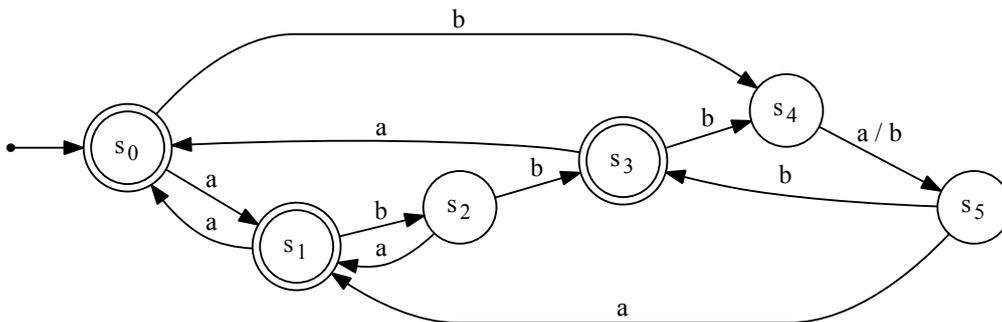
2018-H-02

Minimierung endlicher Automaten

/ 9

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

δ :



- (a) Minimieren Sie A und geben Sie den minimierten Automaten A' vollständig an. Geben Sie insbesondere die Minimierungstabelle und ein Zustandsüberförungsdiagramm 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 3**12 Punkte****2018-H-03****Monotone Grammatiken / Pumping-Lemma**

/ 12

Gegeben sei die Sprache

$$L = \{w \in \{a, b, c, d, e\}^* \mid w = a^m b^n c^{m+n} d^m e^n \text{ mit } m, n \in \mathbb{N}_+\}$$

Bspw. gilt:

$$abccde, \underbrace{aabcccdde}_{m=5}, \underbrace{aaaaa}_{n=6} \underbrace{bbbbbb}_{m+n=11} \underbrace{ccccccccc}_{m} \underbrace{dddd}_{n} \underbrace{eeeeee} \in L$$

$$\lambda, a, b, abc, cde, aaccdd, bbccee \notin L$$

- (a) Geben Sie eine **monotone oder kontextsensitive** Grammatik G an mit $L(G) = L$. Geben Sie G vollständig an.

/ 6

- (b) Zeigen Sie mithilfe des Pumping-Lemmas für EA-Sprachen, dass L **nicht rechtslinear** ist.

/ 4

- (c) Kreuzen Sie an, welche Aussagen **nach den Ergebnissen aus (a) und (b)** wahr bzw. falsch bzw. nicht beantwortbar (unbekannt) sind. Für falsche Antworten werden **keine** Punkte abgezogen.

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

Hinweise:

- Nehmen Sie die Aussagen aus der Fragestellung zu (a) und (b) als wahr an, auch wenn Sie die Teilaufgabe(n) nicht gelöst haben.
- Nehmen Sie an, dass außer diesen beiden Aussagen keine Kenntnisse über L verfügbar sind.

| | Wahr | Falsch | Unbekannt |
|-------------------|--------------------------|--------------------------|--------------------------|
| L ist vom Typ 0 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| L ist vom Typ 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| L ist vom Typ 2 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| L ist vom Typ 3 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Aufgabe 4 **12 Punkte**

2018-H-04

Turingmaschinen

/ 12

Gegeben sei wieder die Sprache L aus Aufgabe 3:

$$L = \{w \in \{a, b, c, d, e\}^* \mid w = a^m b^n c^{m+n} d^m e^n \text{ mit } m, n \in \mathbb{N}_+\}$$

Gegeben sei außerdem die Turingmaschine

$$T = (\{a, b\}, \{a, \dots, e, A, B, \star\}, \{s_0, \dots, s_4, s_X, s_{z1}, s_{z2}\}, \delta, s_0)$$

die auf Eingabe $w = a^m b^n$ hin die „zugehörige“ Ausgabe $f_T(w) = a^m b^n c^{m+n} d^m e^n \in L$ erzeugt. T berechnet also folgende Funktion:

$$f_T : \{a\}^* \times \{b\}^* \rightarrow \{a, b, c, d, e\}^* : f_T(a^m b^n) = a^m b^n c^{m+n} d^m e^n \text{ für } m, n \in \mathbb{N}_+$$

| δ | a | b | c | d | e | A | B | \star |
|----------|------------------|------------------|------------------|------------------|------------------|---------------|------------------|-------------------|
| s_0 | (s_X, a, N) | | | | | | | |
| s_X | (s_1, A, R) | (s_1, B, R) | (s_{z2}, c, L) | | | | | |
| s_1 | (s_1, a, R) | (s_1, b, R) | (s_1, c, R) | | | | (s_1, B, R) | (s_{z1}, c, L) |
| s_2 | (s_2, a, R) | | (s_2, c, R) | (s_2, d, R) | | | (s_2, B, R) | (s_{z2}, d, L) |
| s_3 | (s_3, a, R) | (s_3, b, R) | | | | | (s_4, b, R) | |
| s_4 | | | (s_4, c, R) | (s_4, d, R) | (s_4, e, R) | | (s_4, B, R) | (s_{z2}, e, L) |
| s_{z1} | (s_{z1}, a, L) | (s_{z1}, b, L) | (s_{z1}, c, L) | | | (s_X, A, R) | (s_X, B, R) | |
| s_{z2} | (s_{z2}, a, L) | (s_3, b, R) | (s_{z2}, c, L) | (s_{z2}, d, L) | (s_{z2}, e, L) | (s_2, a, R) | (s_{z2}, B, L) | (s_3, \star, R) |

(a) Geben Sie die Konfigurationsfolgen von T bei folgenden Eingaben an:

- $w = a$:

/ 5

- $w = b$:

(b) Warum ist es – streng nach Definition von Turingmaschinen – in Ordnung, dass sich T für diese (in Bezug zu f_T so ähnlichen) Eingaben a bzw. b so unterschiedlich verhält?

Hinweis: Überlegen Sie, inwiefern diese beiden Eingaben „untypisch“ für die angegebene Funktion f_T sind.

/ 2

(Wiederholung der Zustandsübergangstabelle.)

| δ | a | b | c | d | e | A | B | \star |
|----------|------------------|------------------|------------------|------------------|------------------|---------------|------------------|-------------------|
| s_0 | (s_X, a, N) | | | | | | | |
| s_X | (s_1, A, R) | (s_1, B, R) | (s_{z2}, c, L) | | | | | |
| s_1 | (s_1, a, R) | (s_1, b, R) | (s_1, c, R) | | | | (s_1, B, R) | (s_{z1}, c, L) |
| s_2 | (s_2, a, R) | | (s_2, c, R) | (s_2, d, R) | | | (s_2, B, R) | (s_{z2}, d, L) |
| s_3 | (s_3, a, R) | (s_3, b, R) | | | | | (s_4, b, R) | |
| s_4 | | | (s_4, c, R) | (s_4, d, R) | (s_4, e, R) | | (s_4, B, R) | (s_{z2}, e, L) |
| s_{z1} | (s_{z1}, a, L) | (s_{z1}, b, L) | (s_{z1}, c, L) | | | (s_X, A, R) | (s_X, B, R) | |
| s_{z2} | (s_{z2}, a, L) | (s_3, b, R) | (s_{z2}, c, L) | (s_{z2}, d, L) | (s_{z2}, e, L) | (s_2, a, R) | (s_{z2}, B, L) | (s_3, \star, R) |

(c) Erklären Sie für jeden der Zustände in S **kurz**, welche Wirkung er im Verlauf der Rechnung hat. Beschreiben Sie außerdem **kurz** das Gesamtverhalten von T .

• s_0 :

| |
|-----|
| / 5 |
|-----|

• s_X :

• s_1 :

• s_2 :

• s_3 :

• s_4 :

• s_{z1} :

• s_{z2} :

• Gesamtverhalten von T :

Aufgabe 5**8 Punkte****2018-H-05****Berechenbarkeit**

/ 8

Sei U^X ein **universeller Automat** mit folgenden Eigenschaften:

- Für $X \in \{TM, LBA, KA, EA\}$ ist U^X ein Automat vom Typ X .
- Als Eingabe erhält U^X :
 - einen **Automaten** A vom Typ X und
 - ein Wort w .
- U^X akzeptiert seine Eingabe genau dann, wenn A die Eingabe w akzeptiert.

Zum Beispiel wäre U^{TM} eine universelle Turingmaschine.

(a) Geben Sie die Sprache $L(U^X)$ formal an.

$$L(U^X) =$$

/ 2

(b) Argumentieren Sie, warum U^{EA} nicht existieren kann.

Hinweise:

- Überlegen Sie, welcher Sprachklasse $L(U^{EA})$ angehören müsste und welche aus der Vorlesung bekannte Eigenschaft dieser Sprachklasse dann nicht erfüllt sein kann.
- Versuchen Sie, eine nicht notwendigerweise formale, aber schlüssige Begründung zu finden. Formulieren Sie umgangssprachlich präzise.

/ 4

(c) Philosophieren Sie **sehr kurz**, ob ein U^{KA} bzw. ein U^{LBA} existieren kann oder nicht.

Hinweise:

- Beachten Sie die Hinweise unter (b). Eventuell reicht eine kurze Reflexion, warum Ihre Argumentation dort auch hier gilt – oder eben nicht gilt.

/ 2

Aufgabe 6

10 Punkte

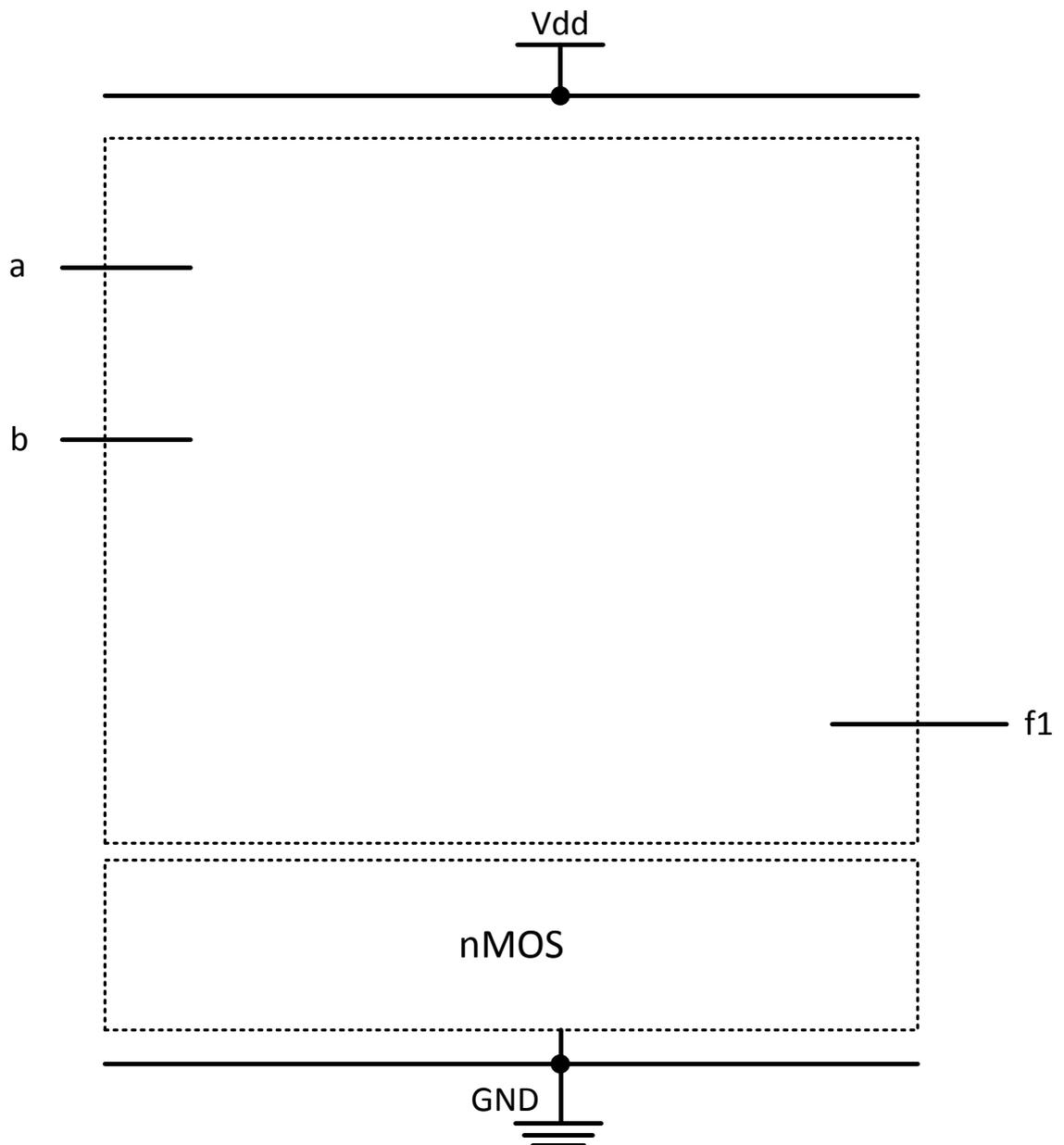
2018-H-06

CMOS / Schaltnetze

/ 10

- (a) Zeichnen Sie den **pMOS-Teil** eines Äquivalenzgatters: Das Ausgangssignal f_1 muss also genau dann auf Eins gesetzt werden, wenn beide Eingangssignale a und b den gleichen Wert haben.

/ 6

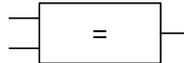


- (b) Benutzen Sie das in Teilaufgabe (a) beschriebene Äquivalenzgatter, um ein **Schaltnetz** zu entwerfen, das die vier Eingangssignale w , x , y und z auf Parität prüft. Das Ausgangssignal f_2 muss genau dann auf Eins gesetzt werden, wenn eine gerade Anzahl an Einsen anliegt. Auch keine Eins gilt als gerade Anzahl. Sie dürfen das Äquivalenzgatter als Blackbox mit zwei Eingängen und einem Ausgang verwenden.

Hinweis: Sie dürfen auch andere aus der Vorlesung bekannte Gatter verwenden. Dies ist aber nicht notwendig.

| |
|-----|
| / 4 |
|-----|

Symbol für Äquivalenzgatter:



w ———

x ———

y ———

z ———

———— f2

Aufgabe 7 **7 Punkte**

2018-H-07

Huffman

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

Gegeben sei der folgende Text aus 25 Zeichen:

| |
|---------------------------|
| MENS_SANA_IN_CAMPARI_SODA |
|---------------------------|

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.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| M | I | O | E | D | C | _ | S | N | R | P | A |
| | | | | | | | | | | | |

| Zeichen | Kodierung |
|---------|-----------|
| M | |
| I | |
| O | |
| E | |
| D | |
| C | |
| _ | |
| S | |
| N | |
| R | |
| P | |
| A | |

Aufgabe 8**6 Punkte****2018-H-08****Kodierung – Fano-Bedingung**

- (a) Seien A und B Alphabete, $c : A \rightarrow B^3$ eine Blockkodierung und c^* die natürliche Fortsetzung von c . Zeigen Sie, dass

$$c \text{ injektiv} \iff c^* \text{ injektiv.}$$

Hinweis: Die natürliche Fortsetzung meint die Erweiterung eines Codes von einzelnen Zeichen zu ganzen Codewörtern. Injektivität von c^* bedeutet, dass jedes Codewort eindeutig dekodiert werden kann.

- (b) Seien $A := \{x, y, z\}$ und $B := \{0, 1\}$ Alphabete. Definieren Sie eine **injektive** Kodierung

$$c : A \rightarrow B^*$$

sodass c die Fano-Bedingung **nicht** erfüllt, aber ihre natürliche Fortsetzung c^* dennoch **injektiv** ist. Begründen Sie Ihre Antwort kurz.

Hinweis: Gesucht ist also eine Kodierung, die die Fano-Bedingung nicht erfüllt, aber trotzdem eindeutig dekodiert werden kann.

$$c(x) =$$

$$c(y) =$$

$$c(z) =$$

Aufgabe 9**6 Punkte****2018-H-09****Rechnerarchitektur**

/ 6

- (a) In der Vorlesung wurde das klassische Konzept eines Universalrechners nach Burks, Goldstine und von Neumann vorgestellt. Kennzeichnen Sie im gegebenen Schaubild die einzelnen Komponenten dieses Konzepts und zeichnen Sie den Datenfluss ein. Beschreiben Sie **kurz** die Funktion jeder Komponente.

/ 4



- (b) Das Konzept des Befehlsphasen-Pipelining führt zu einer erheblichen Beschleunigung der Programmausführung. Es können dabei aber auch Probleme auftreten. Nennen Sie zwei und geben Sie je einen Ansatz zu deren Lösung.

/ 2

Aufgabe 10**7 Punkte****2018-H-10****Programmiersprachen**

/ 7

In der Vorlesung haben wir gesehen, dass alle gängigen Programmiersprachen dieselbe Ausdrucksmächtigkeit haben, also einander simulieren können. Im Folgenden seien gegeben:

- eine **GOTO-Sprache** (das sei eine übliche Assemblersprache, die Sprünge in Abhängigkeit des aktuellen Akkumulator-Werts erlaubt: `JNZ label`) und
- eine **WHILE-Sprache** (das sei eine höhere Sprache wie Java, die allgemeine WHILE-Schleifen unterstützt: `while (b) { * do something * }`).

(a) Gegeben sei ein Programmausschnitt in der GOTO-Sprache:

/ 5

```
label:  LOAD R1
        ADD R2
        STORE R1
        LOAD R2
        SUB 1
        STORE R2
        JNZ label // „JUMP NOT ZERO“
```

Simulieren Sie das Programm durch die WHILE-Sprache, d. h. geben Sie ein äquivalentes WHILE-Programm an.

Hinweise:

- Nutzen Sie möglichst Java-Notation oder einen verständlichen Pseudocode.
- Der Akkumulator sei über ACC erreichbar, die Register über ihre Namen Ri.

(b) Was müsste im gegebenen Programmausschnitt verändert werden, damit am Ende die Fakultät $R2!$ in R1 steht?

Hinweis: Es sind **an zwei Stellen** Veränderungen nötig. Tragen Sie diese einfach im gegebenen Code ein.

/ 2

Aufgabe 11 **6 Punkte**

2018-H-11

Betriebssysteme

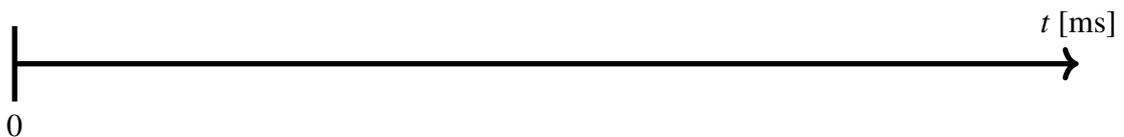
/ 6

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

| Prozesse | CPU-Zeit in ms |
|----------|----------------|
| P1 | 35 |
| P2 | 25 |
| P3 | 20 |
| P4 | 5 |

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) Nehmen Sie nun an, Prozess P1 wird **blockiert**, nachdem er für 5 ms **aktiv** war. Nachdem weitere 40 ms vergangen sind, wird der Zustand von P1 wieder auf **bereit** gesetzt. Teilen Sie den Prozessen aus Aufgabenteil (a) unter dieser Voraussetzung Rechenzeit gemäß dem Round-Robin Verfahren zu mit einer Zeitscheibenlänge $Z = 10$ ms. Veranschaulichen Sie Ihr Ergebnis auf dem gegebenen Zeitstrahl.

/ 3

