

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

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

Matr.-Nr.: \_\_\_\_\_ Semester: \_\_\_\_\_ (SS 2013)

Ich bestätige, dass ich die folgenden Angaben gelesen und mich von der Vollständigkeit dieses Klausurexemplars überzeugt habe (Seiten 1-18).

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

# Aufgabenübersicht

1) <b>Minimierung endlicher Automaten</b> (9 Punkte) . . . . .	2
2) <b>Kontextfreie Grammatiken, Pumping-Lemma</b> (10 Punkte) . .	4
3) <b>Kontextfreie Grammatiken, Chomsky-Normalform</b> (6 Punkte)	5
4) <b>Turingmaschine</b> (8 Punkte) . . . . .	7
5) <b>Komplexitätstheorie</b> (9 Punkte) . . . . .	8
6) <b>Schaltnetz und CMOS</b> (9 Punkte) . . . . .	9
7) <b>Binary-Decision-Diagram (BDD)</b> (8 Punkte) . . . . .	11
8) <b>Zahlendarstellung</b> (10 Punkte) . . . . .	13
9) <b>Programmierung</b> (6 Punkte) . . . . .	14
10) <b>Pipelining</b> (8 Punkte) . . . . .	16
11) <b>Dateiverwaltung und -organisation</b> (7 Punkte) . . . . .	18

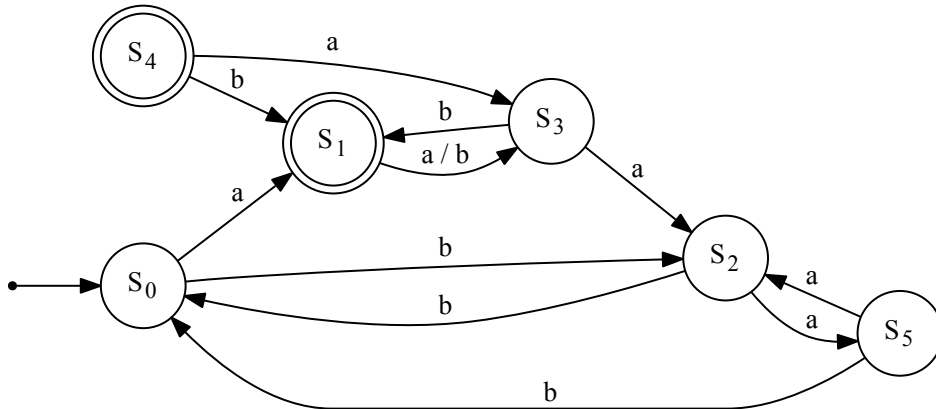
**Aufgabe 1** **9 Punkte**

2013-N-01

**Minimierung endlicher Automaten**

/ 9

Gegeben sei der deterministische endliche Automat  $A = (E, S, \delta, s_0, F)$ . Durch das abgebildete Zustandsdiagramm sei  $\delta$  definiert.



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

/ 5

**Lösung:**

- Da der Zustand  $s_4$  nicht erreichbar ist, kann dieser direkt weggelassen werden.
- Übergangstabelle:

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

- Dreieckstabelle:

$s_1$	$\times_0$			
$s_2$	$\times_1$	$\times_0$		
$s_3$	$\times_1$	$\times_0$	$\times_1$	
$s_5$	$\times_1$	$\times_0$	–	$\times_1$
	$s_0$	$s_1$	$s_2$	$s_3$

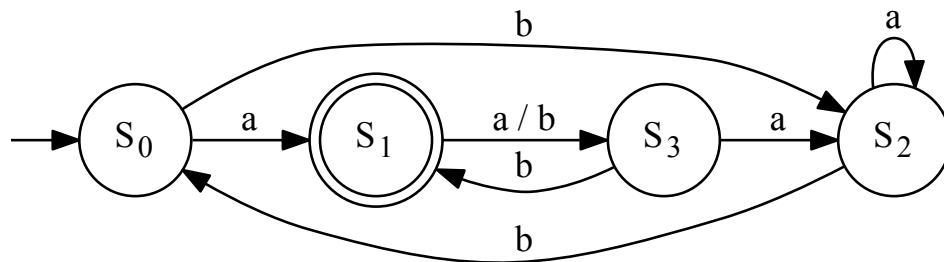
- neue Übergangstabelle

	<i>a</i>	<i>b</i>
<i>s</i> <sub>0</sub>	<i>s</i> <sub>1</sub>	<i>s</i> <sub>2</sub>
<i>s</i> <sub>1</sub>	<i>s</i> <sub>3</sub>	<i>s</i> <sub>3</sub>
<i>s</i> <sub>2</sub>	<i>s</i> <sub>2</sub>	<i>s</i> <sub>0</sub>
<i>s</i> <sub>3</sub>	<i>s</i> <sub>2</sub>	<i>s</i> <sub>1</sub>

- Der minimierte Automat *A'* lautet somit wie folgt:

$$A' = (\{a, b\}, \{s_0, s_1, s_2, s_3\}, \delta, s_0, \{s_1\})$$

mit  $\delta$



- (b) Geben Sie die Mengen aller zueinander *k*-äquivalenten Zustände für  $k \in \{1, 2\}$  und die Mengen äquivalenter Zustände des vereinfachten (nicht minimierten) endlichen Automaten *A* an. Verwenden Sie hierfür die nachfolgende Tabelle. Geben Sie auch einelementige Mengen an.

/ 3

**Lösung:**

0-Äquivalenz	$\{s_0, s_2, s_3, s_5\}, \{s_1\}$
1-Äquivalenz	$\{s_0\}, \{s_1\}, \{s_3\}, \{s_2, s_5\}$
<i>k</i> -Äquivalenz	$\{s_0\}, \{s_1\}, \{s_3\}, \{s_2, s_5\}$

- (c) Inwiefern ist es ein schwierigeres Problem, einen nichtdeterministischen endlichen Automaten zu minimieren im Vergleich zu einem deterministischen?

/ 1

**Lösung:** Die Minimierung nichtdeterministischer endlicher Automaten ist alles andere als trivial. Man kann dabei Zustände **und** Kanten minimieren. Allein die Minimierung der Zustände ist ein schwieriges Problem.

**Aufgabe 2**

**10 Punkte**

2013-N-02

**Kontextfreie Grammatiken, Pumping-Lemma**

/ 10

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

$$P = \{S \rightarrow ASB \mid \lambda, \\ A \rightarrow a, \\ B \rightarrow CBC \mid b, \\ C \rightarrow c\}$$

- (a) Geben Sie die Sprache  $L(G)$  **formal oder umgangssprachlich** korrekt und präzise an. Leiten Sie zusätzlich **drei verschiedene Wörter**  $w_1, \dots, w_3 \in L(G)$  mit der Grammatik ab und geben Sie die vollständigen Ableitungen explizit an.

/ 4

**Lösung:**

$$L(G) = a^n \cdot \prod_{i=1}^n c^{k_i} b c^{k_i}, n, k_i \in \mathbb{N}_0$$

also gilt insbesondere

$$a^n b^n \subseteq L(G) \text{ und } w \in L(G) \Rightarrow |w|_a = |w|_b$$

- (b) Zeigen Sie mithilfe des Pumping-Lemmas für EA-Sprachen, dass  $L(G)$  nicht von einem endlichen Automaten akzeptiert werden kann.

/ 5

**Lösung:** Wähle  $a^n b^n$  als Beispielwort und verfare wie bei dieser Sprache.

- (c) Zu welchen Sprachklassen der Chomsky-Hierarchie gehört die Sprache  $L(G)$  also? Geben Sie alle zutreffenden an.

/ 1

**Lösung:** Typen 0, 1, 2.

**Aufgabe 3**

**6 Punkte**

2013-N-03

**Kontextfreie Grammatiken, Chomsky-Normalform**

/ 6
-----

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

$$\begin{aligned}
 P = \{ & S \rightarrow ASB \mid \lambda, \\
 & A \rightarrow a, \\
 & B \rightarrow CBC \mid b, \\
 & C \rightarrow c \}
 \end{aligned}$$

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

**Hinweise:**

- Benennen Sie die Schritte, die zu dem Algorithmus gehören (oder geben Sie eine kurze Beschreibung an).
- Geben Sie  $G_{CNF}$  und alle **unterschiedlichen** Zwischengrammatiken vollständig an.

/ 4
-----

**Lösung:**

- (1)  $\lambda$ -frei machen
- (2) reine Umbenennungen entfernen (fällt weg)
- (3) auf der rechten Seite nur 1 Terminal oder beliebig viele Nonterminale (fällt weg)
- (4) genau 2 Nonterminale

$$\begin{aligned}
 G_{CNF} = (\{ & S', S, A, B, C, G, H \}, \{a, b, c\}, P, S') \\
 P = \{ & S' \rightarrow S \mid \lambda, \\
 & S \rightarrow AG \mid AB, \\
 & A \rightarrow a, \\
 & B \rightarrow CH \mid b, \\
 & C \rightarrow c, \\
 & G \rightarrow SB, \\
 & H \rightarrow BC \}
 \end{aligned}$$

(b) Erklären Sie in kurzen Worten, welchen Zweck der Cocke-Younger-Kasami-Algorithmus erfüllt und nach welcher Grundidee er funktioniert. Weshalb ist es notwendig, dass die zugrundeliegende Grammatik in CNF vorliegt?

**Lösung:**

- Überprüfen, ob ein Wort aus einer vorgegebenen kontextfreie grammatik erzeugt werden kann.
- Grundidee: Konzept der dynamischen Programmierung, findet nacheinander für Teilwörter der Wortes  $w$  der Länge  $m = 1, 2, \dots$  heraus, über welche Nonterminalsymbole sie erzeugt werden können.
- CNF: pro Ableitung darf nur ein weiteres Symbol dazu kommen, demnach muss die Grammatik in CNF vorliegen.

**Aufgabe 4** **8 Punkte**

2013-N-04

**Turingmaschine**

/ 8
-----

Gegeben sei das Alphabet  $E = \{1\}$ . Im Folgenden sei eine Zahl  $k \in \mathbb{N}_0$  unär als  $1^k \in E^*$ , also als Folge von  $k$  Einsen dargestellt. Beispielsweise entspricht die Zahl 5 demnach dem Wort 11111.

Geben Sie eine Turingmaschine  $T = (E, B, S, \delta, s_0, F)$  an, die folgende Funktion berechnet:

$$f : E^* \rightarrow E^* : f(w) = w \bmod 3$$

Bei Eingabe des Wortes  $w = 11111111$  soll dieses beispielsweise in  $f(w) = 11$  konvertiert werden.

**Hinweise:**

- Achten Sie auf eine korrekte Verarbeitung aller möglichen Eingaben, auch des leeren Wortes.
- Der Schreib-/Lesekopf steht zu Beginn über dem linken Zeichen des Eingabewortes.
- Er darf sich am Ende an einer beliebigen Stelle des Bandes befinden.
- Geben Sie  $T$  vollständig an.

**Lösung:**

$$T = (\{1\}, \{1, \star\}, \{s_0, s_1, s_2, s_3, s_4, s_5\}, \delta, s_0, \{s_5\})$$

$\delta$	1	★		
$s_0$	$(s_1, 1, R)$	$(s_5, \star, L)$		
$s_1$	$(s_2, 1, R)$	$(s_4, \star, L)$		
$s_2$	$(s_0, 1, R)$	$(s_3, \star, L)$		
$s_3$	$(s_4, 1, L)$			
$s_4$	$(s_5, 1, L)$			
$s_5$	$(s_5, \star, L)$			



**Aufgabe 5****9 Punkte****2013-N-05****Komplexitätstheorie**

/ 9

Für ein Alphabet  $E$  und eine Komplexitätsklasse  $K \subseteq \varphi(E^*)$  sei durch

$$Co-K = \{L \mid E^* \setminus L \in K\}$$

die Komplementklasse gegeben, also die Menge aller Sprachen  $L$ , deren Komplementsprache  $\overline{L} = E^* \setminus L$  in  $K$  liegt.

Anders gesagt enthält  $Co-K$  die Probleme, für die bei jeder Instanz die Antwort gegensätzlich ist ( $true \Leftrightarrow false$ ) als bei einem Problem aus  $K$ .

- (a) Argumentieren Sie, dass  $P = Co-P$  gilt.

**Hinweis:** Überlegen Sie, wie man aus einem gegebenen Polynomialzeitalgorithmus für ein Problem  $L \in P$  einen Polynomialzeitalgorithmus für  $\overline{L}$  erzeugen könnte.

/ 2

- (b) Bis heute ist es ein offenes Problem, ob  $NP = Co-NP$  gilt. Warum ist diese Frage offenbar so viel schwieriger zu beantworten als die deterministische Variante aus (a)?

**Hinweis:** Betrachten Sie das unterschiedliche Akzeptanzverhalten deterministischer und nichtdeterministischer Turingmaschinen.

/ 4

- (c) Beschreiben Sie umgangssprachlich präzise, wie die Instanzen des Komplements  $\overline{SAT}$  von  $SAT$ , dem Erfüllbarkeitsproblem der Aussagenlogik, beschaffen sind.

/ 3

**Lösung:** Das sind aussagenlogische Formeln, für die keine Belegung der Variablen existiert, sodass sie wahr werden.

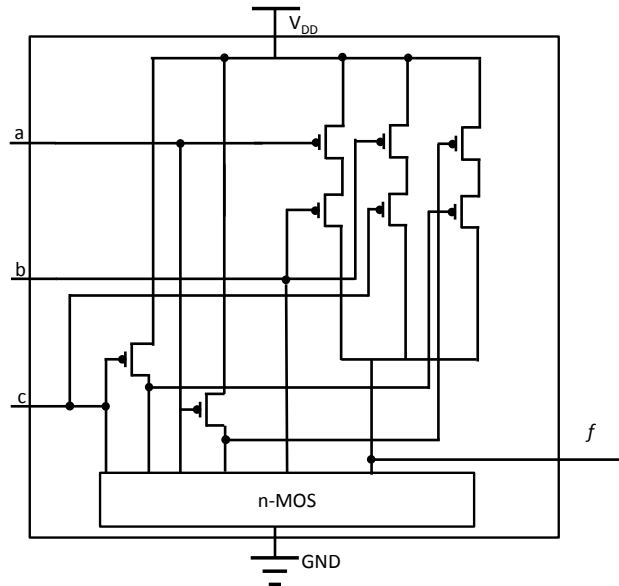
**Aufgabe 6** **9 Punkte**

2013-N-06

Schaltnetz und CMOS

/ 9

Die Boolesche Funktion  $f : \mathbb{B}^3 \rightarrow \mathbb{B}$  sei durch folgende CMOS-Schaltung gegeben.



(a) Geben Sie die Funktion  $f$  als Booleschen Term an.

/ 4

**Lösung:**  $f = (\neg a \wedge \neg b) \vee (\neg b \wedge \neg c) \vee (a \wedge c)$

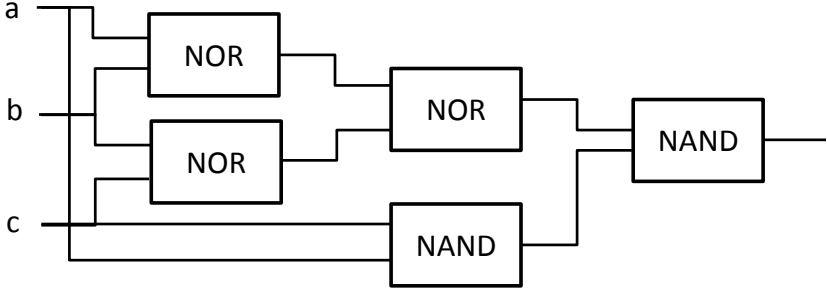
(b) Zeichnen Sie für die Funktion  $f$  ein Schaltnetz. Verwenden Sie hierbei nur die 2-elementigen Gatter NOR und NAND.

/ 5

**Lösung:**

$$\begin{aligned}
 f &= (\neg a \wedge \neg b) \vee (\neg b \wedge \neg c) \vee (a \wedge c) \\
 &= \neg\{\neg[(\neg a \wedge \neg b) \vee (\neg b \wedge \neg c)] \wedge \neg(a \wedge c)\} \\
 &= \neg[(\neg a \wedge \neg b) \vee (\neg b \wedge \neg c)] \text{ NAND } \neg(a \wedge c) \\
 &= [(\neg a \wedge \neg b) \text{ NOR } (\neg b \wedge \neg c)] \text{ NAND } (a \text{ NAND } c) \\
 &= [\neg(a \vee b) \text{ NOR } \neg(b \vee c)] \text{ NAND } (a \text{ NAND } c) \\
 &= [(a \text{ NOR } b) \text{ NOR } (b \text{ NOR } c)] \text{ NAND } (a \text{ NAND } c)
 \end{aligned}$$

Somit ergibt sich folgendes Schaltnetz.



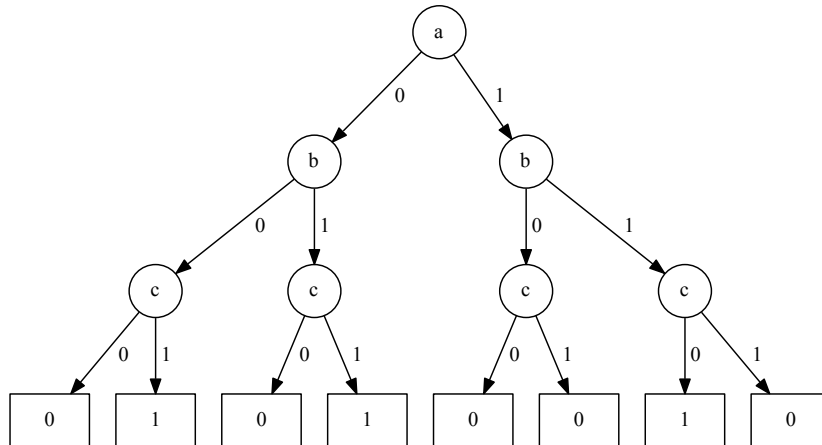
**Aufgabe 7** **8 Punkte**

2013-N-07

**Binary-Decision-Diagram (BDD)**

/ 8

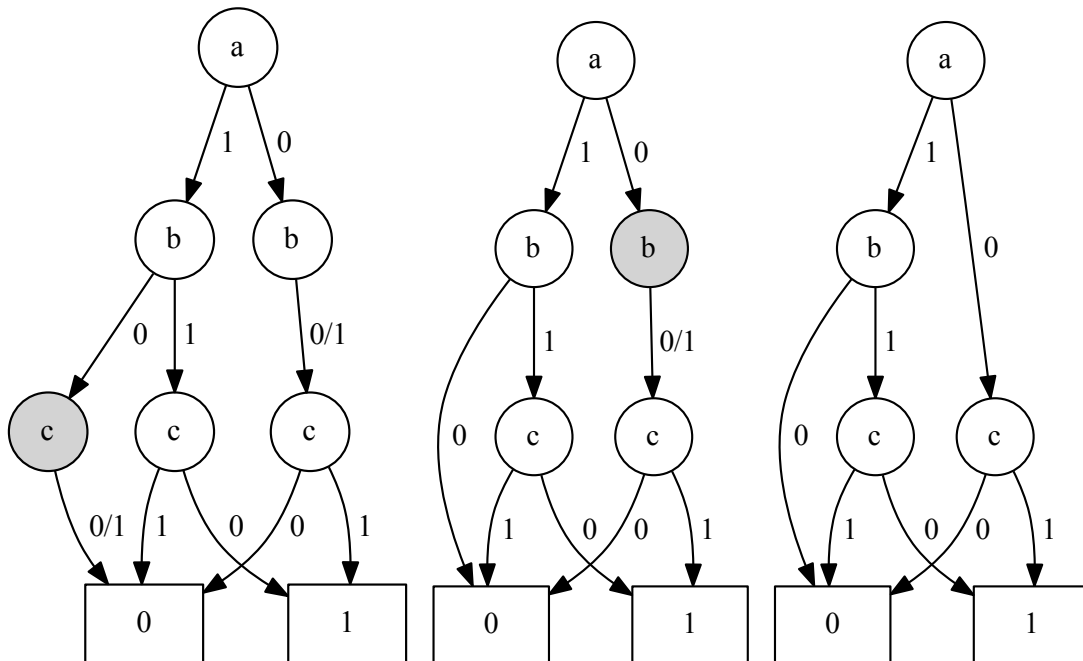
Gegeben sei die durch den abgebildeten Baum definierte Boolesche Funktion  $f : \mathbb{B}^3 \rightarrow \mathbb{B}$ .



(a) Erzeugen Sie das zu  $f$  gehörende BDD.

/ 6

**Lösung:**



(b) Geben Sie die Funktion  $f$  als Booleschen Term an.

/ 2

**Lösung:**  $f(a, b, c) = abc' + a'c$

**Aufgabe 8**

**10 Punkte**

**2013-N-08**

**Zahlendarstellung**

/ 10

Gegeben sei das folgende 32-Bitwort:

1011 1001 1001 0001 1110 0110 0010 0000

Im Folgenden ist es ausreichend, wenn Sie als Ergebnis nicht vollständig ausgerechnete Terme angeben.

- (a) Geben Sie den Dezimalwert des Wortes an, wenn man es als **ganze Zahl in Zweikomplementdarstellung** interpretiert.

/ 3

**Lösung:**

- (b) Geben Sie den Dezimalwert des Wortes an, wenn man es als **Fixpunktzahl mit Dezimalpunkt genau in der Mitte** interpretiert.

/ 2

**Lösung:**

- (c) Geben Sie den Dezimalwert des Wortes an, wenn man es als **Gleitpunktzahl nach IEEE-754** interpretiert.

/ 5

**Lösung:**  $-1 \cdot 2^{2^0+2^1+2^4+2^5+2^6-127} \cdot (1 + 2^{-3} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-13} + 2^{-14} + 2^{-18}) \approx -2^{-12} \cdot 1.1398354 \approx -2.7828012 \cdot 10^{-4}$

**Aufgabe 9**

**6 Punkte**

2013-N-09

Programmierung

/ 6

Gegeben sei das folgende Assemblerprogramm:

	STORE	#1 R2	
	STORE	#1 R4	
LOOP	LOAD	R1	Beginn der for-Schleife
	JUMPZERO	FERTIG	Abbruch der for-Schleife falls $n = 0$
	SUBTRACT	R1 #1 R1	Wir zählen den Wert $n$ herunter
	ADD	R2 R3 R4	Beginn des Blocks in der for-Schleife
	STORE	R2 R3	
	STORE	R4 R2	
	JUMP	LOOP	
FERTIG	HALT		Das Programm ist beendet

**Hinweise:**

- Für unmittelbare Adressierung wird das Präfix # genutzt, kein Präfix steht für direkte Adressierung.
- Der letzte von mehreren Operanden bezeichnet jeweils die Zieladresse.
- Sie können davon ausgehen, dass alle Register, außer R1, mit 0 initialisiert sind.

Das Register R1 ist zu Beginn mit einem Wert  $n \in \mathbb{N}_0$  belegt. Der Wert  $f(n)$  sei definiert als Ergebnis der Rechnung, das nach Terminierung des Programms in Register R4 steht.

(a) Geben Sie für  $n \in \{0, \dots, 4\}$  jeweils den Wert  $f(n)$  an.

/ 5

(b) Wie hängt  $f(n)$  für  $n > 1$  von den vorangegangenen Werten  $f(k)$  für  $k < n$  ab?

/ 1

**Lösung:** Es handelt sich um die Fibonacci-Zahlen (beginnend mit 1, 1, 2, usw.; die 0 am Beginn fällt weg).

Zugehöriges Java-Programm:

```
int fib(int n) {
    if (n == 0) return 0;
    if (n == 1) return 1;

    int f'' = 0;
    int f' = 1;
```

```
int f;  
  
for (int i = 2; i <= n; i++) {  
    f = f'' + f';  
    f'' = f';  
    f' = f;  
}  
  
return f;  
}
```

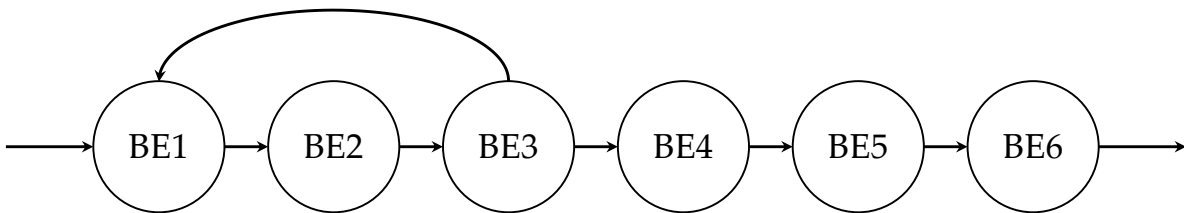


<b>Aufgabe 10</b>	<b>8 Punkte</b>
<b>2013-N-10</b>	<b>Pipelining</b>
	/ 8

Gegeben sei eine Kuh, die bekanntermaßen vier Mägen besitzt und (stark vereinfacht) über die folgenden Bearbeitungseinheiten (BE) für zugeführte Nahrung verfüge:

- (BE1) **Mundhöhle**: Kauen und Schlucken.
- (BE2) **Pansen**: Durchmischung, Verdauung von Kohlenhydraten.
- (BE3) **Netzmagen**: Filterung – kleine Teile gehen weiter, zu grobe Teile werden zur Mundhöhle **zurückgeschickt**.
- (BE4) **Blättermagen**: Eindicken durch Auspressen des Wassers.
- (BE5) **Labmagen**: Verdauung von Eiweißen und Fetten.
- (BE6) **Darm**: Ausscheidung unverdaulicher Teile.

Es ergeben sich also folgende Hauptströme der Nahrung:



- (a) Inwiefern sind die Abläufe im Verdauungstrakt einer Kuh vergleichbar mit dem Konzept des Pipelinings? Wo liegen wesentliche Unterschiede? Nennen Sie jeweils mindestens zwei Kriterien.

/ 3

**Lösung:**

Vergleichbarkeit: Parallelität, sequentielle Ausführung

Unterschiede: Nahrung kann sich aufteilen, Zurückschwappen der Nahrung

Im Folgenden nehmen wir zur weiteren Vereinfachung an, dass Nahrung aus atomaren Quanten bestehe, die einzeln von einer BE verarbeitet werden und nicht weiter teilbar sind.

- (b) Welche Auswirkungen hat die Möglichkeit, Nahrung von BE3 nach BE1 zurückzuschicken, auf den Ablauf? Nennen Sie hierfür mindestens zwei Aspekte.

/ 3

**Lösung:** Anhalten der weiteren Nahrungsaufnahme notwendig, „Pipeline“ kann leerlaufen, Endlosschleife möglich, etc.

- (c) Könnte durch die Methode der „spekulativen Programmausführung“ eine Verbesserung dieser Auswirkungen erzielt werden? Begründen Sie.

/ 2

**Lösung:**

**Aufgabe 11**

**7 Punkte**

**2013-N-11**

**Dateiverwaltung und -organisation**

/ 7

Dateiverwaltungssysteme sind bei der Vielzahl der heute in Rechnersystemen zu verarbeitenden Informationen von großer Bedeutung.

- (a) Erläutern Sie kurz die Begriffe **Satz**, **Satztyp** und **Datei** in diesem Zusammenhang.

/ 2

**Lösung:**

**Satz:** Zusammenfassung logisch zusammengehörender Daten, die dasselbe Objekt oder denselben Sachverhalt betreffen, zu einer logischen Einheit.

**Satztyp:** Beschreibung des formalen Aufbaus eines Satzes.

**Datei:** Zusammenfassung von Sätzen meist eines Satztyps.

- (b) Welches Konstrukt wird in der Dateiverwaltung verwendet, um einen Satz eindeutig identifizieren zu können? Welche zwei Typen werden hierbei unterschieden? Grenzen Sie diese voneinander ab.

/ 3

**Lösung:** Konstrukt: Schlüssel

- Primärschlüssel: von vorherein ausgezeichnet und identifizierend
- Sekundärschlüssel: beliebig, muss nicht identifizierend sein

- (c) Bei der physischen Speicherung von Sätzen können diese auf verschiedene Arten angeordnet werden. Nennen Sie zwei Primärorganisationsformen und geben Sie für jede der zwei ein Beispiel an.

/ 2

**Lösung:**

- sequentielle Primärorganisation: unsortiert oder seriell
- gestreute Primärorganisation: Hash-Organisation oder index-sequentielle Organisation