Karlsruher Institut für Technologie

aifb
Institut für Angewandte Informatik
und Formale Beschreibungsverfahren

21.07.2014

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

Name: ______________________ Vorname: ______________________

Matr.-Nr.: ______________________ Semester: __________ (SS 2014)

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 **9 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 | - | - | - | - | - | - | - | gesamt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (13) | (8) | (10) | (12) | (10) | (10) | (12) | (8) | (7) | | | | | | | | (90) |

# Aufgabenübersicht

## Aufgabe 1 — 13 Punkte

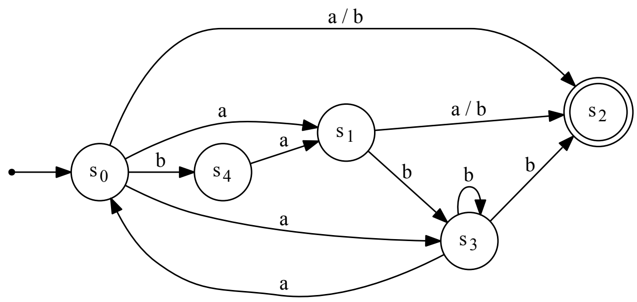
**2014-N-01** **Endliche Automaten**

/ 13

Gegeben sei der folgende nichtdeterministische endliche Automat:

$$A = (\{a, b\}, \{s_0, s_1, s_2, s_3, s_4\}, \delta, s_0, \{s_2\})$$

$\delta$:




(a) Wandeln Sie $A$ durch den Algorithmus aus der Vorlesung in einen äquivalenten deterministischen endlichen Automaten $A'$ um.

/ 8

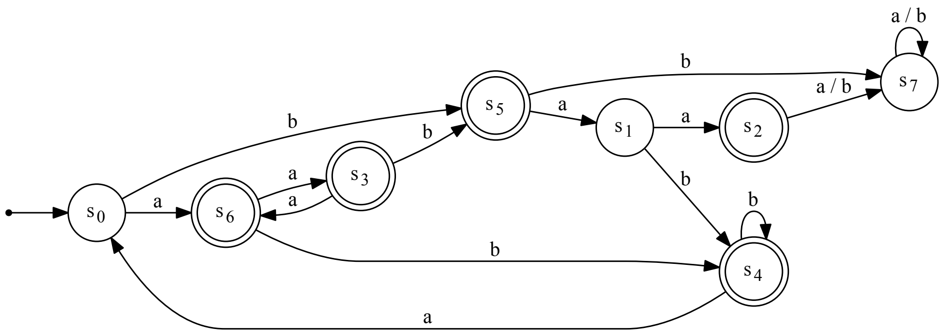
**Hinweis:** Geben Sie $A'$ vollständig an. Sie müssen jedoch kein Zustandsüberführungsdiagramm zeichnen, eine korrekte Angabe des Tupels und der Tabelle reichen aus.

**Lösung:**

| | $a$ | $b$ |
|---|---|---|
| $\{s_0\} \hat{=} s_0$ | $\{s_1, s_2, s_3\} \hat{=} s_6$ | $\{s_2, s_4\} \hat{=} s_5$ |
| $\{s_1\} \hat{=} s_1$ | $\{s_2\} \hat{=} s_2$ | $\{s_2, s_3\} \hat{=} s_4$ |
| $\{s_2\} \hat{=} s_2$ | $\emptyset \hat{=} s_7$ | $\emptyset \hat{=} s_7$ |
| $\{s_0, s_2\} \hat{=} s_3$ | $\{s_1, s_2, s_3\} \hat{=} s_6$ | $\{s_2, s_4\} \hat{=} s_5$ |
| $\{s_2, s_3\} \hat{=} s_4$ | $\{s_0\} \hat{=} s_0$ | $\{s_2, s_3\} \hat{=} s_4$ |
| $\{s_2, s_4\} \hat{=} s_5$ | $\{s_1\} \hat{=} s_1$ | $\emptyset \hat{=} s_7$ |
| $\{s_1, s_2, s_3\} \hat{=} s_6$ | $\{s_0, s_2\} \hat{=} s_3$ | $\{s_2, s_3\} \hat{=} s_4$ |
| $\emptyset \hat{=} s_7$ | $\emptyset \hat{=} s_7$ | $\emptyset \hat{=} s_7$ |

$$A' = (\{a, b\}, \{s_0, \ldots, s_7\}, \delta', s_0, \{s_1, s_2, s_3, s_5, s_6\})$$

$\delta'$:




(b) Je nach Benennung der Zustände kann sich für den deterministischen endlichen Automaten $A'$ folgende Minimierungstabelle ergeben:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $s_1$ | $\times_2$ | | | | | | |
| $s_2$ | $\times_0$ | $\times_0$ | | | | | |
| $s_3$ | $\times_0$ | $\times_0$ | $\times_1$ | | | | |
| $s_4$ | $\times_0$ | $\times_0$ | $\times_1$ | $\times_1$ | | | |
| $s_5$ | $\times_0$ | $\times_0$ | $\times_2$ | $\times_1$ | $\times_1$ | | |
| $s_6$ | $\times_0$ | $\times_0$ | $\times_1$ | $\times_2$ | $\times_1$ | $\times_1$ | |
| $s_7$ | $\times_1$ | $\times_1$ | $\times_0$ | $\times_0$ | $\times_0$ | $\times_0$ | $\times_0$ |
| | $s_0$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ |

Wie viele Zustände hat demnach ein äquivalenter minimaler endlicher Automat $A''$?

/ 1

**Lösung:** 8 Zustände (denn kein Paar der anfänglichen Zustände $s_0, \ldots, s_7$ ist äquivalent).

(c) Geben Sie die Mengen zueinander $k$-äquivalenter Zustände von $A'$ für $k = 0, 1, 2, \ldots$ an. Gehen Sie von der Minimierungstabelle in Aufgabenteil (b) aus, **nicht** von Ihrem Automaten in Aufgabenteil (a).

/ 4

**Hinweis:** Einelementige Mengen müssen Sie nicht angeben.

| $k$ | Mengen zueinander $k$-äquivalenter Zustände |
|---|---|
| 0 | $\{s_0, s_1, s_7\}, \{s_2, s_3, s_4, s_5, s_6\}$ |
| 1 | $\{s_0, s_1\}, \{s_7\}, \{s_2, s_5\}, \{s_4\}, \{s_3, s_6\}$ |
| 2 | $\{s_0\}, \{s_4\}, \{s_2\}, \{s_6\}, \{s_3\}, \{s_5\}, \{s_7\}, \{s_1\}$ |
| 3 | Keine Änderung |
| 4 | ... |
| 5 | ... |

## Aufgabe 2 — 8 Punkte

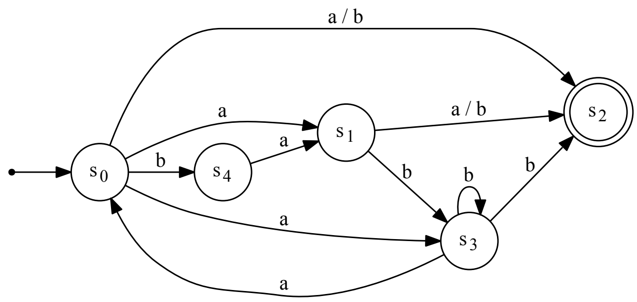
**2014-N-02** — **Rechtslineare Grammatiken**

/ 8

Gegeben sei wieder der nichtdeterministische endliche Automat:

$$A = (\{a, b\}, \{s_0, s_1, s_2, s_3, s_4\}, \delta, s_0, \{s_2\})$$

$\delta$:




Geben Sie eine rechtslineare Grammatik $G$ an mit $L(G) = L(A)$. Geben Sie die Grammatik vollständig an.

**Hinweis:** Es könnte hilfreich sein, die Nonterminale ähnlich wie die Zustände des Automaten zu benennen, also etwa $N = \{S_0, \ldots, S_4\}$.

**Lösung:**

$$\begin{aligned} G &= (\{S_0, \ldots, S_4\}, \{a, b\}, P, S_0) \\ P &= \{S_0 \to aS_1 \mid aS_2 \mid bS_2 \mid aS_3 \mid bS_4, \\ &\quad S_1 \to aS_2 \mid bS_2 \mid bS_3, \\ &\quad S_2 \to \lambda, \\ &\quad S_3 \to aS_0 \mid bS_2 \mid bS_3, \\ &\quad S_4 \to aS_1\} \end{aligned}$$

| **Aufgabe 3** | **10 Punkte** |
|---|---|

**2014-N-03** **Pumping-Lemma für EA-Sprachen**

/ 10

Gegeben sei die Sprache $L \subset \{1\}^\star$ durch

$$111 \in L$$

und

$$w \in L \Rightarrow w^{|w|} \in L$$

Es gilt also:

$$111, \underbrace{111\,111\,111}_{3\cdot 3=9}, \underbrace{1\cdots 1}_{9\cdot 9=81}, \ldots \in L$$

Zeigen Sie mithilfe des Pumping-Lemmas für EA-Sprachen, dass $L \notin L_3$.

**Lösung:** Angenommen, $L$ wäre vom Typ 3. Dann existiert ein $n \in \mathbb{N}$, sodass jedes Wort $w \in L$ mit $|w| \geq n$ aufgeteilt werden kann in $w = xyz$ mit

(a) $|xy| \leq n$,

(b) $y \neq \lambda$,

(c) $\forall i \in \mathbb{N} : xy^i z \in L$.

Betrachte $i = 2$: Es gilt offensichtlich $|w| < |xy^2z| \leq 2|xyz| = 2|w|$. Da das kleinste Wort in $L$ drei Zeichen hat, gilt für beliebige Wörter $w \neq w' \in L$, dass $w$ und $w'$ sich in der Länge mindestens um den Faktor drei unterscheiden. Also können zwei Wörter, deren Längen sich um höchstens den Faktor zwei unterscheiden ($w$ und $xy^2z$), nicht beide gleichzeitig in $L$ sein. Damit gilt $xy^2z \notin L$, wir sind zu einem Widerspruch gekommen und müssen die anfängliche Vermutung aufgeben, dass $L$ vom Typ 3 ist.

| **Aufgabe 4** | **12 Punkte** |
|---|---|

**2014-N-04** **Turingmaschine (Busy Beaver)**

/ 12

*Diese Aufgabe kann zeitaufwändig sein, es könnte sinnvoll sein, sie am Ende zu bearbeiten.*

Geben Sie eine Turingmaschine an, die

- genau fünf Zustände und das Bandalphabet $\{1, \star\}$ hat,
- bei leerer Eingabe **nach endlich vielen Schritten anhält** und
- dabei **möglichst viele Einsen auf das Band schreibt** (es zählt die Bandinschrift am Ende der Rechnung; Lücken sind erlaubt: „1 $\star$ 11“).

Geben Sie die Turingmaschine vollständig an.

**Hinweise:**

- Damit die Turingmaschine nicht in eine Endlosschleife gerät, muss mindestens ein Zustand-Bandsymbol-Paar undefiniert bleiben.
- Die volle Punktzahl erhalten Sie, wenn die Turingmaschine **mindestens 7 Einsen** auf das Band schreibt; für 6 Einsen gibt es noch 10 Punkte.
- Sie erhalten **4 Zusatzpunkte**, wenn bei dieser Aufgabe kein Klausurteilnehmer mehr Einsen schafft als Sie.

**Lösung:** (Exkurs – Busy Beaver wird eine Turingmaschine genannt, die für eine feste Zustandszahl $n$ die größtmögliche Anzahl Einsen auf das zu Beginn leere Arbeitsband schreibt. Ist $b(n)$ die Anzahl dieser Einsen bei $n$ Zuständen, dann ist $b$ eine extrem schnell wachsende Funktion. Sie wächst schneller als jede berechenbare Funktion und ist somit selbst nicht berechenbar. Um ein Gefühl dafür zu bekommen, wie schnell sie wächst, kann man sich statt einer Turingmaschine mit $n$ Zuständen ein Java-Programm mit $n$ Zeichen vorstellen (mit nur einer Klasse und einer main-Methode). Sobald eine gewisse Schwelle $n$ überschritten wurde, hat man genügend Zeichen, um in der main-Methode eine Schleife zu bilden, etwa durch `for (int i = 0; i < 9; i++) System.out.print(1);` jedes weitere Zeichen erlaubt dann beispielsweise, eine 9 zum Schleifen-Abbruchkriterium hinzuzufügen, also die Anzahl an Einsen exponentiell zu erhöhen. Ein weiterer Sprung in der Anzahl möglicher Einsen entsteht, wenn man genügend Zeichen hat, um Schleifen zu schachteln. Und eine extreme Vergrößerung der möglichen Anzahl an Einsen wird durch `while`-Schleifen erzielt. Intuitiv ist klar, dass für größere $n$ die Angabe von $b(n)$ unmöglich ist – man weiß ja nicht, ob eine arbeitende Turingmaschine schon in eine Endlosschleife geraten ist oder nicht. Bei Turingmaschinen ist schon für 5 Zustände kein Busy Beaver bekannt – deshalb wurde in der Aufgabe auch nicht nach einem solchen gefragt. Es ist jedoch bekannt, dass (mit einer kleinen Erweiterung, s. u.) mindestens 4098 Einsen möglich sind!)

Die folgenden beiden Turingmschinen schreiben 6 bzw. 9 Einsen auf das Arbeitsband. Der darauffolgende echte Busy Beaver für 4 Zustände schreibt 13 Einsen auf das Band.

Beispiel 1:

| | 1 | $\star$ |
|---|---|---|
| $s_0$ | | $(s_1, 1, R)$ |
| $s_1$ | | $(s_2, \star, R)$ |
| $s_2$ | | $(s_3, \star, R)$ |
| $s_3$ | | $(s_4, \star, R)$ |
| $s_4$ | $(s_0, 1, L)$ | $(s_4, 1, L)$ |

| Bandinhalt | Zustand |
|---|---|
| $\hat{\star}$ | $[(s_0, \star) \to (s_1, 1)/R]$ |
| $1\hat{\star}$ | $[(s_1, \star) \to (s_2, \star)/R]$ |
| $1 \star \hat{\star}$ | $[(s_2, \star) \to (s_3, \star)/R]$ |
| $1 \star \star\hat{\star}$ | $[(s_3, \star) \to (s_4, \star)/R]$ |
| $1 \star \star \star \hat{\star}$ | $[(s_4, \star) \to (s_4, 1)/L]$ |
| $1 \star \star\hat{\star}1$ | $[(s_4, \star) \to (s_4, 1)/L]$ |
| $1 \star \hat{\star}11$ | $[(s_4, \star) \to (s_4, 1)/L]$ |
| $1\hat{\star}111$ | $[(s_4, \star) \to (s_4, 1)/L]$ |
| $\hat{1}1111$ | $[(s_4, 1) \to (s_0, 1)/L]$ |
| $\hat{\star}11111$ | $[(s_0, \star) \to (s_1, 1)/R]$ |
| $1\hat{1}1111$ | akzeptiert |

Beispiel 2:

| $\delta$ | 1 | $\star$ |
|---|---|---|
| $s_0$ | $(s_4, 1, L)$ | $(s_3, 1, R)$ |
| $s_1$ | $(s_2, 1, L)$ | $(s_1, 1, L)$ |
| $s_2$ | | $(s_4, 1, L)$ |
| $s_3$ | $(s_3, \star, R)$ | $(s_1, \star, R)$ |
| $s_4$ | $(s_4, 1, R)$ | $(s_0, 1, R)$ |

| Bandinhalt | Zustand |
|---|---|
| $\hat{\star}$ | $[(s_0, \star) \to (s_3, 1)/R]$ |
| $1\hat{\star}$ | $[(s_3, \star) \to (s_1, \star)/R]$ |
| $1 \star \hat{\star}$ | $[(s_1, \star) \to (s_1, 1)/L]$ |
| $1\hat{\star}1$ | $[(s_1, \star) \to (s_1, 1)/L]$ |
| $\hat{1}11$ | $[(s_1, 1) \to (s_2, 1)/L]$ |
| $\hat{\star}111$ | $[(s_2, \star) \to (s_4, 1)/L]$ |
| $\hat{\star}1111$ | $[(s_4, \star) \to (s_0, 1)/R]$ |

| | |
|---|---|
| $1\hat{1}111$ | $[(s_0, 1) \to (s_4, 1)/L]$ |
| $\hat{1}1111$ | $[(s_4, 1) \to (s_4, 1)/R]$ |
| $1\hat{1}111$ | $[(s_4, 1) \to (s_4, 1)/R]$ |
| $11\hat{1}11$ | $[(s_4, 1) \to (s_4, 1)/R]$ |
| $111\hat{1}1$ | $[(s_4, 1) \to (s_4, 1)/R]$ |
| $1111\hat{1}$ | $[(s_4, 1) \to (s_4, 1)/R]$ |
| $11111\hat{\star}$ | $[(s_4, \star) \to (s_0, 1)/R]$ |
| $111111\hat{\star}$ | $[(s_0, \star) \to (s_3, 1)/R]$ |
| $1111111\hat{\star}$ | $[(s_3, \star) \to (s_1, \star)/R]$ |
| $1111111 \star \hat{\star}$ | $[(s_1, \star) \to (s_1, 1)/L]$ |
| $1111111\hat{\star}1$ | $[(s_1, \star) \to (s_1, 1)/L]$ |
| $111111\hat{1}11$ | $[(s_1, 1) \to (s_2, 1)/L]$ |
| $11111\hat{1}111$ | akzeptiert |

Echter Busy Beaver für 4 Zustände (hier wird ein zusätzlicher Nur-Halte-Zustand zugelassen, der in der Aufgabenstellung der Einfachheit halber weggelassen wurde):

| | 1 | $\star$ |
|---|---|---|
| $s_0$ | $(s_1, 1, L)$ | $(s_1, 1, R)$ |
| $s_1$ | $(s_2, \star, L)$ | $(s_0, 1, L)$ |
| $s_2$ | $(s_3, 1, L)$ | $(s_4, 1, R)$ |
| $s_3$ | $(s_0, \star, R)$ | $(s_3, 1, R)$ |
| $s_4$ | | |

| Bandinhalt | Zustand |
|---|---|
| $\hat{\star}$ | $[(s_0, \star) \to (s_1, 1)/R]$ |
| $1\hat{\star}$ | $[(s_1, \star) \to (s_0, 1)/L]$ |
| $\hat{1}1$ | $[(s_0, 1) \to (s_1, 1)/L]$ |
| $\hat{\star}11$ | $[(s_1, \star) \to (s_0, 1)/L]$ |
| $\hat{\star}111$ | $[(s_0, \star) \to (s_1, 1)/R]$ |
| $1\hat{1}11$ | $[(s_1, 1) \to (s_2, \star)/L]$ |
| $\hat{1} \star 11$ | $[(s_2, 1) \to (s_3, 1)/L]$ |
| $\hat{\star}1 \star 11$ | $[(s_3, \star) \to (s_3, 1)/R]$ |
| $1\hat{1} \star 11$ | $[(s_3, 1) \to (s_0, \star)/R]$ |
| $1 \star \hat{\star}11$ | $[(s_0, \star) \to (s_1, 1)/R]$ |
| $1 \star 1\hat{1}1$ | $[(s_1, 1) \to (s_2, \star)/L]$ |
| $1 \star \hat{1} \star 1$ | $[(s_2, 1) \to (s_3, 1)/L]$ |
| $1\hat{\star}1 \star 1$ | $[(s_3, \star) \to (s_3, 1)/R]$ |
| $11\hat{1} \star 1$ | $[(s_3, 1) \to (s_0, \star)/R]$ |
| $11 \star \hat{\star}1$ | $[(s_0, \star) \to (s_1, 1)/R]$ |
| $11 \star 1\hat{1}$ | $[(s_1, 1) \to (s_2, \star)/L]$ |
| $11 \star \hat{1}\star$ | $[(s_2, 1) \to (s_3, 1)/L]$ |
| $11\hat{\star}1\star$ | $[(s_3, \star) \to (s_3, 1)/R]$ |

| | |
|---|---|
| $111\hat{1}\star$ | $[(s_3,1) \rightarrow (s_0,\star)/R]$ |
| $111\star\hat{\star}$ | $[(s_0,\star) \rightarrow (s_1,1)/R]$ |
| $111\star 1\hat{\star}$ | $[(s_1,\star) \rightarrow (s_0,1)/L]$ |
| $111\star\hat{1}1$ | $[(s_0,1) \rightarrow (s_1,1)/L]$ |
| $111\hat{\star}11$ | $[(s_1,\star) \rightarrow (s_0,1)/L]$ |
| $11\hat{1}111$ | $[(s_0,1) \rightarrow (s_1,1)/L]$ |
| $1\hat{1}1111$ | $[(s_1,1) \rightarrow (s_2,\star)/L]$ |
| $\hat{1}\star 1111$ | $[(s_2,1) \rightarrow (s_3,1)/L]$ |
| $\hat{\star}1\star 1111$ | $[(s_3,\star) \rightarrow (s_3,1)/R]$ |
| $1\hat{1}\star 1111$ | $[(s_3,1) \rightarrow (s_0,\star)/R]$ |
| $1\star\hat{\star}1111$ | $[(s_0,\star) \rightarrow (s_1,1)/R]$ |
| $1\star 1\hat{1}111$ | $[(s_1,1) \rightarrow (s_2,\star)/L]$ |
| $1\star\hat{1}\star 111$ | $[(s_2,1) \rightarrow (s_3,1)/L]$ |
| $1\hat{\star}1\star 111$ | $[(s_3,\star) \rightarrow (s_3,1)/R]$ |
| $11\hat{1}\star 111$ | $[(s_3,1) \rightarrow (s_0,\star)/R]$ |
| $11\star\hat{\star}111$ | $[(s_0,\star) \rightarrow (s_1,1)/R]$ |
| $11\star 1\hat{1}11$ | $[(s_1,1) \rightarrow (s_2,\star)/L]$ |
| $11\star\hat{1}\star 11$ | $[(s_2,1) \rightarrow (s_3,1)/L]$ |
| $11\hat{\star}1\star 11$ | $[(s_3,\star) \rightarrow (s_3,1)/R]$ |
| $111\hat{1}\star 11$ | $[(s_3,1) \rightarrow (s_0,\star)/R]$ |
| $111\star\hat{\star}11$ | $[(s_0,\star) \rightarrow (s_1,1)/R]$ |
| $111\star 1\hat{1}1$ | $[(s_1,1) \rightarrow (s_2,\star)/L]$ |
| $111\star\hat{1}\star 1$ | $[(s_2,1) \rightarrow (s_3,1)/L]$ |
| $111\hat{\star}1\star 1$ | $[(s_3,\star) \rightarrow (s_3,1)/R]$ |
| $1111\hat{1}\star 1$ | $[(s_3,1) \rightarrow (s_0,\star)/R]$ |
| $1111\star\hat{\star}1$ | $[(s_0,\star) \rightarrow (s_1,1)/R]$ |
| $1111\star 1\hat{1}$ | $[(s_1,1) \rightarrow (s_2,\star)/L]$ |
| $1111\star\hat{1}\star$ | $[(s_2,1) \rightarrow (s_3,1)/L]$ |
| $1111\hat{\star}1\star$ | $[(s_3,\star) \rightarrow (s_3,1)/R]$ |
| $11111\hat{1}\star$ | $[(s_3,1) \rightarrow (s_0,\star)/R]$ |
| $11111\star\hat{\star}$ | $[(s_0,\star) \rightarrow (s_1,1)/R]$ |
| $11111\star 1\hat{\star}$ | $[(s_1,\star) \rightarrow (s_0,1)/L]$ |
| $11111\star\hat{1}1$ | $[(s_0,1) \rightarrow (s_1,1)/L]$ |
| $11111\hat{\star}11$ | $[(s_1,\star) \rightarrow (s_0,1)/L]$ |
| $1111\hat{1}111$ | $[(s_0,1) \rightarrow (s_1,1)/L]$ |
| $111\hat{1}1111$ | $[(s_1,1) \rightarrow (s_2,\star)/L]$ |
| $11\hat{1}\star 1111$ | $[(s_2,1) \rightarrow (s_3,1)/L]$ |
| $1\hat{1}1\star 1111$ | $[(s_3,1) \rightarrow (s_0,\star)/R]$ |
| $1\star\hat{1}\star 1111$ | $[(s_0,1) \rightarrow (s_1,1)/L]$ |
| $1\hat{\star}1\star 1111$ | $[(s_1,\star) \rightarrow (s_0,1)/L]$ |
| $\hat{1}11\star 1111$ | $[(s_0,1) \rightarrow (s_1,1)/L]$ |
| $\hat{\star}111\star 1111$ | $[(s_1,\star) \rightarrow (s_0,1)/L]$ |
| $\hat{\star}1111\star 1111$ | $[(s_0,\star) \rightarrow (s_1,1)/R]$ |
| $1\hat{1}111\star 1111$ | $[(s_1,1) \rightarrow (s_2,\star)/L]$ |
| $\hat{1}\star 111\star 1111$ | $[(s_2,1) \rightarrow (s_3,1)/L]$ |
| $\hat{\star}1\star 111\star 1111$ | $[(s_3,\star) \rightarrow (s_3,1)/R]$ |

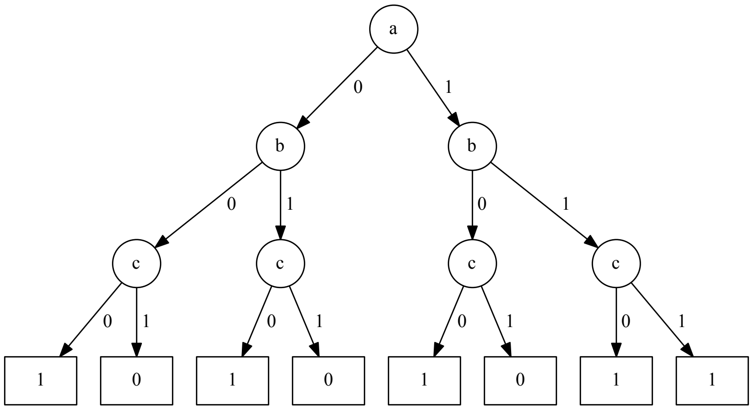
| | |
|---|---|
| $1\hat{1}\star 111 \star 1111$ | $[(s_3,1) \to (s_0,\star)/R]$ |
| $1 \star \hat{\star}111 \star 1111$ | $[(s_0,\star) \to (s_1,1)/R]$ |
| $1 \star 1\hat{1}11 \star 1111$ | $[(s_1,1) \to (s_2,\star)/L]$ |
| $1 \star \hat{1} \star 11 \star 1111$ | $[(s_2,1) \to (s_3,1)/L]$ |
| $1\hat{\star}1 \star 11 \star 1111$ | $[(s_3,\star) \to (s_3,1)/R]$ |
| $11\hat{1} \star 11 \star 1111$ | $[(s_3,1) \to (s_0,\star)/R]$ |
| $11 \star \hat{\star}11 \star 1111$ | $[(s_0,\star) \to (s_1,1)/R]$ |
| $11 \star 1\hat{1}1 \star 1111$ | $[(s_1,1) \to (s_2,\star)/L]$ |
| $11 \star \hat{1} \star 1 \star 1111$ | $[(s_2,1) \to (s_3,1)/L]$ |
| $11\hat{\star}1 \star 1 \star 1111$ | $[(s_3,\star) \to (s_3,1)/R]$ |
| $111\hat{1} \star 1 \star 1111$ | $[(s_3,1) \to (s_0,\star)/R]$ |
| $111 \star \hat{\star}1 \star 1111$ | $[(s_0,\star) \to (s_1,1)/R]$ |
| $111 \star 1\hat{1} \star 1111$ | $[(s_1,1) \to (s_2,\star)/L]$ |
| $111 \star \hat{1} \star \star 1111$ | $[(s_2,1) \to (s_3,1)/L]$ |
| $111\hat{\star}1 \star \star 1111$ | $[(s_3,\star) \to (s_3,1)/R]$ |
| $1111\hat{1} \star \star 1111$ | $[(s_3,1) \to (s_0,\star)/R]$ |
| $1111 \star \hat{\star} \star 1111$ | $[(s_0,\star) \to (s_1,1)/R]$ |
| $1111 \star 1\hat{\star}1111$ | $[(s_1,\star) \to (s_0,1)/L]$ |
| $1111 \star \hat{1}11111$ | $[(s_0,1) \to (s_1,1)/L]$ |
| $1111\hat{\star}111111$ | $[(s_1,\star) \to (s_0,1)/L]$ |
| $111\hat{1}1111111$ | $[(s_0,1) \to (s_1,1)/L]$ |
| $11\hat{1}11111111$ | $[(s_1,1) \to (s_2,\star)/L]$ |
| $1\hat{1} \star 11111111$ | $[(s_2,1) \to (s_3,1)/L]$ |
| $\hat{1}1 \star 11111111$ | $[(s_3,1) \to (s_0,\star)/R]$ |
| $\star\hat{1} \star 11111111$ | $[(s_0,1) \to (s_1,1)/L]$ |
| $\hat{\star}1 \star 11111111$ | $[(s_1,\star) \to (s_0,1)/L]$ |
| $\hat{\star}11 \star 11111111$ | $[(s_0,\star) \to (s_1,1)/R]$ |
| $1\hat{1}1 \star 11111111$ | $[(s_1,1) \to (s_2,\star)/L]$ |
| $\hat{1} \star 1 \star 11111111$ | $[(s_2,1) \to (s_3,1)/L]$ |
| $\hat{\star}1 \star 1 \star 11111111$ | $[(s_3,\star) \to (s_3,1)/R]$ |
| $1\hat{1} \star 1 \star 11111111$ | $[(s_3,1) \to (s_0,\star)/R]$ |
| $1 \star \hat{\star}1 \star 11111111$ | $[(s_0,\star) \to (s_1,1)/R]$ |
| $1 \star 1\hat{1} \star 11111111$ | $[(s_1,1) \to (s_2,\star)/L]$ |
| $1 \star \hat{1} \star \star 11111111$ | $[(s_2,1) \to (s_3,1)/L]$ |
| $1\hat{\star}1 \star \star 11111111$ | $[(s_3,\star) \to (s_3,1)/R]$ |
| $11\hat{1} \star \star 11111111$ | $[(s_3,1) \to (s_0,\star)/R]$ |
| $11 \star \hat{\star} \star 11111111$ | $[(s_0,\star) \to (s_1,1)/R]$ |
| $11 \star 1\hat{\star}11111111$ | $[(s_1,\star) \to (s_0,1)/L]$ |
| $11 \star \hat{1}111111111$ | $[(s_0,1) \to (s_1,1)/L]$ |
| $11\hat{\star}1111111111$ | $[(s_1,\star) \to (s_0,1)/L]$ |
| $1\hat{1}11111111111$ | $[(s_0,1) \to (s_1,1)/L]$ |
| $\hat{1}111111111111$ | $[(s_1,1) \to (s_2,\star)/L]$ |
| $\hat{\star} \star 111111111111$ | $[(s_2,\star) \to (s_4,1)/R]$ |
| $1\hat{\star}111111111111$ | akzeptiert |

| **Aufgabe 5** | **10 Punkte** |
|---|---|

**2014-N-05** **Binary Decision Diagram (BDD)**

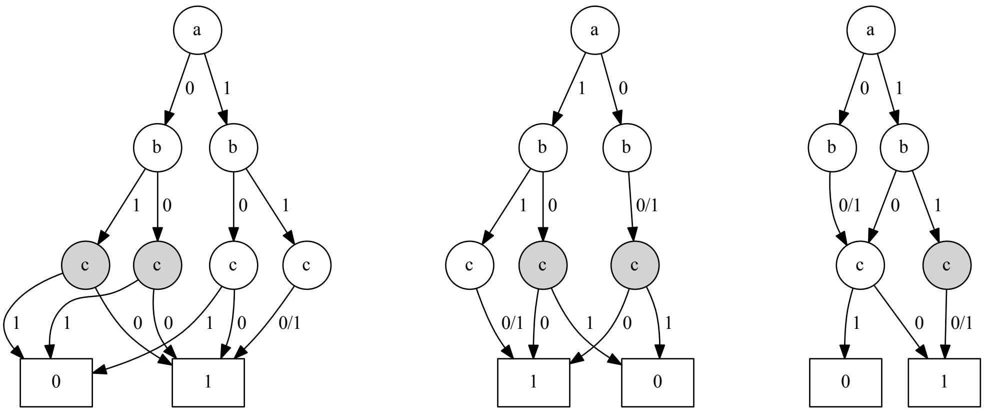
/ 10

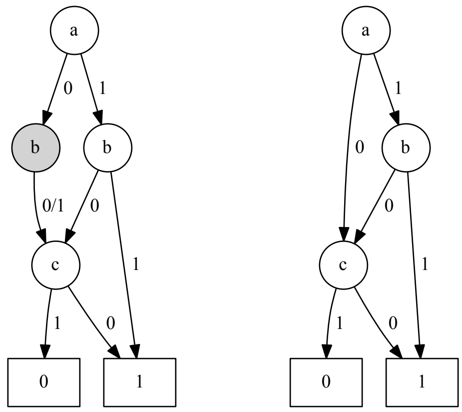
Gegeben sei eine Boolesche Funktion $f : \mathbb{B}^3 \to \mathbb{B}$ durch folgenden Baum:




Generieren Sie mit dem Algorithmus aus der Vorlesung ein BDD zur Funktion $f$.

**Lösung:**




a
0
1
b
b
0/1
0
1
c
1
0
0
1
a
1
0
b
0
c
1
1
0
0
1

| **Aufgabe 6** | **10 Punkte** |
|---|---|

**2014-N-06** **Zahlendarstellung**

/ 10

Gegeben sei der folgende Binärstring (Leerzeichen dienen der Lesbarkeit):

0000 0010 1001 0000 0000 0000 1000 0011

Welcher Zahlenwert ergibt sich, wenn man den String interpretiert als (es genügen jeweils Terme aus Summen von Zweierpotenzen etc.)

(a) Zahl in Einskomplement-Darstellung.

/ 2

**Lösung:**

$$2^{25} + 2^{23} + 2^{20} + 2^7 + 2^1 + 2^0 = 42991747$$

(b) Zahl in Zweikomplement-Darstellung.

/ 1

**Lösung:**

$$2^{25} + 2^{23} + 2^{20} + 2^7 + 2^1 + 2^0 = 42991747$$

(c) Zahl in Festkomma-Darstellung mit Komma in der Mitte.

/ 2

**Lösung:**

$$2^9 + 2^7 + 2^4 + 2^{-9} + 2^{-15} + 2^{-16} = 656{,}0019989013671875$$

(d) Zahl in Gleitpunkt-Darstellung nach IEEE-754.

/ 3

**Lösung:**

$$\begin{aligned} v &= (-1)^0 = 1; \\ c &= 2^0 + 2^2 = 5; e = 5 - 127 = -122; \\ m &= 2^{-3} + 2^{-16} + 2^{-22} + 2^{-23}; \\ &\Rightarrow v \cdot 2^e \cdot (1 + m) = 2{,}1159192026959298 \cdot 10^{-37} \end{aligned}$$

(e) Nennen Sie jeweils einen Vorteil einer Gleitkommadarstellung gegenüber einer Festkommadarstellung und umgekehrt.

/ 2

**Lösung:**

## Aufgabe 7 **12 Punkte**

**2014-N-07** **Fehlererkenn- und ~korrigierbarkeit**

/ 12

Gegeben sei ein Code $C$ von Wörtern der Länge 8 durch:

$$0101\,0101 \in C$$

und für alle $s, t \in \{0, 1\}^\star, a, b \in \{0, 1\}$:

$$sabt \in C \Rightarrow \begin{cases} sbat \in C, & \text{falls } |s| = |t|, \\ s00t \in C, & \text{falls } |s| < |t|, \\ s11t \in C, & \text{falls } |s| > |t| \end{cases}$$

(a) Geben Sie sechs verschiedene Codewörter $c_1, \ldots, c_6 \in C$ an.

/ 3

**Lösung:** Der Code enthält folgende Wörter:

| | |
|---|---|
| 0101 0101<br>0100 0101<br>0001 0101<br>0000 0101<br>0000 1101 | 0100 1101<br><br>0000 1101 |
| 0101 1101<br>0100 1111<br>0001 1101<br>0000 1111 | <br>0101 0111<br><br>0001 0111 |
| 0100 0111<br>0000 0111 | |
| 0101 1111<br>0100 1111<br>0001 1111 | <br>0101 0111 |

(b) Wie groß ist die Hammingzahl des Codes $C$ (begründen Sie kurz)?

/ 2

**Lösung:** 1, denn 0101 0101 und 0100 0101 sind Codewörter.

Gegeben sei ein zweiter Code $C'$ durch

$$0101\,0101 \in C'$$

und für alle $s, t \in \{0, 1\}^\star, a, b \in \{0, 1\}$:

$$sabt \in C' \Rightarrow a^{|s|+1} b^{|t|+1} \in C', \text{ falls } a \neq b$$

(c) Geben Sie alle Codewörter aus $C'$ an.

/ 3

**Lösung:** Der Code enthält folgende Wörter:

| | | |
|---|---|---|
| 0101 | 0101 | |
| 0111 | 1111 | $*$ |
| 1100 | 0000 | |
| 0001 | 1111 | $*$ |
| 1111 | 0000 | $\star$ |
| 0000 | 0111 | |
| 1111 | 1100 | $\star$ |
| 0000 | 0001 | |

(d) Geben Sie die Hammingzahl von $C'$ an.

/ 2

**Lösung:** 2 (siehe Kennzeichnung in Aufgabenteil c.)

(e) Geben Sie die Fehlererkenn- und ~korrigierbarkeit von $C'$ an.

/ 2

**Lösung:**

| **Aufgabe 8** | **8 Punkte** |
|---|---|

**2014-N-08** **Assembler**

/ 8

Gegeben sei das folgende Assemblerprogramm:

| | | | |
|---|---|---|---|
| `LOOP` | `LOAD` | `R1` | Beginn einer „Schleife“ |
| | `JUMPZERO` | `FERTIG` | Abbruch der „Schleife“ falls $n = 0$ |
| | `ADD` | `R1 R2 R2` | Addiere den aktuellen Wert zur Zwischensumme |
| | `SUBTRACT` | $R1$ `#1` $R1$ | Zähle den aktuellen Wert herunter |
| | `JUMP` | `LOOP` | Springe zum Beginn der „Schleife“ |
| `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 $R2$ steht.

(a) Geben Sie für jedes $n \in \{1, \ldots, 4\}$ jeweils den Wert $f(n)$ an.

/ 4

(b) Geben Sie eine allgemeine Definition für $f(n)$ an.

/ 2

(c) Wie viele Assembler-Befehle (Programmzeilen) werden bei der Berechnung von $f(n)$ abgearbeitet? (Geben Sie den genauen Wert an, nicht nur eine asymptotische Abschätzung.)

/ 2

**Lösung:** Berechnet wird $f(n) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ nach $5n + 3$ Befehlaufrufen.

| **Aufgabe 9** | **7 Punkte** |
|---|---|

**2014-N-09** **Betriebssysteme**

/ 7

(a) Nennen Sie die zwei in der Vorlesung betrachteten Klassen von Verfahren, mit denen ein Betriebssystem Rechenzeit auf wartende Prozesse zuteilen kann.

/ 1

(b) Nennen Sie für jede der Klassen ein Beispiel und erklären Sie jeweils kurz das zugrunde liegende Prinzip und mindestens einen Vorteil.

/ 4

(c) Was versteht man in diesem Zusammenhang unter „preemptive Scheduling“ bzw. „nonpreemptive Scheduling“?

/ 2

**Lösung:**

(a) Einfache Zuteilungsverfahren und Prioritätsgesteuerte Verfahren.

(b) (1) Einfache Zuteilungsverfahren arbeiten nach dem Prinzip, möglichst viel Prozessorleistung für Benutzerprozesse und möglichst wenig Prozessorleistung für Zuteilungsverfahren zu verbrauchen . Beispiel: First Come First Serve (FCFS) .

(2) Prioritätsgesteuerte Verfahren ordnen vorhandenen Prozessen eine Priorität $\in \mathbb{N}$ zu und arbeiten diese nach entsprechenden Prioritätsregeln ab . Beispiel: Höchste Priorität für kürzesten Prozess oder Prioritätsberechnung durch Kombination von Wartezeit und erwarteter Laufzeit.

(c) Man unterscheidet zwei Typen des Zeitscheibenverfahrens.

(1) preemptive: Wie bei Round-Robin (siehe oben). Ein Prozess kann vor Fertigstellung unterbrochen werden, um anderen Prozessen Rechenzeit zu geben .

(2) nonpreemptive: Ein Prozess wird nicht vorzeitig unterbrochen, er behält solange den Prozessor, bis er durch eine Aktion blockiert oder fertig ist .