Grundbegriffe der Informatik
Musterlösung zu Aufgabenblatt 10

Aufgabe 10.1 (1+4 Punkte)
Gegeben sei folgende Funktion $T(n)$, für $n \in \mathbb{N}_+$:

$$T(1) = 0, \qquad \forall n \geq 2 : \; T(n) = \sqrt{n} \cdot T(\lfloor \sqrt{n} \rfloor) + n$$

a) Berechnen Sie $T(4)$ und $T(16)$.

b) Zeigen Sie durch vollständige Induktion: $T(n) \in \mathrm{O}\left(n \log_2(\log_2 n)\right)$

Lösung 10.1

a) $T(4) = 8$ und $T(16) = 48$

b) Zu zeigen: $T(n) \in O(n \log_2(\log_2 n))$,
also $\forall n \geq n_0 : T(n) \leq c \cdot (n \log_2(\log_2 n))$, für ein $c \in \mathbb{R}_+ \wedge n_0 \in \mathbb{N}_0$

Wir zeigen das ganze intervallweise: Für $k \in \mathbb{N}_+$ sei $g_k = 2^{2^k}$. Dann ist $g_k^2 = g_{k+1}$

Wir definieren nun Intervalle I_k

$$I_k = \{n \in \mathbb{N}_+ \mid g_k \leq n \leq g_{k+1}\}$$

Genauer wollen wir jetzt zeigen: Für alle $k \in \mathbb{N}_+$ gilt: Wenn $n \in I_k$ ist, dann ist $T(n) \leq c \cdot n \log_2(\log_2 n)$

Induktionsanfang: $k = 1$. Für $n \in I_1 = \{4, 5, \ldots, 16\}$ ist zu zeigen: $T(n) \leq 4 \cdot n \log_2(\log_2 n)$.
Rechnung: $T(2) = \sqrt{2} \cdot T(1) + 2 = 2$ und $T(3) = \sqrt{3} \cdot T(1) + 3 = 3$.
Nun Fallunterscheidung:

$n \in \{4, \ldots, 8\}$**:** Dann ist $T(n) = \sqrt{n} \cdot T(2) + n = 2\sqrt{n} + n \leq 4n \leq 4n \log_2(\log_2 n)$

$n \in \{9, \ldots, 15\}$**:** Dann ist $T(n) = \sqrt{n} \cdot T(3) + n = 3\sqrt{n} + n \leq 4n \leq 4n \log_2(\log_2 n)$

$n = 16$**:** Dann ist $\log_2(\log_2 n) = 2$ und daher $T(n) = \sqrt{n} \cdot T(4) + n \leq 8n = 4n \log_2(\log_2 n)$.

Induktionsvoraussetzung: Für ein beliebiges aber festes k gelte: Wenn $n \in I_k$, dann ist $T(n) \leq 4 \cdot n \log_2(\log_2 n)$

Induktionsschluss: Zu zeigen: Wenn $n \in I_{k+1}$, dann ist $T(n) \leq 4 \cdot n \log_2(\log_2 n)$.

Sei also $n \in I_{k+1}$, also $g_{k+1} \leq n \leq g_{k+2}$. Da $\sqrt{\cdot}$ und $\lfloor \cdot \rfloor$ monoton wachsende Funktionen sind, gilt dann

$$\begin{array}{rc}
 & \sqrt{g_{k+1}} \leq \sqrt{n} \leq \sqrt{g_{k+2}} \\
\text{also} & g_k \leq \sqrt{n} \leq g_{k+1} \\
\text{und} & g_k \leq \lfloor \sqrt{n} \rfloor \leq g_{k+1}
\end{array}$$

Auf $\lfloor \sqrt{n} \rfloor$ ist also die Induktionsvoraussetzung anwendbar. Daher ergibt sich:

$$\begin{aligned}
T(n) &= \sqrt{n} \cdot T(\lfloor \sqrt{n} \rfloor) + n \\
&\leq \sqrt{n} \cdot 4 \cdot \lfloor \sqrt{n} \rfloor \log_2(\log_2 \lfloor \sqrt{n} \rfloor) + n && \text{nach Induktionsvoraussetzung} \\
&\leq \sqrt{n} \cdot 4 \cdot \sqrt{n} \log_2(\log_2 \sqrt{n}) + n \\
&= 4 \cdot n \cdot \log_2(\frac{1}{2} \log_2 n) + n && \text{da} \log_2 n^{1/2} = (1/2) \cdot \log_2 n \\
&= 4 \cdot n\big((\log_2(\log_2 n) - 1) + 1\big) && \text{da} \log_2(1/2) = -1 \\
&= 4 \cdot n \cdot \log_2(\log_2 n)
\end{aligned}$$

Damit ist die Behauptung gezeigt. d.h. mit der Wahl von $n_0 = 4$ und $c = 4$ gilt $\forall n \geq 4 : T(n) \leq 4 \cdot n \log_2(\log_2 n)$

Hinweis: Leider war der (formal korrekte) Beweis etwas schwerer als erwartet. Diese Lösung ist daher deutlich ausführlicher als wir es von den Studierenden erwarten.

Aufgabe 10.2 (2+2+2 Punkte)

Geben Sie (wenn möglich) mit Hilfe des Master-Theorems einen Ausdruck für die Laufzeit von $T(n)$ an. Falls das Mastertheorem nicht anwendbar ist, begründen Sie, warum das nicht möglich ist. In diesem Fall brauchen Sie keine Abschätzung anzugeben.

a) $T(n) = 2T(n/4) + 4\sqrt{n}$

b) $T(n) = 16T(n/4) + \frac{n^2}{\log n}$

c) $T(n) = 42\sqrt{n^3} + 2\sqrt{2}T(n/2) + 1212$

Lösung 10.2

a) $T(n) \in \Theta(\sqrt{n}\log n)$, nach Fall 2 des Master-Theorems: $4\sqrt{n} \in \Theta(n^{\log_4 2})$.

b) Mastertheorem nicht anwendbar, da $\frac{n^2}{\log n}$ nicht polynomiell kleiner als $\Theta(n^{\log_4 16})$. (Es lässt sich also kein ε finden, so dass gilt $\frac{n^2}{\log n} \in \mathrm{O}\left(n^{2-\varepsilon}\right)$.)

c) Zuerst ein wenig Umformen: $42\sqrt{n^3} + 1212 = 42n^{\frac{3}{2}} + 1212$

$n^{\log_2(2\sqrt{2})} = n^{\log_2(2^{\frac{3}{2}})} = n^{\frac{3}{2}}$

$T(n) \in \Theta(n^{\frac{3}{2}}\log n)$, nach Fall 2 des Master-Theorems: $42n^{\frac{3}{2}} + 1212 \in \Theta(n^{\frac{3}{2}})$

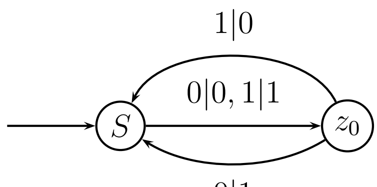
Aufgabe 10.3 (2+3 Punkte)

Geben Sie jeweils einen endlichen Mealy-Automaten an, so dass für alle Eingaben $w \in \{\mathtt{0},\mathtt{1}\}^*$ die Ausgabe $g^{**}(w) = w'$ ist, wobei w' wie folgt definiert ist:

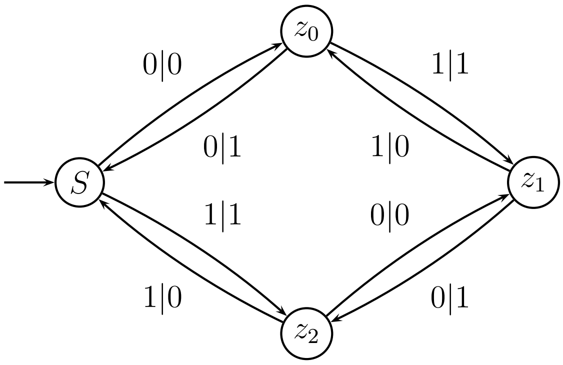
a) $w'(i) = \begin{cases} w(i) & \text{, wenn } i \bmod 2 \equiv 0 \\ 1 & \text{, wenn } i \bmod 2 \equiv 1 \wedge w(i) = 0 \\ 0 & \text{, wenn } i \bmod 2 \equiv 1 \wedge w(i) = 1 \end{cases}$

b) **jedes** zweite Vorkommen von `0` in w wird durch `1` ersetzt und **jedes** zweite Vorkommen von `1` in w wird durch `0` ersetzt.

Lösung 10.3



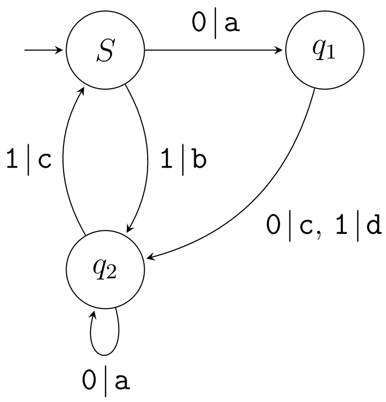
a)



b)

Aufgabe 10.4 (1+3 Punkte)

Gegeben sei folgender Mealy-Automat $M = (Z_m, S, \{\mathtt{0}, \mathtt{1}\}, f_m, \{\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d}\}, g_m)$:

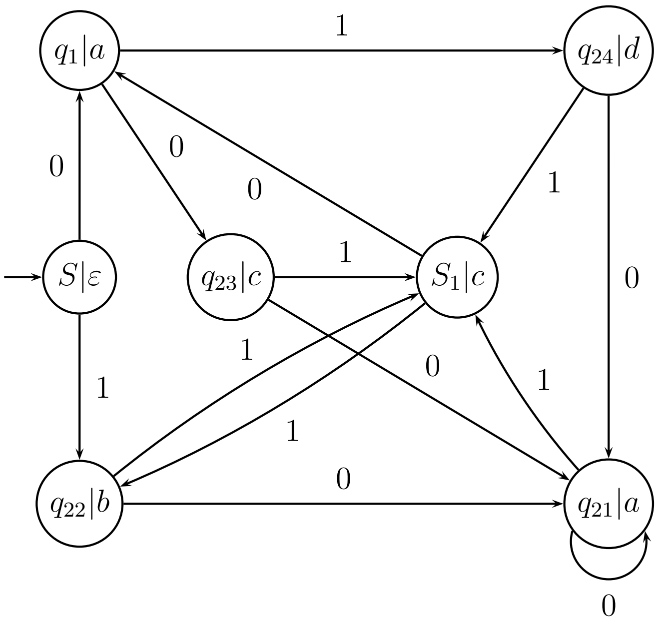


a) Geben Sie $f^{**}(S, \mathtt{0110})$ und $g^{**}(S, \mathtt{0110})$ an.

b) Geben Sie einen Moore-Automaten $N = (Z_n, S, \{\mathtt{0}, \mathtt{1}\}, f_n, \{\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d}\}, g_n)$ an, so dass für alle $w \in \{0, 1\}^+$ gilt: $g_m^{**}(S, w) = g_n^{**}(S, w)$.

Lösung 10.4

a) $f^{**}(S, \mathtt{0110}) = (Sq_1q_2Sq_1)$ und $g^{**}(S, \mathtt{0110}) = \mathtt{adca}$



b)