Diskrete Modellierung

Wintersemester 2016/2017

Mario Holldack, M. Sc.
Prof. Dr. Georg Schnitger
Hannes Seiwert, M. Sc.

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institut für Informatik
AG Theoretische Informatik

Übungsblatt 11

Ausgabe: 12.01.17
Abgabe: 19.01.17

Aufgabe 11.1 *Grundbegriffe endlicher Automaten* $(5+10+4+6=25$ Punkte)

Der DFA A_1 über dem Alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ sei durch das folgende Python-Programm gegeben:

```python
def dfa(word):                  # word bestehe nur aus den Symbolen 'a' und 'b'
    state = 0                   # Anfangszustand
    for symbol in word:         # Lies jedes Symbol des Eingabewortes
        if state == 0:          # aktueller Zustand: 0
            if symbol == 'a':
                state = 1       # Uebergang von 0 nach 1: delta(0,a) = 1
        elif state == 1:
            if symbol == 'a':
                state = 2
            if symbol == 'b':
                state = 0
        elif state == 2:
            if symbol == 'a':
                state = 3
            else:
                state = 0
        elif state == 3:
            state = 3

    if state == 3:
        return (state,True)     # Eingabewort akzeptieren
    else:
        return (state,False)    # Eingabewort verwerfen
```

Kommentar: Wir haben diese nicht sehr elegante Implementierung gewählt, um verschiedene Implementierungsmöglichkeiten von DFAs darzustellen.

a) Akzeptiert die obige Python-Implementierung wirklich nur Wörter aus $\{\mathtt{a}, \mathtt{b}\}^*$?

b) Geben Sie den DFA A_1 über dem Alphabet $\{\mathtt{a}, \mathtt{b}\}$ in grafischer Darstellung an.

c) Welche der folgenden Wörter liegen in $L(A_1)$, welche nicht?

- $w_1 := \varepsilon$
- $w_2 := 3$
- $w_3 := \mathtt{bbab}$
- $w_4 := \mathtt{baabaaab}$

Hinweis: Als Begründung können Sie beispielsweise den Zustand des Automaten nach dem Lesen des jeweiligen Wortes w_i angeben.

d) Beschreiben Sie die vom DFA A_1 akzeptierte Sprache $L(A_1) \subseteq \{\mathtt{a}, \mathtt{b}\}^*$ mathematisch oder umgangssprachlich.

Aufgabe 11.2 *Äquivalenzrelationen* (12 + (9+4) = 25 Punkte)

a) Sei $R = \{(1,1),(1,2),(3,2)\}$ eine Relation über der Menge $A = \{1,2,3,4\}$. Welche Paare $(x,y) \in A \times A$ müssen zu R mindestens hinzugefügt werden, um Relationen R_r, R_s und R_t zu erhalten, sodass

i) R_r reflexiv ist? ii) R_s symmetrisch ist? iii) R_t transitiv ist?

iv) Welche Äquivalenzklassen besitzt die Äquivalenzrelation $R_ä$, die aus R mit der geringsten Anzahl hinzugefügter Paare $(x,y) \in A \times A$ entsteht?

Eine Begründung ist nicht nötig.

b) Sei $G = (V,E)$ ein endlicher, gerichteter Graph und seien

- $R_{SZ} := \{(u,v) \in V^2 : u$ und v liegen in derselben starken Zusammenhangskomponente$\}$
- $R_{\leq} := \{(x,y) \in \mathbb{N}^2 : x \leq y\}$.

Relationen über V bzw. $\mathbb{N}$.

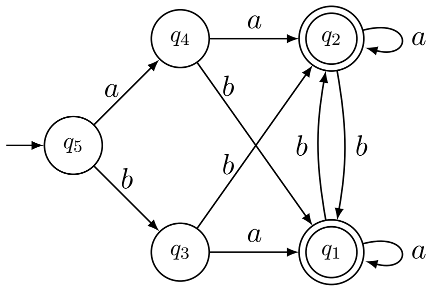
i) Zeigen Sie, dass R_{SZ} eine Äquivalenzrelation ist.

ii) Zeigen Sie, dass $R_{\leq}$ **keine** Äquivalenzrelation über $\mathbb{N}$ ist, indem Sie ein Gegenbeispiel zur Reflexivität, Symmetrie oder Transitivität angeben.

Hinweis: Um zu zeigen, dass eine Relation $R \subseteq A \times A$ eine Äquivalenzrelation ist, müssen Sie zeigen, dass R reflexiv, symmetrisch und transitiv ist.

Aufgabe 11.3 *Verschmelzungsrelation* (10 + 15 = 25 Punkte)

a) Betrachten Sie den folgenden DFA A über dem Alphabet $\Sigma = \{a,b\}$.



i) Weisen Sie die folgenden Inäquivalenzen bzgl. der Verschmelzungsrelation $\equiv_A$ nach, indem Sie geeignete Zeugen $z \in \Sigma^*$ angeben.

- $q_1 \not\equiv_A q_4$
- $q_3 \not\equiv_A q_5$

ii) Gibt es in A zwei verschiedene Zustände q_i und q_j, sodass $q_i \equiv_A q_j$ gilt? Begründen Sie Ihre Antwort. Geben Sie alle Äquivalenzklassen der Verschmelzungsrelation $\equiv_A$ an.

b) Mithilfe deterministischer endlicher Automaten soll die Teilbarkeit von natürlichen Zahlen in Darstellung zu einer Basis b untersucht werden. Für $b \in \mathbb{N}$ mit $b \geq 2$ betrachten wir das Eingabealphabet $\Sigma_b := \{0,1,\ldots,b-1\}$. Wir identifizieren jedes Wort $w = w_1 w_2 \ldots w_{|w|} \in \Sigma_b^*$ mit der natürlichen Zahl

$$z_b(w) := \begin{cases} \sum_{i=1}^{|w|} \left(w_i \cdot b^{|w|-i}\right), & \text{falls } w \neq \varepsilon \\ 0, & \text{falls } w = \varepsilon. \end{cases}$$

Das Wort $w = w_1 \dots w_{|w|} \in \Sigma_b^*$ ist also die Darstellung der natürlichen Zahl $z_b(w)$ im b-ären Zahlensystem. Beispielsweise ist $z_{10}(0123) = 0 \cdot 10^3 + 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 = 123$, $z_{10}(110) = 1 \cdot 10^2 + 1 \cdot 10^1 + 0 \cdot 10^0 = 110$ und $z_2(110) = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 6$.

Geben Sie für die Sprache

$$L := \{w \in \Sigma_2^* : z_2(w) \text{ ist durch 4 teilbar}\}$$

einen DFA B mit drei Zuständen q_1, q_2 und q_3 und $L(B) = L$ an. Weisen Sie die Inäquivalenzen $q_1 \not\equiv_B q_2$, $q_1 \not\equiv_B q_3$ und $q_2 \not\equiv_B q_3$ bzgl. der Verschmelzungsrelation durch geeignete Zeugen $z \in \Sigma_2^*$ nach.

Erläutern Sie kurz die Idee hinter Ihrer Lösung. Sie müssen nicht beweisen, dass $L(B) = L$ gilt.

Aufgabe 11.4 *Modellierung: Simple-TCP* (25 Punkte)

Das *Transmission Control Protocol* (kurz: *TCP*) gehört zu den wichtigsten Protokollen in vernetzten IT-Systemen. Kurz gesagt steuert TCP die Kommunikation zwischen zwei Rechnern, indem es *Sitzungen* zwischen den beiden Rechnern eröffnet. Ist eine Sitzung eröffnet, können beide Rechner beliebig viele Datenpakete austauschen und beispielsweise fehlerhaft erhaltene Nachrichten erneut anfordern. In dieser Aufgabe betrachten wir eine vereinfachte Variante von TCP, hier *Simple-TCP* genannt.

Seien r_1 und r_2 zwei Rechner, die mittels Simple-TCP eine Sitzung eröffnen möchten. Hierfür stehen ihnen jeweils die *Befehle* `SYN` (wie *synchronize*) und `ACK` (wie *acknowledge*) zur Verfügung. Damit r_1 erfolgreich eine Sitzung mit r_2 eröffnen kann, führen die Rechner gemäß Simple-TCP die folgenden drei Schritte der Reihe nach aus. Man spricht daher auch einem *Drei-Wege-Handschlag*:

1. Zunächst sendet r_1 ein `SYN` an r_2.
2. Daraufhin antwortet r_2 mit `SYN` und `ACK` (oder mit `ACK` und `SYN` in umgekehrter Reihenfolge).
3. Schließlich bestätigt r_1, indem er ein `ACK` an r_2 sendet.

Vor und nach jedem der drei Schritte dürfen von beiden Rechnern `SYN`- und `ACK`-Befehle gesendet werden. Auch während r_2 den Schritt 2 ausführt, darf r_1 beliebige `SYN`- und `ACK`-Sequenzen senden.

Modellieren Sie die Eröffnung einer Sitzung durch einen DFA $A_{\text{Simple-TCP}}$. Verwenden Sie das aus vier Symbolen bestehende Alphabet $\Sigma = \{(\texttt{SYN,-}), (\texttt{ACK,-}), (\texttt{-,SYN}), (\texttt{-,ACK})\}$, um die Kommunikation der beiden Rechner zu beschreiben. Beispielsweise repräsentiert das Symbol (`SYN`,-), dass r_1 ein `SYN` an r_2 sendet (und r_2 schweigt).

Der DFA $A_{\text{Simple-TCP}}$ soll genau die Wörter über Σ^* akzeptieren, die einen Drei-Wege-Handschlag enthalten, wobei die drei Schritte nicht zwingenderweise direkt nacheinander vorkommen müssen. So wird auch durch das Wort

$$\texttt{(ACK,-)(-,ACK)}\underbrace{\texttt{(SYN,-)}}_{1.}\texttt{(SYN,-)}\underbrace{\underbrace{\texttt{(-,ACK)}}_{2.a}\texttt{(ACK,-)(SYN,-)}\underbrace{\texttt{(-,SYN)}}_{2.s}}_{2.}\texttt{(-,SYN)(SYN,-)}\underbrace{\texttt{(ACK,-)}}_{3.}\texttt{(-,SYN)}$$

erfolgreich eine Sitzung eröffnet. Achtung: Die `SYN`-Befehle zwischen den Schritten 1 und 2 sowie zwischen 2 und 3 sind gemäß Simple-TCP erlaubt, da zwischen den drei Schritten von beiden Rechnern beliebige Befehle gesendet werden dürfen. Ebenfalls erlaubt ist, dass Schritt 2 zwischen 2.a und 2.s durch weitere Vorkommnisse von `SYN`- und `ACK`-Befehlen unterbrochen wird.

Es genügt eine grafische Darstellung des DFAs ohne weitere Begründung.