Karlsruher Institut für Technologie
Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

# Übungsblatt 1

Vorlesung Theoretische Grundlagen der Informatik im WS 20/21

**Ausgabe:** 5. November 2020
**Abgabe:** 24. November 2020, 11:30 Uhr (digital im ILIAS)

Bitte bearbeiten Sie die Aufgaben **handschriftlich** und laden Sie eine eingescannte PDF-Version im Übungsmodul Ihrer ILIAS-Tutoriumsgruppe hoch! Beschriften Sie Ihren handschriftlichen Aufschrieb gut sichtbar mit Name und Matrikelnummer. Nicht handschriftliche oder unbeschriftete Abgaben werden nicht akzeptiert!

## Aufgabe 1 (1 + 1 + 1 + 1 + 1 = 5 Punkte)

Gegeben seien zwei Sprachen $L_1, L_2 \subset \Sigma^*$ über dem Alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Dabei sei $L_1$ die Sprache der Wörter, die mit a beginnen und $L_2$ die Sprache der Wörter mit einer geraden Anzahl an b. Geben Sie reguläre Ausdrücke für folgende Sprachen an.

(a) $L_1 \cup L_2$

(b) $L_2 \cdot L_1$

(c) $L_1 \setminus L_2$

(d) $L_2/L_1$

(e) $L_1^c$

**Lösung:**

(a) $\mathsf{a}(\mathsf{a} \cup \mathsf{b})^* \cup (\mathsf{b}\mathsf{a}^*\mathsf{b} \cup \mathsf{a})^*$

(b) $(\mathsf{b}\mathsf{a}^*\mathsf{b} \cup \mathsf{a})^*\mathsf{a}(\mathsf{a} \cup \mathsf{b})^*$

(c) $\mathsf{a}^+\mathsf{b}(\mathsf{b}\mathsf{a}^*\mathsf{b} \cup \mathsf{a})^*$

(d) $(\mathsf{a} \cup \mathsf{b})^*$

(e) $\mathsf{b}(\mathsf{a} \cup \mathsf{b})^* \cup \varepsilon$

## Aufgabe 2 (2 + 2 = 4 Punkte)

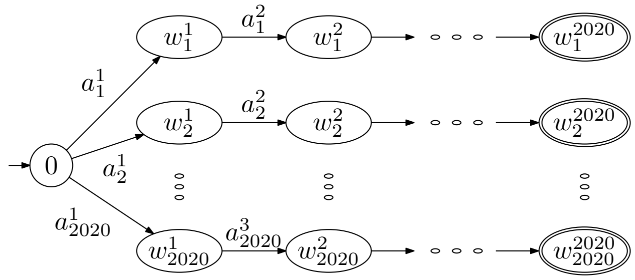
Es sei $\Sigma$ ein endliches Alphabet und $L = \{w_1, w_2, \ldots, w_{2020}\}$ eine Menge von 2020 Wörtern der Länge 2020 über dem Alphabet $\Sigma$.

(a) Skizzieren Sie den Übergangsgraphen eines NEA, der $L$ erkennt.

(b) Geben Sie einen DEA an, der $L$ erkennt.

**Lösung:**

(a)



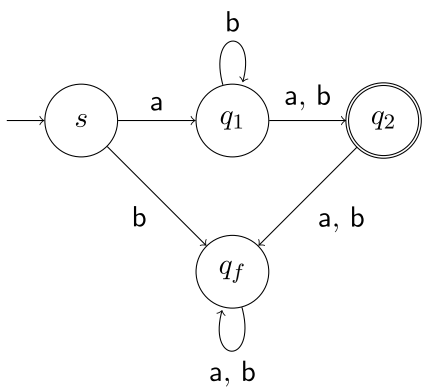
Dabei bezeichnet $a_j^i$ das $i$-te Zeichen des Wortes $w_j \in L$.
(Implizit existiert ein Fehlerzustand, in den alle nicht explizit angegebenen Übergänge führen.)

(b) Als Zustandsmenge $Q$ wählen wir die Menge der Wörter über $\Sigma$, deren Länge höchstens 2020 ist, und außerdem einen Fehlerzustand $f$. Wähle $\varepsilon \in Q$ als Startzustand und $L \subset Q$ als Menge der akzeptierenden Zustände. Für $q \in Q, a \in \Sigma$ definiere $\delta(q, a) = qa$ falls $|q| < 2020$ und $\delta(q, a) = f$ sonst.

## Aufgabe 3

(1 + 3 = 4 Punkte)

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Übergangsgraphen gegeben ist:



(a) Welche Sprache erkennt der Automat $\mathcal{A}$?

(b) Konstruieren Sie mithilfe der Potenzmengenkonstruktion den zu $\mathcal{A}$ äquivalenten deterministischen Automaten $\mathcal{A}'$. Geben Sie außerdem den Übergangsgraphen von $\mathcal{A}'$ an. Bezeichnen Sie insbesondere die Zustände in $\mathcal{A}'$ mit den Teilmengen der Zustände in A, die sie repräsentieren.

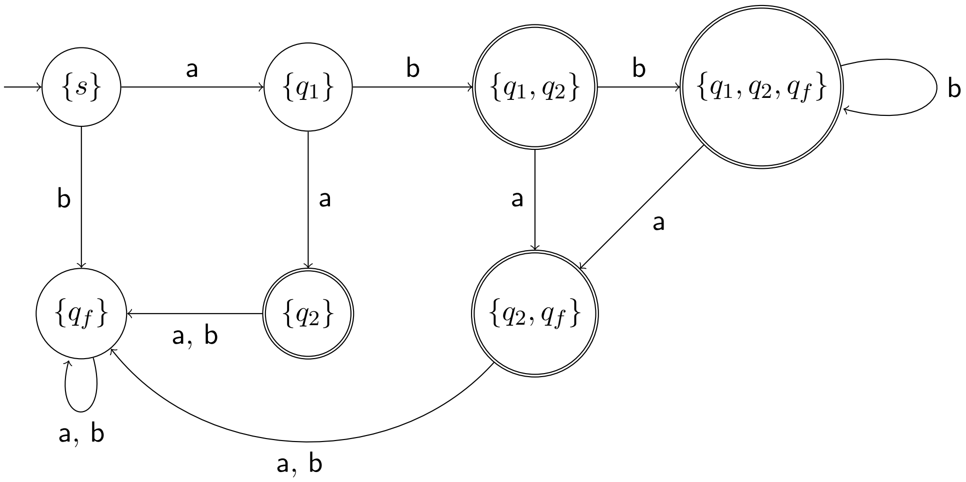
**Lösung:**

(a) $L(\mathcal{A}) = \mathsf{ab}^*(\mathsf{a} \cup \mathsf{b})$

(b) Potenzmengenkonstruktion mit Startzustand $\{s\}$ (Endzustände sind unterstrichen):

| Zustand | Übergänge a | Übergänge b |
|---|---|---|
| $\{s\}$ | $\{q_1\}$ | $\{q_f\}$ |
| $\{q_f\}$ | $\{q_f\}$ | $\{q_f\}$ |
| $\{q_1\}$ | $\{q_2\}$ | $\{q_1, q_2\}$ |
| $\underline{\{q_2\}}$ | $\{q_f\}$ | $\{q_f\}$ |
| $\underline{\{q_1, q_2\}}$ | $\{q_2, q_f\}$ | $\{q_1, q_2, q_f\}$ |
| $\underline{\{q_2, q_f\}}$ | $\{q_f\}$ | $\{q_f\}$ |
| $\underline{\{q_1, q_2, q_f\}}$ | $\{q_2, q_f\}$ | $\{q_1, q_2, q_f\}$ |

Zustandsgraph von $\mathcal{A}'$:




## Aufgabe 4

(1 + 1 + 1 = 3 Punkte)

Gegeben seien drei Grammatiken $G_i = (\Sigma, V, S, R_i), i = 1, 2, 3$ mit $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}, V = \{S, A, B, C\}$ und folgenden Produktionsregeln

(a) $R_1 = \{S \to AB, A \to \mathsf{ba}A\mathsf{ba} \mid \mathsf{baba}, B \to \mathsf{c}S \mid \mathsf{c}\}$

(b) $R_2 = \{S \to AB, A \to \mathsf{ba}A\mathsf{ba} \mid \mathsf{c}, B \to \mathsf{c}S \mid \mathsf{c}\}$

(c) $R_3 = \{S \to AC, A \to \mathsf{a}BA \mid \mathsf{a}B, B \to \mathsf{b}B \mid \varepsilon, C \to \mathsf{c}C \mid \varepsilon\}$

Ist $L(G_i)$ eine reguläre Sprache? Wenn ja, geben Sie einen regulären Ausdruck an. Wenn nicht, geben Sie eine formale Definition der Sprache in Mengenschreibweise an.

**Lösung:**

(a) Ja. $((\mathsf{baba})^+\mathsf{c})^+$

(b) Nein. $L(G_2) = \{(\mathsf{ba})^i\mathsf{c}(\mathsf{ba})^i\mathsf{c} \mid i \in \mathbb{N}_0\}^+$.

(c) Ja. $(\mathsf{ab}^*)^+\mathsf{c}^*$

## Aufgabe 5

(1 + 2 = 3 Punkte)

(a) Offensichtlich lässt sich jeder deterministische endliche Automat (DEA) durch einen nichtdeterministischen endlichen Automaten (NEA) simulieren, indem der Nichtdeterminismus in der Überführungsfunktion nicht ausgenutzt wird. Sei $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ ein NEA, der wie ein DEA arbeitet. Welche Bedingungen müssen für die Überführungsfunktion $\delta\colon Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ gelten? Formulieren Sie die Bedingungen als prädikatenlogische Formel.

(b) Für NEA wurde in der Vorlesung die erweiterte Überführungsfunktion $\bar{\delta}\colon Q \times \Sigma^* \to 2^Q$ eingeführt, die einen Zustand $q$ und ein Wort $w$ auf die Menge der Zustände abbildet, die von $q$ nach der Abarbeitung von $w$ erreichbar sind. Mithilfe der Simulation aus dem vorigen Aufgabenteil lässt sich $\bar{\delta}$ auch bei DEA anwenden. Kreuzen Sie in der folgenden Tabelle an, welche Aussage für *alle* NEA bzw. DEA $\mathcal{A}$ mit $L(\mathcal{A}) \neq \emptyset$ und $L(\mathcal{A}) \neq \Sigma^*$ gelten.

| Aussage | NEA | DEA |
|---|---|---|
| $\exists w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$ | | |
| $\exists w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$ | | |
| $\forall w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$ | | |
| $\forall w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$ | | |
| $\forall w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$ | | |
| $\forall w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \notin F$ | | |

**Lösung:**

(a) $\forall q \in Q\colon (\delta(q, \varepsilon) = \emptyset \land \forall x \in \Sigma\colon |\delta(q, x)| = 1)$

(b)

| Aussage | NEA | DEA |
|---|---|---|
| $\exists w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$ | × | × |
| $\exists w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$ | | × |
| $\forall w \in L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$ | × | × |
| $\forall w \in L(\mathcal{A}) \forall q \in \bar{\delta}(s, w)\colon q \in F$ | | × |
| $\forall w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \in F$ | | |
| $\forall w \notin L(\mathcal{A}) \exists q \in \bar{\delta}(s, w)\colon q \notin F$ | (*) | × |

(*) Auf den ersten Blick mag es so scheinen, als ob dieses Feld angekreuzt werden muss: Für Wörter $w \in L(\mathcal{A})$ sind *alle* möglichen Abarbeitungen nicht akzeptierend, also erscheint es logisch, dass auch *eine* nicht akzeptierende Abarbeitung existieren muss. Das ist nicht der Fall, weil es bei einem NEA vorkommen kann, dass es für ein Wort *gar keine* Abarbeitung gibt, also $\bar{\delta}(s, w) = \emptyset$. Das liegt daran, dass die NEA-Überführungsfunktion auf eine Menge von möglichen Folgezuständen abbildet. Diese Menge darf auch leer sein. In diesem Fall wird das gelesene Wort $w$ nicht akzeptiert, weil die Akzeptanzbedingung $\bar{\delta}(s, w) \cap F \neq \emptyset$ nicht erfüllt ist.

## Aufgabe 6

(2 + 3 + 2 + 3 = 10 Punkte)

Sei $L$ eine reguläre Sprache über einem endlichen Alphabet $\Sigma$. Für ein Wort $w = w_1 w_2 \dots w_{n-1} w_n \in \Sigma^*$ definieren wir dessen Umkehrung als $\overleftarrow{w} := w_n w_{n-1} \dots w_2 w_1$.

Wir betrachten zunächst das Beispielalphabet $\Sigma = \{\mathsf{a}, \mathsf{b}\}$ und folgende Beispielsprache:

$$L_1 := \{\mathsf{aa}x\mathsf{b} \mid x \in \Sigma^* \text{ und } x \text{ enthält } \mathsf{ba} \text{ als Teilwort}\}$$

(a) Geben Sie reguläre Ausdrücke für $L_1$ und $\overleftarrow{L_1}$ an.

(b) Geben Sie endliche Automaten für $L_1$ und $\overleftarrow{L_1}$ an.

Nun betrachten wir ein beliebiges endliches Alphabet $\Sigma$ und eine beliebige reguläre Sprache $L$. Sie sollen nun auf zwei verschiedene Wege zeigen, dass $\overleftarrow{L} := \{\overleftarrow{w} \mid w \in L\}$ ebenfalls regulär ist.

(c) Beschreiben Sie, wie aus einem regulären Ausdruck für $L$ ein regulärer Ausdruck für $\overleftarrow{L}$ konstruiert werden kann. Beweisen Sie, dass Ihre Konstruktion korrekt ist.

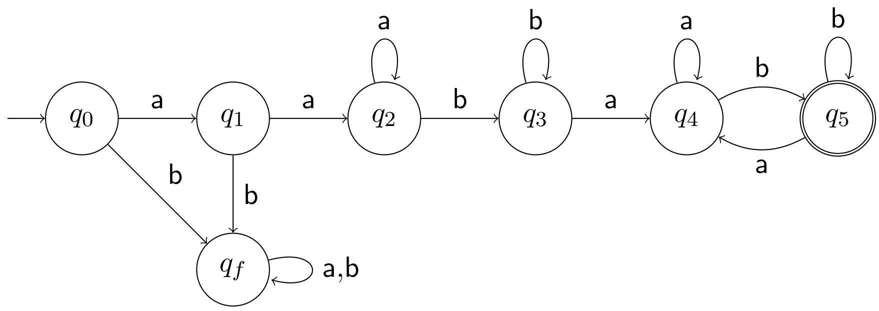
(d) Beschreiben Sie, wie aus einem endlichen Automaten für $L$ ein endlicher Automat für $\overleftarrow{L}$ konstruiert werden kann. Beweisen Sie, dass Ihre Konstruktion korrekt ist.

**Lösung:**

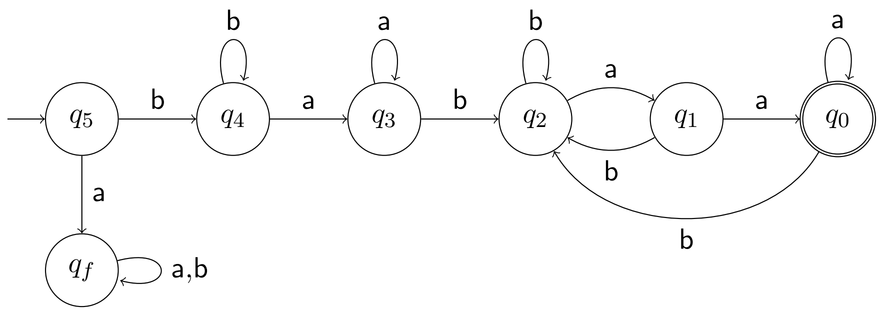
(a) $\alpha(L_1) = \mathsf{aa}(\mathsf{a} \cup \mathsf{b})^*\mathsf{ba}(\mathsf{a} \cup \mathsf{b})^*\mathsf{b}$

$\alpha(\overleftarrow{L_1}) = \mathsf{b}(\mathsf{a} \cup \mathsf{b})^*\mathsf{ab}(\mathsf{a} \cup \mathsf{b})^*\mathsf{aa}$

(b) Automat für $L_1$:




Automat für $\overleftarrow{L_1}$:



(c) Sei $\alpha$ ein regulärer Ausdruck für $L$. Wir zeigen per Induktion über der Anzahl $n$ der Operationen $\cup$, $\cdot$, $^*$ in $\alpha$, dass es einen regulären Ausdruck $\beta$ gibt, der $\overleftarrow{L}$ beschreibt.

$n = 0$**:** Dann ist $\alpha = \varepsilon$ oder $\alpha = a$ mit $a \in \Sigma$ oder $\alpha = \emptyset$. In jedem Fall gilt $\overleftarrow{L} = L$ und wir wählen $\beta = \alpha$.

$n \geq 1$**:** Dann ist $\alpha = \alpha_1 \cup \alpha_2$ oder $\alpha = \alpha_1 \cdot \alpha_2$ oder $\alpha = \alpha_1^*$ für reguläre Ausdrücke $\alpha_1, \alpha_2$, je mit höchstens $n-1$ Operationen $\cup$, $\cdot$, $^*$. Nach IV existieren reguläre Ausdrücke $\beta_1$ und $\beta_2$ die $\overleftarrow{L(\alpha_1)}$ beziehungsweise $\overleftarrow{L(\alpha_2)}$ beschreiben. Wir wählen $\beta$ wie folgt:

$$\begin{aligned} \alpha = \alpha_1 \cup \alpha_2 &\quad\Rightarrow\quad \beta = \beta_1 \cup \beta_2 \\ \alpha = \alpha_1 \cdot \alpha_2 &\quad\Rightarrow\quad \beta = \beta_2 \cdot \beta_1 \\ \alpha = \alpha_1^\star &\quad\Rightarrow\quad \beta = \beta_1^\star \end{aligned}$$

Nun ist $\beta$ ein regulärer Ausdruck der die Sprache $\overleftarrow{L}$ beschreibt.

(d) Sei $\mathcal{A} = \{Q, \Sigma, \delta, s, F\}$ ein deterministischer Automat, der $L$ akzeptiert. Konstruiere NEA $\mathcal{A}' = \{Q', \Sigma, \delta', s', F'\}$ wie folgt:

- $Q' = Q \cup \{s'\}$ mit $s' \notin Q$
- Für jedes $a \in \Sigma$ und $q \in Q$ definieren wir $\delta'(q, a) = \{q' \in Q \mid \delta(q', a) = q\}$. Es werden also alle Zuständsübergänge *umgedreht*. Dabei können nichtdeterministische Übergängen entstehen.
- Der Startzustand von $\mathcal{A}$ wird der akzeptierende Zustand von $\mathcal{A}'$: $F' = \{s\}$
- $s'$ ist der Startzustand von $\mathcal{A}'$ mit $\delta'(s', \varepsilon) = F$. Es gibt also $\varepsilon$-Übergänge.

Durch Induktion lässt sich für jedes $w \in \Sigma^*$ und $q \in Q$ zeigen: $\overline{\delta'}(q, \overleftarrow{w}) = \{q' \in Q \mid \delta(q', w) = q\}$. Wir zeigen nun, dass ein beliebiges Wort $w \in \Sigma^*$ genau dann von $\mathcal{A}$ akzeptiert wird, wenn es eine akzeptierende Abarbeitung von $\overleftarrow{w}$ in $\mathcal{A}'$ gibt:

$$\begin{aligned} \overleftarrow{w} \in L(A') &\Leftrightarrow \overline{\delta'}(s', \overleftarrow{w}) \cap F' \neq \emptyset \\ &\Leftrightarrow s \in \overline{\delta'}(s', \overleftarrow{w}) \\ &\Leftrightarrow \exists f \in F\colon s \in \overline{\delta'}(f, \overleftarrow{w}) \\ &\Leftrightarrow \exists f \in F\colon s \in \{q' \in Q \mid \delta(q', w) = f\} \\ &\Leftrightarrow \exists f \in F\colon \delta(s, w) = f \\ &\Leftrightarrow \delta(s, w) \in F \\ &\Leftrightarrow w \in L(\mathcal{A}) \end{aligned}$$