Prof. T. Nipkow
M. Eberl, J. Rädle, S. Wimmer
Technische Universität München
KEINE ABGABE

Einführung in die theoretische Informatik

Sommersemester 2019 – Übungsblatt Lösungsskizze 3

AUFGABE 3.1. (*Wichtige Begriffe*) Stufe A

Überprüfen Sie, dass Sie die Folgenden Begriffe korrekt definieren können.

- regulärer Ausdruck
- ε-NFA
- Produktkonstruktion
- rekursive Prozedur

AUFGABE 3.2. (*Automata Tutor*) Stufe B

Hinweis: Die folgenden Aufgaben stehen auch im AutomataTutor zur Verfügung. Zudem finden Sie dort weitere neue Aufgaben. Die Aufgaben finden Sie unter den Kategorien "English to Regular Expression" und "Regular Expression to NFA". Einige der Aufgaben sind auch bepunktete Hausaufgaben. Beachten Sie, dass der AutomataTutor automatisch $c \mid \varepsilon$ als $c?$ abkürzt.

Geben Sie für jede der folgenden Sprachen einen regulären Ausdruck an, der genau die Sprache beschreibt. Verwenden Sie für die ersten drei Aufgaben das Alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ und für die letzten beiden $\Sigma = \{0, 1\}$.

(a) Wörter gerader Länge.
(b) Wörter, die mit einem a beginnen und enden, sowie Wörter, die mit einem b beginnen und enden.
(c) Wörter, in denen kein a neben einem b steht.
(d) Zahlen in Binärdarstellung (most-significant-bit-first), die durch 2 teilbar sind.
(e) Zahlen in Binärdarstellung (most-significant-bit-first), die nicht durch 4 teilbar sind.

Lösungsskizze

(a) $((\mathsf{a} \mid \mathsf{b} \mid \mathsf{c})(\mathsf{a} \mid \mathsf{b} \mid \mathsf{c}))^*$
(b) $\mathsf{a} \mid \mathsf{b} \mid \mathsf{a}(\mathsf{a} \mid \mathsf{b} \mid \mathsf{c})^*\mathsf{a} \mid \mathsf{b}(\mathsf{a} \mid \mathsf{b} \mid \mathsf{c})^*\mathsf{b}$
(c) $((\mathsf{a}^* \mid \mathsf{b}^*)\mathsf{cc}^*)^*(\mathsf{a}^* \mid \mathsf{b}^*)$
(d) $(0 \mid 1)^*0$
(e) $(0 \mid 1)^*(1 \mid 10)$

AUFGABE 3.3. Stufe B

Sei $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Geben Sie einen regulären Ausdruck mit möglichst wenigen Zeichen für die Sprache an, in der alle Wörter gleich oft die Zeichenketten ab und ba enthalten.
Beispiel: Das Wort abab enthält zweimal ab, aber nur einmal ba und soll somit *kein* Element der Sprache sein.

Lösungsskizze

$$\varepsilon \mid (\mathsf{a}^+\mathsf{b}^+)^*\mathsf{a}^+ \mid (\mathsf{b}^+\mathsf{a}^+)^*\mathsf{b}^+$$

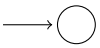
AUFGABE 3.4. Stufe C

(a) Geben Sie für jeden Teilausdruck des regulären Ausdrucks $((\mathsf{a}\emptyset)^*\mathsf{b} \mid \mathsf{ab})^*$ einen ε-NFA an. Verwenden Sie hierbei das Verfahren aus der Vorlesung ohne Abänderung, d.h. Sie sollen weder die regulären Ausdrücke noch die Automaten, die Sie konstruieren, vereinfachen.
Hinweis: Es gibt 9 Teilausdrücke. Der Ausdruck selbst ist auch ein Teilausdruck.
(b) Überführen Sie den ε-NFA für den gesamten Ausdruck aus Teilaufgabe a) in einen DFA. Kombinieren Sie hierzu die Potenzmengenkonstruktion mit der Umwandlung eines ε-NFA in einen NFA. (Vorlesung Folie 58)

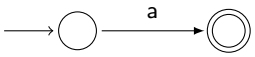
Lösungsskizze

(a)

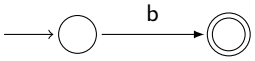
- $\emptyset$:



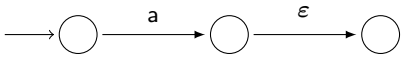
- a:



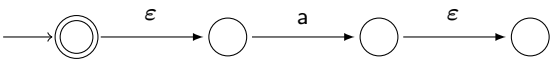
- b:



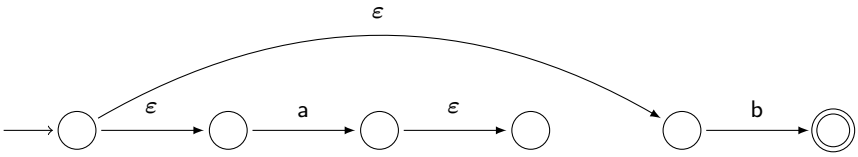
- a$\emptyset$:



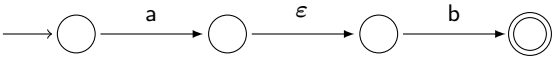
- $(a\emptyset)^*$:



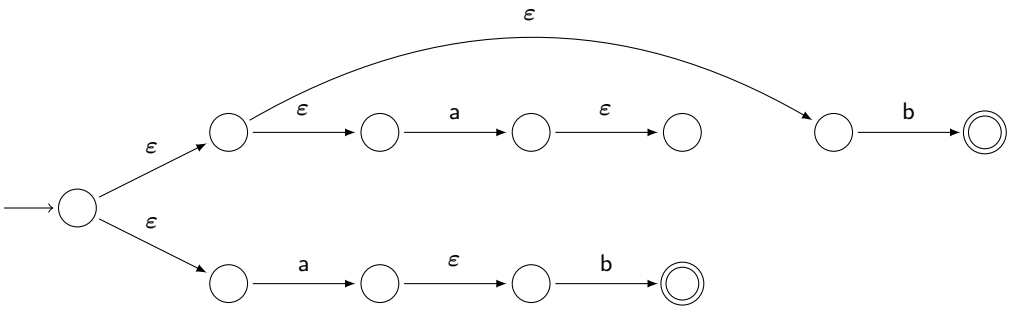
- $(a\emptyset)^*b$:



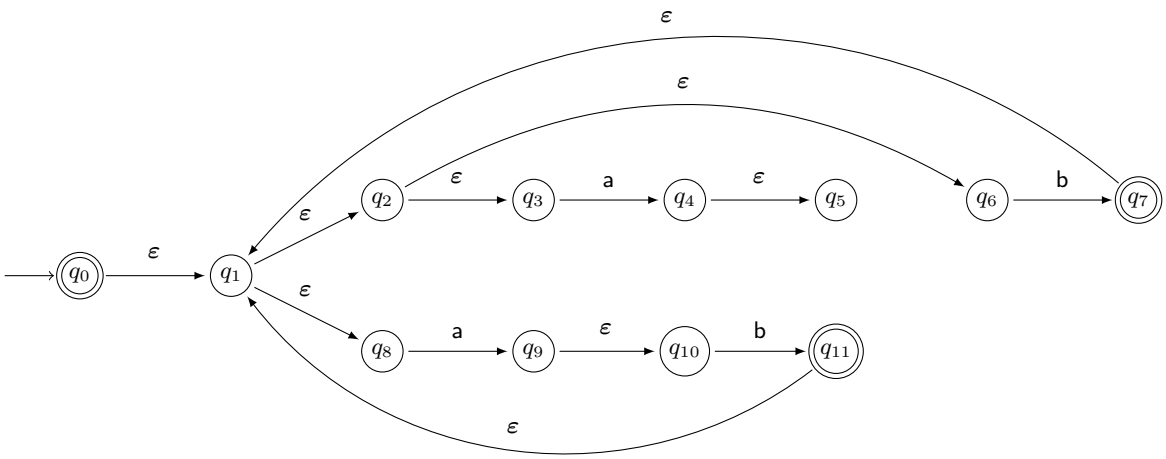
- ab:



- $(a\emptyset)^*b \mid ab$:



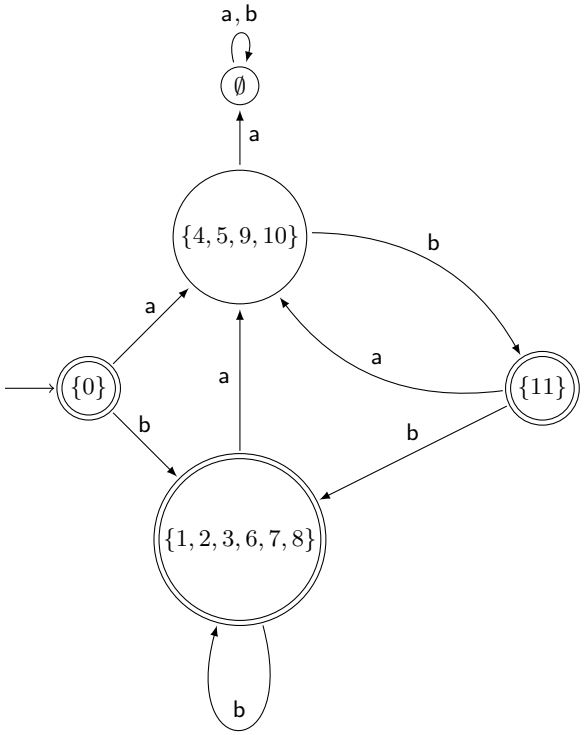
- $((a\emptyset)^*b \mid ab)^*$:



(b) Wir konstruieren zuerst einen NFA für $((\mathsf{a}\emptyset)^*\mathsf{b} \mid \mathsf{ab})^*$ indem wir die ε-Transitionen ersetzen. Aufgrund der vielen Transitionen geben wir den Automaten in Tabellennotation an:

	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}
q_0	-	b	b	b	a	a	b	b	b	a	a	-
q_1	-	b	b	b	a	a	b	b	b	a	a	-
q_2	-	b	b	b	a	a	b	b	b	-	-	-
q_3	-	-	-	-	a	a	-	-	-	-	-	-
q_4	-	-	-	-	-	-	-	-	-	-	-	-
q_5	-	-	-	-	-	-	-	-	-	-	-	-
q_6	-	b	b	b	-	-	b	b	b	-	-	-
q_7	-	b	b	b	-	-	b	b	b	-	-	-
q_8	-	-	-	-	-	-	-	-	-	a	a	-
q_9	-	b	b	b	-	-	b	-	b	-	-	b
q_{10}	-	b	b	b	-	-	b	-	b	-	-	b
q_{11}	-	b	b	b	a	a	b	b	b	a	a	-

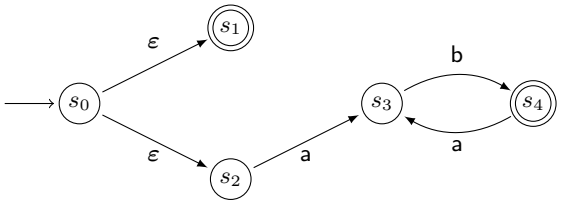
Die Zustände q_4 und q_5 sowie q_9 und q_{10} sind äquivalent, da sie die gleichen ausgehenden und eingehenden Kanten haben und keine Endzustände sind. Daher kann man sie für die Potenzmengenkonstruktion jeweils als ein Zustand betrachten.
Mit der Potenzmengenkonstruktion ergibt sich der folgende DFA:



Es ist auch möglich die Potenzmengenkonstruktion direkt auf dem ε-NFA durchzuführen.

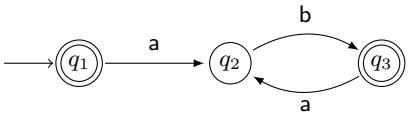
AUFGABE 3.5. Stufe C

Gegeben sei der folgende ε-NFA:



Wandeln Sie den Automat gemäß dem Verfahren aus der Vorlesung in einen äquivalenten regulären Ausdruck um und vereinfachen Sie diesen soweit wie möglich.
Achten Sie hierbei darauf die regulären Ausdrücke die Sie als Zwischenergebnisse erhalten zu vereinfachen.

Lösungsskizze
Zunächst müssen wir den Automaten in einen DFA umwandeln. Dazu müssen wir lediglich die ε-Transitionen entfernen, der resultierende Automat ist bereits deterministisch:



Alle nicht eingezeichneten Transitionen führe in den Fangzustand. Da der Fangzustand nie Zwischenzustand außer auf Pfaden, die in ihm selbst enden, sein kann und da der Fangzustand kein Endzustand ist, können wir ihn im folgenden außer Betracht lassen.
Wir wenden das Verfahren aus der Vorlesung schrittweise an. Für die Ausdrücke ohne Zwischenzustand (Schritt 0) erhalten wir (leere Ausdrücke werden nicht dargestellt):

$$\alpha_{11}^0 = \varepsilon \quad \alpha_{12}^0 = a \quad \alpha_{22}^0 = \varepsilon \quad \alpha_{23}^0 = b \quad \alpha_{32}^0 = a \quad \alpha_{33}^0 = \varepsilon$$

Mit q_1 als Zwischenzustand (Schritt 1) ergibt sich nichts neues: $\forall i, j.\, \alpha_{ij}^1 \equiv \alpha_{ij}^0$.
Mit q_2 als Zwischenzustand erhalten wir zwei neue Ausdrücke (Schritt 2):

$$\alpha_{13}^2 \equiv \emptyset \mid a(\varepsilon)^* b \equiv ab \quad \alpha_{33}^2 \equiv \varepsilon \mid a(\varepsilon)^* b \equiv \varepsilon \mid ab$$

Der Schritt, in dem wir q_3 als Zwischenzustand betrachten ist der letzte Schritt. Deswegen interessieren wir uns nur noch für Ausdrücke, die vom Startzustand in einen Endzustand führen:

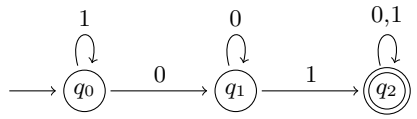
$$\alpha_{13}^3 \equiv ab \mid (ab(\varepsilon \mid ab)^*(\varepsilon \mid ab)) \equiv ab(ab)^* \quad \alpha_{11}^3 \equiv \varepsilon \mid (ab(\varepsilon \mid ab)^* \emptyset) \equiv \varepsilon$$

Damit ergibt sich für den Gesamtausdruck: $\alpha \equiv \varepsilon \mid ab(ab)^* \equiv (ab)^*$.

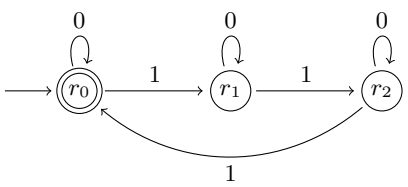
AUFGABE 3.6. Stufe C

Gegeben seien zwei DFAs D_1 und D_2 über dem gleichen Eingabealphabet $\Sigma = \{0, 1\}$.

D_1 :



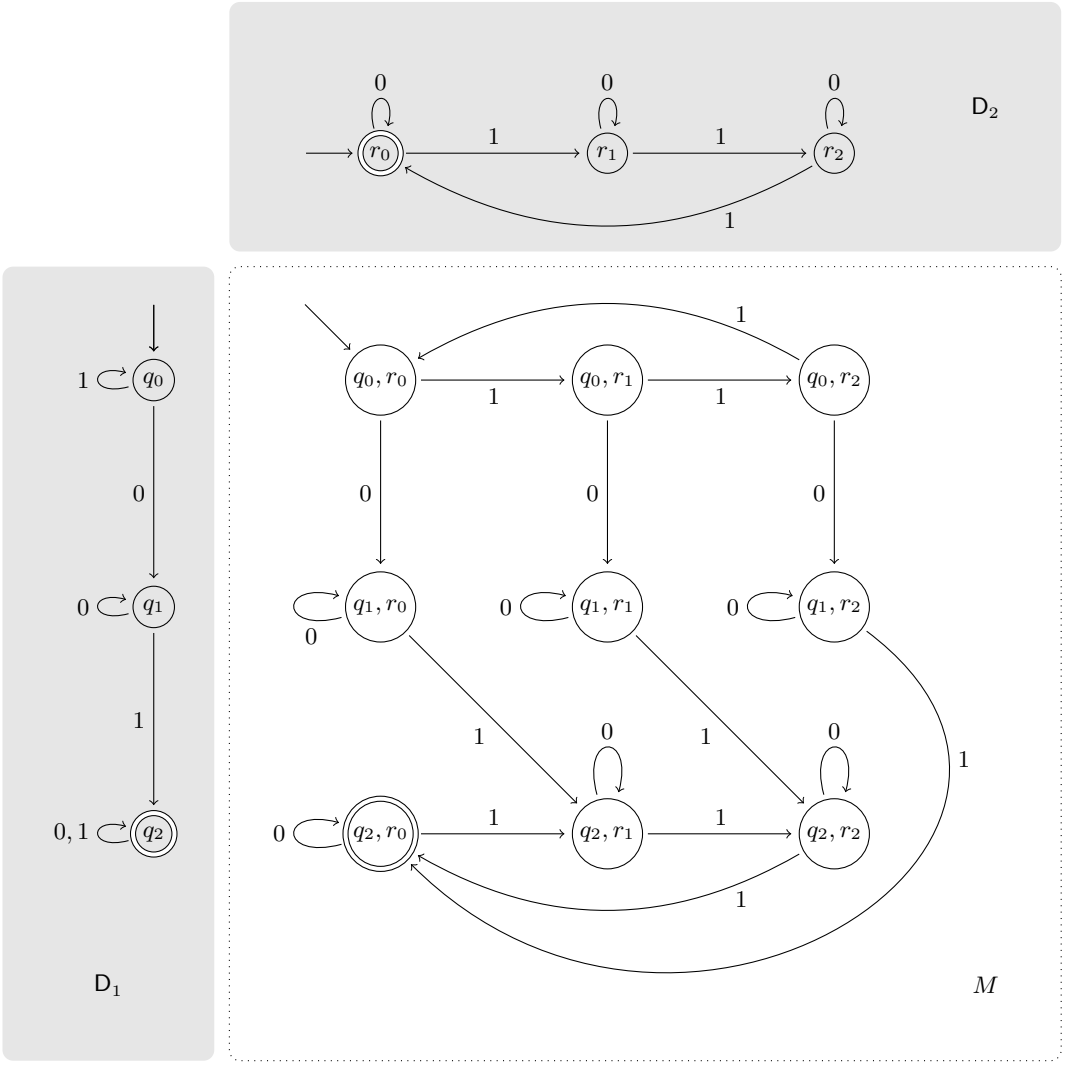
D_2 :



(a) Verwenden Sie das Verfahren aus der Vorlesung um einen DFA $\mathsf{D}_\cap$ anzugeben, so dass $\mathsf{L}(\mathsf{D}_\cap) = \mathsf{L}(\mathsf{D}_1) \cap \mathsf{L}(\mathsf{D}_2)$.
(b) Geben Sie einen DFA $\mathsf{D}_\cup$ an, so dass $\mathsf{L}(\mathsf{D}_\cup) = \mathsf{L}(\mathsf{D}_1) \cup \mathsf{L}(\mathsf{D}_2)$. Sie dürfen Ergebnisse aus Teilaufgabe a) verwenden.
(c) Was müssen Sie ändern, um einen DFA für $\mathsf{L}(\mathsf{D}_1) \setminus \mathsf{L}(\mathsf{D}_2)$ anzugeben? Verallgemeinern Sie Ihre Erkenntnisse für allgemeine binäre Operationen auf Sprachen.

Lösungsskizze

(a) Wir konstruieren den Schnittautomaten aus D_1 und D_2 durch Ausnutzen der Tabellennotation:



(b) Der Unterschied liegt nur in den Endzuständen.
Die Endzustände von $\mathsf{D}_\cup$ sind $\{(q_0, r_0), (q_1, r_0), (q_2, r_0), (q_2, r_1), (q_2, r_2)\}$

(c) Wieder nur die Endzustände ändern, nämlich auf $\{(q_2, r_1), (q_2, r_2)\}$.
Sei $\circ$ eine binäre Operation auf Sprachen, die folgendermaßen definiert ist: $\mathsf{L}_1 \circ \mathsf{L}_2 = \{w \mid w \in \mathsf{L}_1 \odot w \in \mathsf{L}_2\}$, wobei $\odot$ ein binärer boolscher Operator ist (z.B. bei Vereinigung, also $\circ = \cup$, gilt $\odot = \vee$).
Gegeben zwei Automaten D_1 und D_2 konstruiert man den Automaten $\mathsf{D}_\circ$ mit $\mathsf{L}(\mathsf{D}_\circ) = \mathsf{L}(\mathsf{D}_1) \circ \mathsf{L}(\mathsf{D}_2)$, indem man die Produktkonstruktion anwendet und dann die Zustände (q, r) zu Endzuständen macht, für die gilt, dass $q \in \mathsf{F}_1 \odot r \in \mathsf{F}_2$.

AUFGABE 3.7.

Stufe C

Geben Sie eine rekursive Prozedur *empty*(r) an, die für einen gegebenen regulären Ausdruck r entscheidet, ob $L(r) = \emptyset$. Für Ihre Definition sollten Sie das folgende Gerüst verwenden:

- *empty*$(\emptyset) =$
- *empty*$(\mathsf{a}) =$
- *empty*$(\varepsilon) =$
- *empty*$(\alpha\beta) =$
- *empty*$(\alpha \mid \beta) =$
- *empty*$(\alpha^*) =$

Beweisen Sie mittels struktureller Induktion, dass Ihre Definition korrekt ist.

Lösungsskizze

Konstruktion

- $empty(\emptyset) = true$
- $empty(\mathsf{a}) = false$
- $empty(\varepsilon) = false$
- $empty(\alpha\beta) = empty(\alpha) \vee empty(\beta)$
- $empty(\alpha \mid \beta) = empty(\alpha) \wedge empty(\beta)$
- $empty(\alpha^*) = false$

Korrektheit Wir zeigen $\mathsf{L}(r) = \emptyset \iff empty(r)$ mittels struktureller Induktion. Die Fälle für $\emptyset$, a und ε sind trivial. Zudem wissen wir, dass $\varepsilon \in \mathsf{L}(\alpha^*)$ für alle α und damit $\mathsf{L}(\alpha^*) = \emptyset \iff empty(\alpha^*)$. In den Fällen $\alpha\beta$ und $\alpha \mid \beta$ dürfen wir jeweils die Aussage für α und β als Induktionshypothese annehmen, d.h. $\mathsf{L}(\alpha) = \emptyset \iff empty(\alpha)$ und $\mathsf{L}(\beta) = \emptyset \iff empty(\beta)$. Es gilt $\mathsf{L}(\alpha\beta) = \mathsf{L}(\alpha)\mathsf{L}(\beta)$ und

$$\mathsf{L}(\alpha)\mathsf{L}(\beta) = \emptyset \iff \mathsf{L}(\alpha) = \emptyset \vee \mathsf{L}(\beta) = \emptyset \overset{\text{IH}}{\iff} empty(\alpha) \vee empty(\beta) \iff empty(\alpha\beta)$$

Analog gilt $\mathsf{L}(\alpha \mid \beta) = \mathsf{L}(\alpha) \cup \mathsf{L}(\beta)$ und

$$\mathsf{L}(\alpha) \cup \mathsf{L}(\beta) = \emptyset \iff \mathsf{L}(\alpha) = \emptyset \wedge \mathsf{L}(\beta) = \emptyset \overset{\text{IH}}{\iff} empty(\alpha) \wedge empty(\beta) \iff empty(\alpha \mid \beta)$$

Definition (Suffix-Sprache)

Sei $\mathsf{L} \subseteq \Sigma^*$ eine Sprache über dem Alphabet Σ. Wir definieren $\mathsf{L}_{\mathsf{sf}} \coloneqq \{v \in \Sigma^* \mid \exists u \in \Sigma^*.\ uv \in \mathsf{L}\}$ und bezeichnen L_{sf} als *die Sprache der Suffixe* von L.

AUFGABE 3.8.

Stufe D

Sei $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$. Wir zeigen nun, dass wenn L regulär ist, dann ist auch L_{sf} regulär.

(a) Geben Sie die Sprache der Suffixe für $\mathsf{L} = \{\mathsf{abc}, \mathsf{d}\}$ an.
(b) Sei L regulär. Beweisen Sie, dass auch L_{sf} regulär ist, indem Sie aus einem NFA $\mathsf{N} = (\mathsf{Q}, \Sigma, \delta, \mathsf{q}_0, \mathsf{F})$ mit $\mathsf{L} = \mathsf{L}(\mathsf{N})$ einen ε-NFA N' mit $\mathsf{L}_{\mathsf{sf}} = \mathsf{L}(\mathsf{N}')$ angeben.
(c) Geben Sie direkt, d.h. ohne den Umweg über NFAs einen regulären Ausdruck r mit $\mathsf{L}(r) = \mathsf{L}((\mathsf{ab} \mid \mathsf{b})^*\mathsf{cd})_{\mathsf{sf}}$ an.
(d) Beschreiben Sie eine rekursive Prozedur, die einen gegebenen regulären Ausdruck r direkt in einen regulären Ausdruck r' mit $\mathsf{L}(r') = \mathsf{L}(r)_{\mathsf{sf}}$ umschreibt.

Lösungsskizze

(a) $\mathsf{L}_{\mathsf{sf}} = \{\varepsilon, \mathsf{c}, \mathsf{bc}, \mathsf{abc}, \mathsf{d}\}$.
(b) **Konstruktion:** O.B.d.A. nehmen wir an, dass N keine von q_0 unerreichbaren Zustände hat. Sei $\mathsf{q}_0' \notin \mathsf{Q}$ ein neuer Zustand, von dem man zu einer beliebigen Zustand in N springen kann und somit u überspringen kann. Wir definieren:

$$\mathsf{N}' = (\mathsf{Q} \cup \{\mathsf{q}_0'\}, \Sigma, \delta \cup \{(\mathsf{q}_0', \varepsilon, q) \mid q \in \mathsf{Q}\}, \mathsf{q}_0', \mathsf{F})$$

Korrektheit: Um die Gleichheit der beiden Mengen zu zeigen, beweisen wir beide Inklusionen:

(i) $\mathsf{L}_{\mathsf{sf}} \subseteq \mathsf{L}(\mathsf{N}')$: Sei $v \in \mathsf{L}_{\mathsf{sf}}$. Dann existiert ein $u \in \Sigma^*$ mit $uv \in \mathsf{L}$. Somit gibt einen akzeptierenden Lauf auf N:

$$\mathsf{q}_0 \xrightarrow{u^{(1)}} q_1 \dots \xrightarrow{u^{(|u|)}} q_{|u|} \xrightarrow{v^{(1)}} q_{|u|+1} \dots \xrightarrow{v^{(|v|)}} q_{|uv|} \in \mathsf{F}$$

Wir konstruieren nun aus dem akzeptierendem Lauf auf N für uv einen akzeptierenden Lauf auf N' für v, in dem aus dem Zustand q_0' direkt zu $q_{|u|}$ springen:

$$\mathsf{q}_0' \xrightarrow{\varepsilon} q_{|u|} \xrightarrow{v^{(1)}} q_{|u|+1} \dots \xrightarrow{v^{(|v|)}} q_{|uv|} \in \mathsf{F}$$

(ii) $\mathsf{L}_{\mathsf{sf}} \supseteq \mathsf{L}(\mathsf{N}')$: Analog.

(c) $r = \varepsilon \mid \mathsf{d} \mid \mathsf{cd} \mid (\mathsf{ab} \mid \mathsf{b} \mid \varepsilon)(\mathsf{ab} \mid \mathsf{b})^*\mathsf{cd}$
(d) **Konstruktion:** Wir definieren folgende rekursive Prozedur, die reguläre Ausdrücke auf reguläre Ausdrücke abbildet:

- $suff(\emptyset) = \emptyset$
- $suff(\alpha\beta) = \begin{cases} \emptyset & \mathsf{L}(\alpha) = \emptyset \\ suff(\beta) \mid suff(\alpha)\beta & \mathsf{L}(\alpha) \neq \emptyset \end{cases}$
- $suff(\varepsilon) = \varepsilon$
- $suff(\alpha \mid \beta) = suff(\alpha) \mid suff(\beta)$
- $suff(\alpha^*) = \begin{cases} \varepsilon & \mathsf{L}(\alpha) = \emptyset \\ suff(\alpha)\alpha^* & \mathsf{L}(\alpha) \neq \emptyset \end{cases}$
- $suff(\mathsf{a}) = \mathsf{a} \mid \varepsilon$

Um $\mathsf{L}(r) = \emptyset$ zu entscheiden, können wir die Prozedur *empty* von vorhin verwenden.

Korrektheit: Wir zeigen $\mathsf{L}(suff(\gamma)) = (\mathsf{L}(\gamma))_{\mathsf{sf}}$ mit struktureller Induktion über den regulären Ausdruck γ:

$\gamma = \emptyset$: $\mathsf{L}(suff(\emptyset)) = \emptyset = \emptyset_{\mathsf{sf}} = (\mathsf{L}(\emptyset))_{\mathsf{sf}}$

$\gamma = \varepsilon$: $\mathsf{L}(suff(\varepsilon)) = \{\varepsilon\} = \{\varepsilon\}_{\mathsf{sf}} = (\mathsf{L}(\varepsilon))_{\mathsf{sf}}$

$\gamma = \mathsf{a}$: $\mathsf{L}(suff(\mathsf{a})) = \{\varepsilon, \mathsf{a}\} = \{\mathsf{a}\}_{\mathsf{sf}} = (\mathsf{L}(\mathsf{a}))_{\mathsf{sf}}$.

$\gamma = \alpha\beta$: Wir brauchen eine zusätzliche Fallunterscheidung über die Sprache $\mathsf{L}(\alpha)$:

$\mathsf{L}(\alpha) \neq \emptyset$: $\mathsf{L}(suff(\alpha\beta)) = \mathsf{L}(suff(\beta) \mid suff(\alpha)\beta) = \mathsf{L}(suff(\beta)) \cup \mathsf{L}(suff(\alpha))\mathsf{L}(\beta) \overset{\mathsf{IH}}{=} (\mathsf{L}(\beta))_{\mathsf{sf}} \cup (\mathsf{L}(\alpha))_{\mathsf{sf}}\mathsf{L}(\beta) = (\mathsf{L}(\alpha\beta))_{\mathsf{sf}}$

$\mathsf{L}(\alpha) = \emptyset$: $\mathsf{L}(suff(\alpha\beta)) = \mathsf{L}(\emptyset) = \emptyset = \emptyset_{\mathsf{sf}} = \mathsf{L}(\emptyset)_{\mathsf{sf}}$

$\gamma = \alpha \mid \beta$: $\mathsf{L}(suff(\alpha \mid \beta)) = \mathsf{L}(suff(\alpha) \mid suff(\beta)) = \mathsf{L}(suff(\alpha)) \cup \mathsf{L}(suff(\beta)) \overset{\mathsf{IH}}{=} (\mathsf{L}(\alpha))_{\mathsf{sf}} \cup (\mathsf{L}(\beta))_{\mathsf{sf}} = (\mathsf{L}(\alpha \mid \beta))_{\mathsf{sf}}$

$\gamma = \alpha^*$: Wir brauchen eine zusätzliche Fallunterscheidung über die Sprache $\mathsf{L}(\alpha)$:

$\mathsf{L}(\alpha) \neq \emptyset$: $\mathsf{L}(suff(\alpha^*)) = \mathsf{L}(suff(\alpha))\mathsf{L}(\alpha^*) \overset{\mathsf{IH}}{=} (\mathsf{L}(\alpha))_{\mathsf{sf}}\mathsf{L}(\alpha^*) = (\mathsf{L}(\alpha^*))_{\mathsf{sf}}$

$\mathsf{L}(\alpha) = \emptyset$: $\mathsf{L}(suff(\alpha^*)) = \mathsf{L}(\varepsilon) = \{\varepsilon\} = \{\varepsilon\}_{\mathsf{sf}} = (\mathsf{L}(\varepsilon))_{\mathsf{sf}}$

Hinweis: Seien A und B Sprachen über dem Alphabet Σ. Für den Korrektheitsbeweis verwenden wir die folgenden drei Gesetze (deren Beweise wir hier aus Platzgründen nicht zeigen).

- $(\mathsf{A} \cup \mathsf{B})_{\mathsf{sf}} = \mathsf{A}_{\mathsf{sf}} \cup \mathsf{B}_{\mathsf{sf}}$
- $\mathsf{A} \neq \emptyset \Rightarrow (\mathsf{AB})_{\mathsf{sf}} = \mathsf{B}_{\mathsf{sf}} \cup \mathsf{A}_{\mathsf{sf}}\mathsf{B}$
- $\mathsf{A} \neq \emptyset \Rightarrow (\mathsf{A}^*)_{\mathsf{sf}} = \mathsf{A}_{\mathsf{sf}}\mathsf{A}^*$

AUFGABE 3.9.

Stufe E

Wir betrachten das duale Modell zu NFAs und definieren für diese Aufgabe *UFA*s (universelle endliche Automaten), die ein Wort w akzeptieren gdw. alle Läufe auf w in einem Endzustand enden. Sei $\mathsf{U} = (\mathsf{Q}, \Sigma, \delta, \mathsf{q}_0, \mathsf{F})$ ein UFA. Dann wird das Wort $w \in \Sigma^*$ genau dann akzeptiert, wenn $\delta(\mathsf{q}_0, w) \subseteq \mathsf{F}$.

(a) Entscheiden Sie, ob die folgende Aussage gilt und begründen Sie Ihre Antwort, indem Sie einen passenden Beweis oder ein passendes Gegenbeispiel angeben:

Jeder UFA ohne Endzustände akzeptiert die leere Sprache.

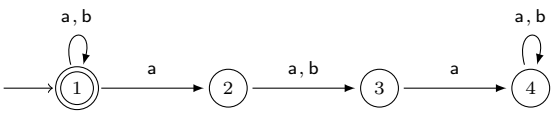
(b) Wir betrachten folgende Sprache $\mathsf{L}_k = \{w \in \Sigma^* \mid \forall 1 \leq i \leq |w| \,.\, w_i = \mathsf{a} \rightarrow w_{i+k} = \mathsf{b}\}$ über dem Alphabet $\Sigma = \{\mathsf{a}, \mathsf{b}\}$. Konstruieren Sie einen NFA und einen UFA für $k = 2$ und $k = 3$. Vergleichen Sie Ihre beiden Lösung bezüglich der Größe der Automaten.

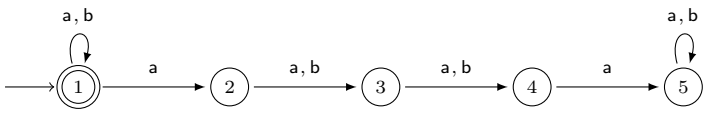
(c) Geben Sie eine Übersetzung von UFAs zu DFAs an und beweisen Sie deren Korrektheit.

Lösungsskizze

(a) Diese Aussage gilt *nicht*. Für einen UFA ohne Endzustände gilt $\mathsf{F} = \emptyset$, d.h. es werden nur Worte akzeptiert, so dass $\delta(\mathsf{q}_0, w) \subseteq \emptyset$. Nach Definition der Transitionsfunktion für NFAs gilt $\delta(\mathsf{q}_0, w) = \emptyset$ gdw. es keinen Lauf für w gibt. Damit kann ein UFA mit $\mathsf{F} = \emptyset$ Wörter akzeptieren, die keinen Lauf im Automaten haben.

(b) Ein UFA, der die Sprache für $k = 2$ bzw. $k = 3$ erkennt, sieht wie folgt aus:





Für alle Wörter, die der Definition der Sprache widersprechen, gibt es einen Lauf, der im Zustand 4 endet. Damit kann der UFA diese Wörter nicht akzeptieren, da *alle* Läufe akzeptierend sein müssen, damit ein UFA ein Wort akzeptiert.

(c) Wir adaptieren die Potenzmengenkonstruktion für NFAs und definieren nur die Endzustände um: $\mathsf{F}' := \{S \in 2^{\mathsf{Q}} \mid S \subseteq \mathsf{F}\}$.