

Einführung in die Theoretische Informatik
Sommersemester 2020 – Übungsblatt Lösungsskizze 2

AUFGABE 2.1. (*Wichtige Begriffe*)

Stufe A

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

- ε -NFA
- Potenzmengenkonstruktion
- rechtslineare Grammatik
- Komplementierung
- Produktkonstruktion
- regulärer Ausdruck
- rekursive Prozedur
- Strukturelle Induktion

AUFGABE 2.2. (*Automata Tutor*)

Stufe B

Lösen Sie die Aufgaben unter “English to Regular Expression” und “Regular Expression to NFA”. Beachten Sie, dass der AutomataTutor automatisch $c|\varepsilon$ als $c?$ abkürzt.

AUFGABE 2.3. (*Tick Tock Boom, Blow Up*)

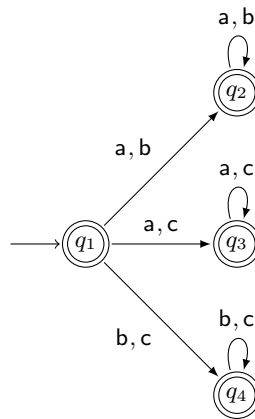
Stufe C

Mit $|w|_x$ bezeichnen wir die Anzahl der Vorkommen des Buchstabens $x \in \Sigma$ in $w \in \Sigma^*$. Sei $\Sigma = \{a, b, c\}$ und $L = \{w \in \Sigma^* \mid \exists x \in \Sigma. |w|_x = 0\}$.

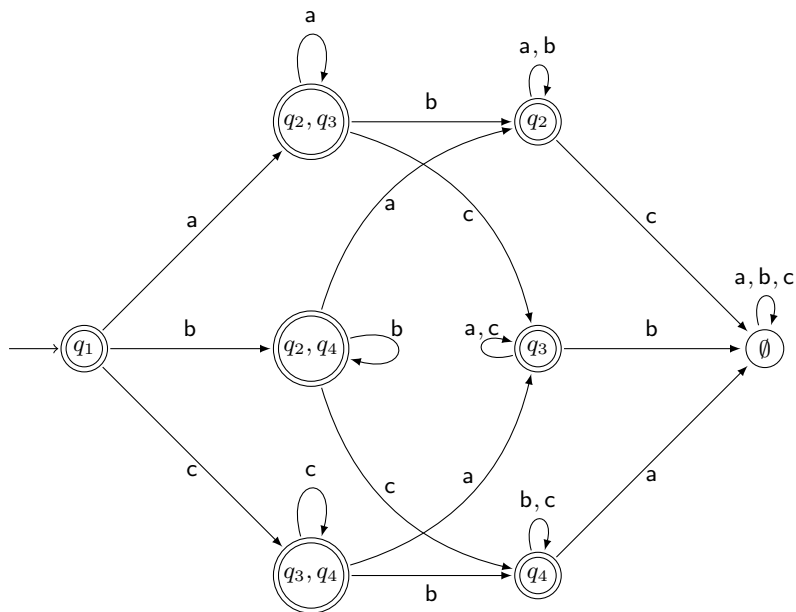
- Konstruieren Sie einen NFA N mit genau 4 Zuständen und $L(N) = L$.
- Determinisieren Sie den NFA N aus (a) mittels der Potenzmengenkonstruktion um einen DFA D mit $L(D) = L(N)$ zu erhalten.
- Geben Sie eine rechtslineare Grammatik G (gemäß Satz 3.13) an, sodass $L(G) = L(D)$.
- Übersetzen Sie die Grammatik G (gemäß Satz 3.14) in einen ε -NFA N' , sodass $L(N') = L(G)$.
- Vergleichen Sie die NFAs N' und N bezüglich Zustands- und Transitionszahl. Diskutieren Sie dann, inwiefern Sie Ihre Beobachtung verallgemeinern können.

Lösungsskizze

(a) Der folgende NFA rät im initialen Zustand welche beiden Buchstaben im Wort auftauchen:



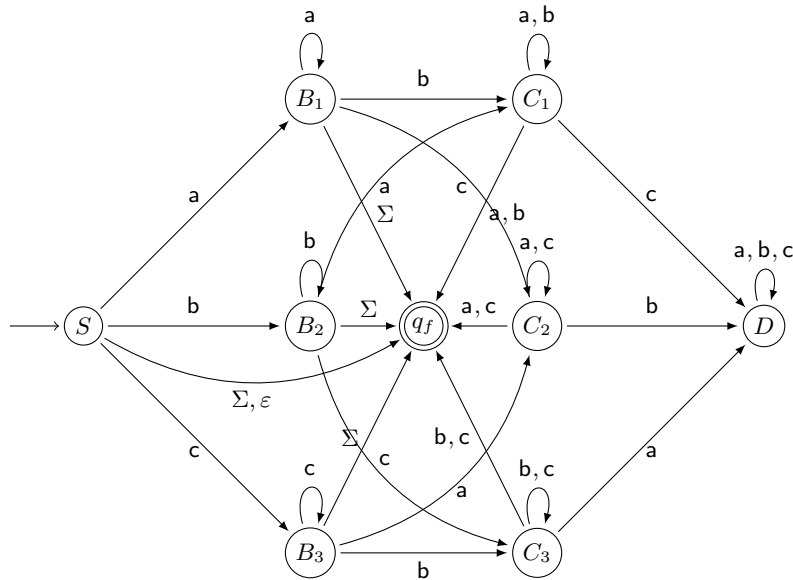
(b) Anwendung der Potenzmengenkonstruktion:



(c) Jeder Zustand entspricht einem Nicht-Terminal, die Terminale sind gerade das Alphabet Σ , der initiale Zustand wird zum Startsymbol. Zur Vereinfachung benennen wir die Zustände von oben nach unten und links nach rechts von A bis C_3 , wie oben gezeichnet. Dann ergibt sich folgende Grammatik

- $S \rightarrow aB_1 \mid bB_2 \mid cB_3 \mid a \mid b \mid c \mid \varepsilon$
- $B_1 \rightarrow a \mid b \mid c \mid aB_1 \mid bC_1 \mid cC_2$
- $B_2 \rightarrow a \mid b \mid c \mid bB_2 \mid aC_1 \mid cC_3$
- $B_3 \rightarrow a \mid b \mid c \mid cB_3 \mid aC_2 \mid bC_3$
- $C_1 \rightarrow a \mid b \mid aC_1 \mid bC_1 \mid cD$
- $C_2 \rightarrow a \mid c \mid aC_2 \mid cC_2 \mid bD$
- $C_3 \rightarrow b \mid c \mid bC_3 \mid cC_3 \mid aD$
- $D \rightarrow aD \mid bD \mid cD$

(d) Wir erhalten folgenden Automaten:



- (e) Diese Übersetzung (NFA \rightarrow DFA \rightarrow Grammatik \rightarrow NFA) induziert eine Normalform, in der es maximal 2 Endzustände gibt. Die Anzahl der Zustände entspricht der Anzahl der Variablen in der Grammatik plus den dedizierten q_f Zustand. Durch die Potenzmengenkonstruktion von N zu D wird die Zustandszahl vergrößert (möglicherweise exponentiell), und nachdem die Grammatik für jeden Zustand des DFA ein Nichtterminal hat, ist die Zustandszahl von N' möglicherweise exponentiell größer als die von N.

AUFGABE 2.4. (I LoVe¹ Induction)

Stufe D

Vervollständigen Sie den Beweis von Satz 3.14 aus der Vorlesung:

Für jede rechtslineare Grammatik G gibt es einen NFA M mit $L(G) = L(M)$.

Lösungsskizze

Sei $G = (V, \Sigma, P, S)$ eine rechtslineare Grammatik und $N = (Q, \Sigma, \delta, S, F)$ der dazu im Satz 3.14 der Vorlesung konstruierte ε -NFA. Nach Definition rechtslinearer Grammatiken ist $(S \rightarrow \varepsilon) \in P$ die einzig mögliche ε -Produktion in P . Somit enthält N eine ε -Kante (S, ε, q_f) genau dann wenn $(S \rightarrow \varepsilon) \in P$.

Damit kann N in einen NFA M überführt werden indem man gegebenenfalls diese ε -Kante entfernt und S zum Endzustand macht. Wir setzen also $M := (Q, \Sigma, \delta', S, F')$ mit $\delta' := \delta \setminus \{(S, \varepsilon, q_f)\}$ und $F' := F \cup \{S \mid (S \rightarrow \varepsilon) \in P\}$. Offensichtlich gilt $L(M) = L(N)$.

Wir müssen noch zeigen, dass $L(M) = L(G)$, was sich wie folgt umschreiben lässt:

$$L(M) = L(G) \iff (\forall w \in \Sigma^*. w \in L(M) \iff w \in L(G))$$

$$\iff (\forall n \in \mathbb{N}. \forall w \in \Sigma^n. w \in L(M) \iff w \in L(G))$$

Letzteres könnten wir nun versuchen, per Induktion über n zu beweisen. In Zukunft werden wir diese Schritte abkürzen und einfach sagen, dass wir die Induktion über die Länge der Wörter $w \in \Sigma^*$ führen. Den Fall $n = 0$ betrachten wir vorweg gesondert – er würde in der Induktion einen Sonderfall auslösen: Es gilt $w = \varepsilon$ und dann

$$\varepsilon \in L(M) \iff S \in F' \stackrel{\text{Def. M}}{\iff} (S \rightarrow \varepsilon) \in P \stackrel{G \text{ rechtslin.}}{\iff} \varepsilon \in L(G).$$

Die Fälle $n > 0$ zeigen wir nun per Induktion. Unser Versuch wird zunächst fehlschlagen, doch aus didaktischen Gründen wird der Fehlschlag hier dokumentiert:

Im Fall $n = 1$ gilt $w = a \in \Sigma$ und somit

$$a \in L(M) \iff \delta'(S, a) \cap F' \neq \emptyset$$

$$\iff (S \rightarrow a) \in P \text{ oder } ((S \rightarrow aS) \in P \text{ und } (S \rightarrow \varepsilon)) \quad (\text{Def. M})$$

$$\iff a \in L(G) \quad (G \text{ rechtslin.}).$$

Im Fall $n + 1$ erhalten wir die Induktionshypothese

$$\forall w \in \Sigma^n. w \in L(M) \iff w \in L(G). \quad (\text{IH}_{\text{fail}})$$

¹LoVe := Logic and Verification

Sei nun $w \in \Sigma^{n+1}$. Dann gibt es $a \in \Sigma$ und $v \in \Sigma^n$ mit $w = av$. Jetzt gilt

$$av \in L(M) \iff \hat{\delta}'(S, av) \cap F' \neq \emptyset \stackrel{\text{Def. } \hat{\delta}'}{\iff} \hat{\delta}'(\delta'(S, a), v) \cap F' \neq \emptyset$$

Nach Induktionshypothese erhalten wir für v :

$$(v \in L(M) \iff v \in L(G)) \iff (\hat{\delta}'(S, v) \cap F' \neq \emptyset \iff v \in L(G)).$$

Die Induktionshypothese sagt also nur, was passiert, wenn wir v beginnend aus S lesen. Allerdings kann ohne Weiteres $\delta'(S, a) \neq \{S\}$ gelten. An diesem Punkt hängen wir fest. Die Induktionshypothese ist zu schwach. Um den Induktionsschritt zu beweisen, muss die Induktionshypothese uns Garantien liefern, wenn wir v aus beliebigen Zustandsmengen $V' \subseteq Q$ lesen. Da $|v| > 0$ und es keine Transitionen aus q_f gibt, reicht es sogar Zustandsmengen $V' \subseteq Q \setminus \{q_f\} = V$ zu betrachten. Analog wird in diesem Fall in der Grammatik v nicht mehr startend von S sondern einem beliebigen Nichtterminal $A \in V$ aus erzeugt.

Wir generalisieren also unsere Aussagen in diese Richtung und beweisen:

$$\forall n \in \mathbb{N}_+. \forall w \in \Sigma^n. \forall V' \subseteq V. \hat{\delta}'(V', w) \cap F' \neq \emptyset \iff \exists A \in V'. A \rightarrow_G^* w$$

Beweis per Induktion über n . Im Fall $n = 1$ gilt $w = a \in \Sigma$ und

$$\begin{aligned} \hat{\delta}'(V', a) \cap F' \neq \emptyset &\iff \exists A \in V'. \delta'(A, a) \cap F' \neq \emptyset \\ &\iff \exists A \in V'. (A \rightarrow a) \in P \text{ oder } ((A \rightarrow aS) \in P \text{ und } (S \rightarrow \varepsilon) \in P) && \text{(Def. } \delta, \delta', F') \\ &\iff \exists A \in V'. A \rightarrow_G^* a && \text{(G rechtslinear)} \end{aligned}$$

Im Fall $n + 1$ erhalten wir die Induktionshypothese

$$\forall w \in \Sigma^n. \forall V' \subseteq V. \hat{\delta}'(V', w) \cap F' \neq \emptyset \iff \exists A \in V'. A \rightarrow_G^* w. \quad \text{(IH)}$$

Wähle nun $a \in \Sigma$ und $v \in \Sigma^n$ mit $w = av$. Wir haben

$$\begin{aligned} \hat{\delta}'(V', av) \cap F' \neq \emptyset &\iff \hat{\delta}'(\delta'(V', a), v) \cap F' \neq \emptyset && \text{(Def. } \hat{\delta}') \\ &\iff \hat{\delta}'(\delta'(V', a) \setminus \{q_f\}, v) \cap F' \neq \emptyset && (v \neq \varepsilon) \\ &\iff \exists A \in \delta'(V', a) \setminus \{q_f\}. A \rightarrow_G^* v && \text{(IH)} \\ &\iff \exists A \in \delta(V', a) \setminus \{q_f\}. A \rightarrow_G^* v && (a \neq \varepsilon) \\ &\iff \exists B \in V'. \exists A \in \delta(V', a) \setminus \{q_f\}. (B \rightarrow aA) \in P \text{ und } A \rightarrow_G^* v && \text{(Def. } \delta) \\ &\iff \exists B \in V'. B \rightarrow_G^* av && \text{(G rechtslinear, } v \neq \varepsilon) \end{aligned}$$

Dabei ist es gerechtfertigt, die IH anzuwenden, da $\delta'(V', a) \setminus \{q_f\} \subseteq V$ wegen der Definition von M . Somit ist der Induktionsbeweis vollständig. Nun können wir die Aussage mit $V' = \{S\}$ spezialisieren, um den Beweis abzuschließen.

AUFGABE 2.5.

Stufe D

Entscheiden Sie, ob die folgenden Aussagen korrekt sind oder nicht und begründen Sie Ihre Behauptung, indem Sie entweder ein Gegenbeispiel oder eine passende Konstruktion angeben.

Für jeden NFA $N = (Q, \Sigma, \delta, q_0, F)$ gibt es einen NFA $N' = (Q', \Sigma', \delta', q'_0, F')$ mit $L(N) = L(N')$ und ...

- der Startzustand hat keine eingehenden Kanten.
- kein Endzustand hat eine ausgehende Kante.
- für jeden Zustand gilt: alle eingehenden Kanten sind mit demselben Zeichen beschriftet.
- für jeden Zustand gilt: alle ausgehenden Kanten sind mit demselben Zeichen beschriftet.

Lösungsskizze

- Ja: Man erstellt vom Startzustand q_0 eine Kopie q'_0 , die nur die ausgehenden Kanten von q_0 erbt. Danach wird q_0 zu einem normalen Zustand, während q'_0 zu einem Startzustand wird. Formal: $N' = (Q', \Sigma', \delta', q'_0, F')$, wobei

- $Q' = Q \uplus \{q'_0\}$,
- $\delta' = \delta \uplus \{(q'_0, x, q) \mid (q_0, x, q) \in \delta\}$
- $F' = F \cup \{q'_0 \mid q_0 \in F\}$
- $\Sigma' = \Sigma$

- Nein: Betrachte einen NFA N für $L = \{\varepsilon, a\}$.

- Ja: man spaltet jeden Zustand $q \in Q$ nach dem Buchstaben der eingehenden Kanten auf, d.h. aus q macht man die Zustände (q, x) für jedes $x \in \Sigma$.

Danach setzt man $\delta'((q, x), y, (q', y)) := \delta(q, y, q')$.

Wird dabei der Startzustand aufgespalten wählt man einen beliebigen daraus hervorgehenden Zustand als neuen Startzustand.

(d) Nein: Betrachte einen NFA N mit $L(N) = \{a, b\} = \Sigma$. Dann gilt $\delta(q_0, a) \cap F \neq \emptyset$ und $\delta(q_0, b) \cap F \neq \emptyset$.

AUFGABE 2.6. (*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 $c|\epsilon$ 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 $\Sigma = \{a, b, c\}$ und für die nächsten 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.
- (f) **Knobelaufgabe:** Sei $\Sigma = \{a, b\}$. Wörter, die 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

- (a) $((a | b | c)(a | b | c))^*$
- (b) $a | b | a(a | b | c)^*a | b(a | b | c)^*b$
- (c) $((a^* | b^*)cc^*)^*(a^* | b^*)$
- (d) $(0 | 1)^*0$
- (e) $(0 | 1)^*(1 | 10)$
- (f) $\epsilon | (a^+b^+)^*a^+ | (b^+a^+)^*b^+$

AUFGABE 2.7. (*Draw Me Like One of Your Deterministic Automata*)

Stufe C

- (a) Geben Sie für jeden Teilausdruck des regulären Ausdrucks $((a\emptyset)^*b | ab)^*$ 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.

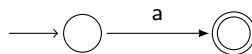
- (b) Überführen Sie den ϵ -NFA für den gesamten Ausdruck aus Teilaufgabe a) in einen DFA. Kombinieren Sie hierzu die Potenzmengenkonstruktion mit der Umwandlung eines ϵ -NFA in einen NFA. (Vorlesung Folie 47)

Lösungsskizze

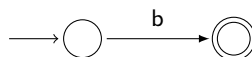
- (a) • \emptyset :



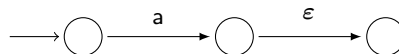
- a :



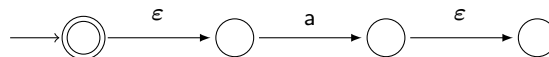
- b :



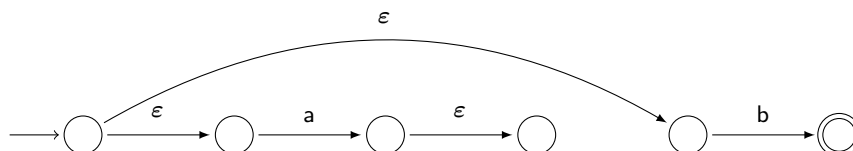
- $a\emptyset$:



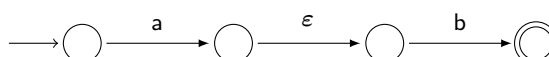
- $(a\emptyset)^*$:



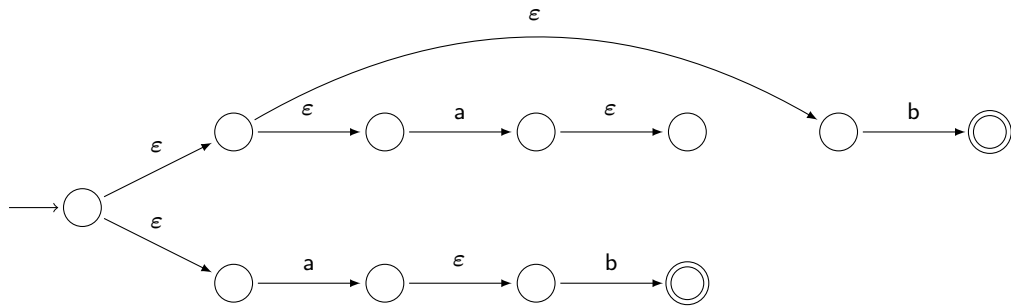
- $(a\emptyset)^*b$:



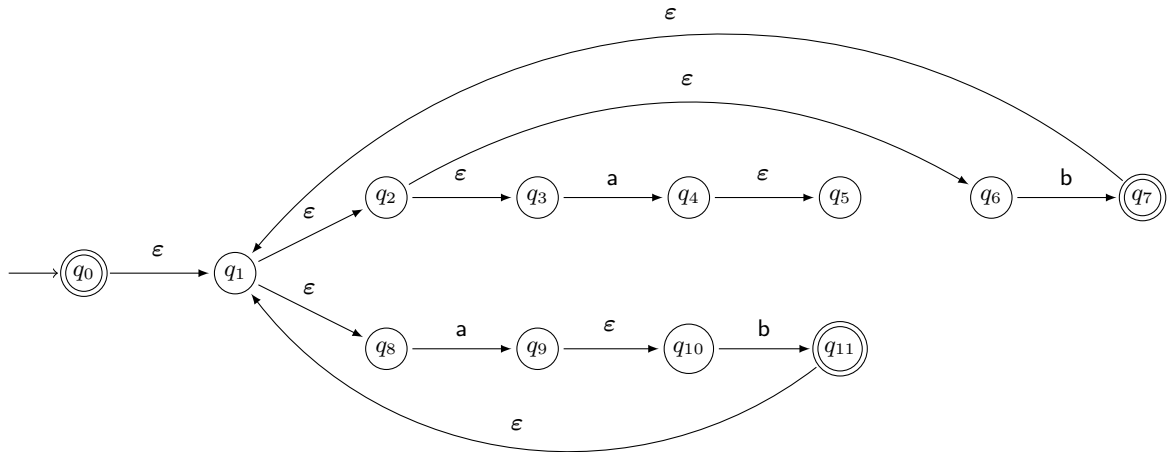
- ab :



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



• $((a\emptyset)^*b \mid ab)^*$:

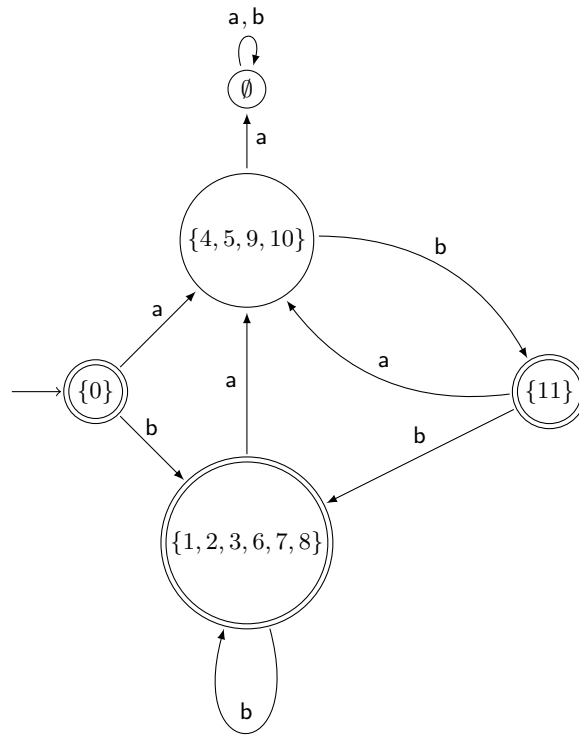


(b) Wir konstruieren zuerst einen NFA für $((a\emptyset)^*b \mid ab)^*$ indem wir die ε -Transitionen ersetzen. Aufgrund der vielen Transitionen geben wir den Automaten in Tabellennotation an:

	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 2.8.

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:

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

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

Lösungsskizze

Konstruktion

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

Korrektheit Wir zeigen $L(r) = \emptyset \iff empty(r)$ mittels struktureller Induktion.

Fall $r = \emptyset$, $r = a$, $r = \varepsilon$. Trivial.

Fall $r = \alpha^*$. Wir haben $\varepsilon \in L(\alpha^*) \neq \emptyset \iff \neg empty(\alpha^*)$. Die Aussage gilt per Definition von $empty$.

Fall $r = \alpha\beta$. Als Induktionshypothesen erhalten wir $L(\alpha) = \emptyset \iff empty(\alpha)$ und $L(\beta) = \emptyset \iff empty(\beta)$.

Es gilt $L(\alpha\beta) = \emptyset \iff L(\alpha)L(\beta) = \emptyset \iff L(\alpha) = \emptyset \vee L(\beta) = \emptyset \stackrel{I.H.}{\iff} empty(\alpha) \vee empty(\beta) = empty(\alpha\beta)$.

Fall $r = \alpha \mid \beta$. Es gelten dieselben Induktionshypothesen wie im vorherigen Fall. Wir haben $L(\alpha \mid \beta) = \emptyset \iff L(\alpha) \cup L(\beta) = \emptyset \iff L(\alpha) = \emptyset \wedge L(\beta) = \emptyset \stackrel{I.H.}{\iff} empty(\alpha) \wedge empty(\beta) = empty(\alpha \mid \beta)$.