Prof. J. Esparza
S. Sickert, J. Krämer

Technische Universität München
KEINE ABGABE

Einführung in die theoretische Informatik

Sommersemester 2017 – Übungsblatt Lösungsskizze 2

Übungsblatt

Wir unterscheiden zwischen Übungs- und Abgabeblättern. Auf diesem *Übungsblatt* finden Sie eine Übersicht über die Kernaspekte, die Sie in Kalenderwoche 19 in den Tutorien diskutieren, üben und vertiefen. Die Aufgaben auf diesem Blatt dienen dem Üben und Verstehen des Vorlesungsstoffes, sowie dem *eigenständigen Erarbeiten* der Kernaspekte. Außerdem sollen Ihnen diese Aufgaben auch helfen, ein Gefühl dafür zu bekommen, was Sie inhaltlich in der Klausur erwartet. Klausuraufgaben können jedoch deutlich von den hier gestellten Aufgaben abweichen. Abschreiben und Auswendiglernen von Lösungen wird Ihnen daher keinen dauerhaften Erfolg in der Vorlesung bringen. Fragen zu den Übungsblättern können Sie montags bis donnerstags von 12 Uhr bis 14 Uhr in der *THEO-Sprechstunde* in Raum 03.11.034 stellen.

Kernaspekte

K2.1 folgende Definitionen korrekt wiedergeben

- DFA
- NFA
- akzeptierte Sprache
- Zustandsgraph
- Akeptanzbedingung von DFAs/NFAs

K2.2 die Potenzmengenkonstruktion in eigenen Worten beschreiben
K2.3 Beispiele angeben, die zeigen, dass verschiedene Arten von Automaten unterschiedlich kompakte Darstellungen von Sprachen erlauben
K2.4 für eine natürlich-sprachlich beschriebene Sprache L einen DFA/NFA A angeben, so dass $\mathsf{L}(\mathsf{A}) = \mathsf{L}$
K2.5 eine rechtlineare Grammatik G in einen NFA N übersetzen, so dass $\mathsf{L}(\mathsf{G}) = \mathsf{L}(\mathsf{N})$
K2.6 einen NFA N in einen DFA D mit Potenzmengenkonstruktion übersetzen, so dass $\mathsf{L}(\mathsf{N}) = \mathsf{L}(\mathsf{D})$
K2.7 einen DFA D in eine rechtslineare Grammatik G übersetzen, so dass $\mathsf{L}(\mathsf{G}) = \mathsf{L}(\mathsf{D})$
K2.8 Aussagen über die Ausdrucksfähigkeit verschiedener Automatenmodelle beweisen oder widerlegen
K2.9 begründet entscheiden, ob gegebene Beispiele neu eingeführte Definitionen erfüllen
K2.10 Aussagen, mit neu eingeführten Definitionen beweisen oder widerlegen

AUFGABE 2.1.

Stufe B

Geben Sie für jede der folgenden Sprachen über dem Alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ einen NFA an, der genau die jeweilige Sprache erkennt:

(a) Alle Wörter, die `aaab` enthalten.
(b) Alle Wörter, deren 3-letzter Buchstabe ein `a` ist, z.B. `babb`. Der Automat sollte nicht mehr als 4 Zustände haben.

Vergleichen Sie Ihre NFAs mit den entsprechenden DFAs aus Aufgabe 2.2

Lösungsskizze

Die Lösungsskizzen finden Sie unter `https://www7.in.tum.de/um/courses/theo/ss2017/exercises/theo17-02-lsg.zip`.

AUFGABE 2.2.

Stufe B

Geben Sie für jede der folgenden Sprachen über dem Alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ einen DFA (graphisch) an, der genau die jeweilige Sprache erkennt:

(a) Alle Wörter, die mit einem `b` beginnen.
(b) Alle Wörter gerader Länge.
(c) Alle Wörter ungerader Länge, die auf ein `c` enden.
(d) Die Sprache, die nur das leere Wort enthält.
(e) Alle Wörter, die `aab` enthalten.
(f) Alle Wörter, die eine durch drei teilbare Anzahl von `c` enthalten.
(g) $\mathsf{L} = \{\mathtt{aabcaa}, \mathtt{aacaa}, \mathtt{baa}\}$
(h) Alle Wörter, deren 3-letzter Buchstabe ein `a` ist, z.B. `babb`.

Beschreiben Sie dann in eigenen Worten, wie man im Allgemeinen einen Automaten konstruiert, der eine Sprache erkennt, die ...

- am Anfang/Ende jedes Wortes eine bestimmte Sequenz von Buchstaben fordert
- nur Worte gerader/ungerader Länge enthält
- von jedem Wort verlangt, eine bestimmte Anzahl an Buchstaben zu enthalten
- die nur Worte enthält, die eine bestimmte Sequenz von Buchstaben enthält
- deren Worte an einer fixierten Position einen eigenen bestimmten Buchstaben haben müssen.

Lösungsskizze
Die Lösungenskizzen finden Sie unter `https://www7.in.tum.de/um/courses/theo/ss2017/exercises/theo17-02-lsg.zip`.

AUFGABE 2.3.

Stufe B / C

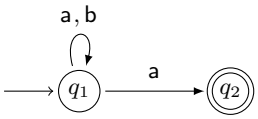
Sei $\Sigma = \{\mathsf{a}, \mathsf{b}\}$ und $\mathsf{A}_k = \{w_1 \dots w_l \in \Sigma^* \mid w_{l-k} = \mathsf{a}\}$ die Sprache aller Wörter über Σ, die an der $(k+1)$-ten Stelle von rechts ein a stehen haben. Insbesondere ist A_0 die Menge aller Wörter, die auf a enden. *Versuchen Sie in beiden Aufgabenteilen NFAs und DFAs mit möglichst wenigen Zuständen anzugeben.*

(a) Geben Sie jeweils einen NFA für A_0, A_1 und A_2 an.
(b) Geben Sie jeweils einen DFA für A_0 und A_1 an.
(c) Beschreiben Sie in eigenen Worten, wie der DFA A_n und der NFA A_n für beliebige $n \in \mathbb{N}$ aussehen.
(d) Beurteilen Sie die folgende Aussage: *Es gibt einen NFA für A_n mit $O(n)$-vielen Zuständen, aber jeder DFA zu A_n hat mindestens $O(2^n)$-viele Zustände.*

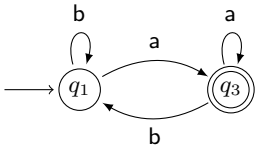
Hinweis: Bearbeiten Sie diese Aufgabe auch für die Sprache $\mathsf{B}_n := \{w \in \Sigma^* \mid \exists i : w_i = w_{i+n}\}$, um die aus der Vorlesung bekannten Konzepte auf eine neue Sprache anzuwenden (A_k findet sich bereits in den Vorlesungsfolien!).

Lösungsskizze

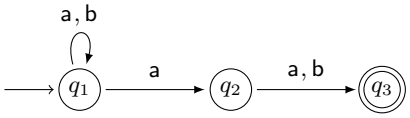
A.) Für die Sprache A_k
(a),(b) Die NFAs auf der linken und die DFAs im Vergleich dazu auf der rechten Seite:



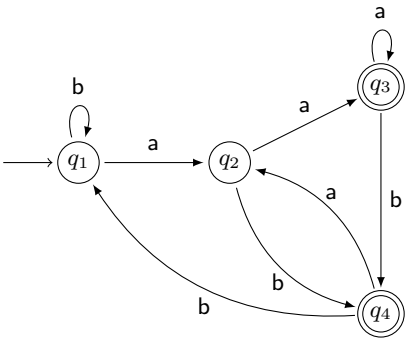
(a) NFA für A_0



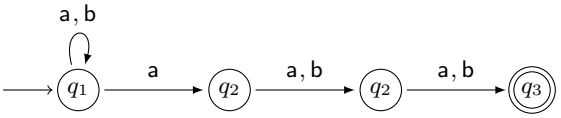
(b) DFA für A_0



(c) NFA für A_1



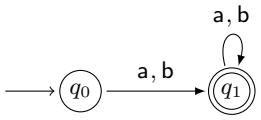
(d) DFA für A_1



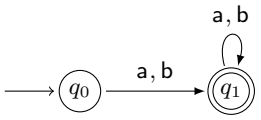
(e) NFA für A_2

(c) Der NFA rät, wann das das k-letzte Zeichen gelesen wird, der DFA muss sich merken, wann er das letzte a gesehen hat und muss für jedes a das er liest nach k Schritten in einem akzeptierenden Zustand sein und für jeden Buchstaben außer a k Schritte später in einem nicht-akzeptierenden Zustand.
(d) Diese Aussage gilt, der Beweis dazu findet sich in den Vorlesungsfolien.

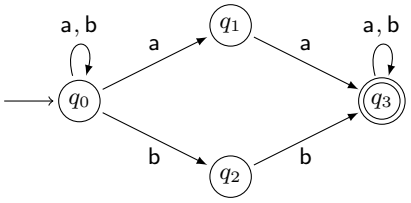
B.) Für die Sprache B_k
(a),(b) Die NFAs auf der linken und die DFAs im Vergleich dazu auf der rechten Seite:
(c) Der NFA rät, ob jetzt i-te Zeichen gelesen wird, welches die Bedingung erfüllt. Der DFA muss sich hingegen immer merken, was er vor n Zeichen gelesen hat, um zu überprüfen, ob die Bedingung $w_i = w_{i+n}$ erfüllt ist.
(d) Der Beweis ist ähnlich zu dem Beweis in den Vorlesungsfolien.



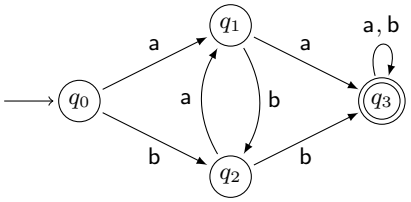
(a) NFA für B_0



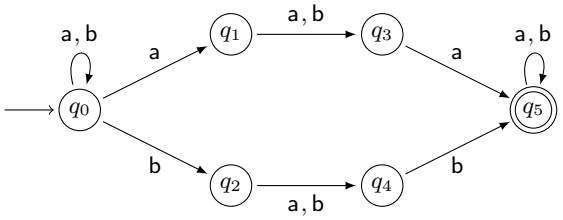
(b) DFA für B_0



(c) NFA für B_1



(d) DFA für B_1



(e) NFA für B_2

AUFGABE 2.4. Stufe C

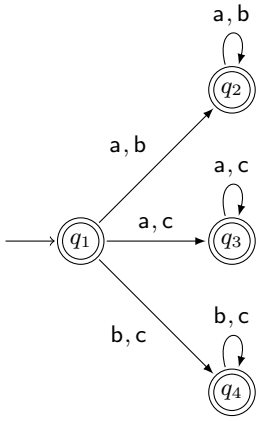
Sei $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$ und $\mathsf{L} = \{w \in \Sigma^* \mid \exists x \in \Sigma.\ |w|_x = 0\}$.

(a) Konstruieren Sie einen NFA N mit genau 4 Zuständen und $\mathsf{L}(\mathsf{N}) = \mathsf{L}$.
(b) Determinisieren Sie den NFA N aus (a) mittels der Potenzmengenkonstruktion.
(c) Geben Sie eine rechtlineare Grammatik G (gemäß Satz 3.12) an, so dass $\mathsf{L}(\mathsf{G}) = \mathsf{L}(\mathsf{D})$.
(d) Übersetzen Sie die Grammatik G (gemäß Satz 3.8) in einen NFA N', so dass $\mathsf{L}(\mathsf{G}) = \mathsf{L}(\mathsf{N}')$.
(e) Vergleichen Sie die NFAs N' und N bezüglich Zustands- und Transitionszahl. Diskutieren Sie dann, inwiefern Sie Ihre Beobachtung verallgemeinern können.

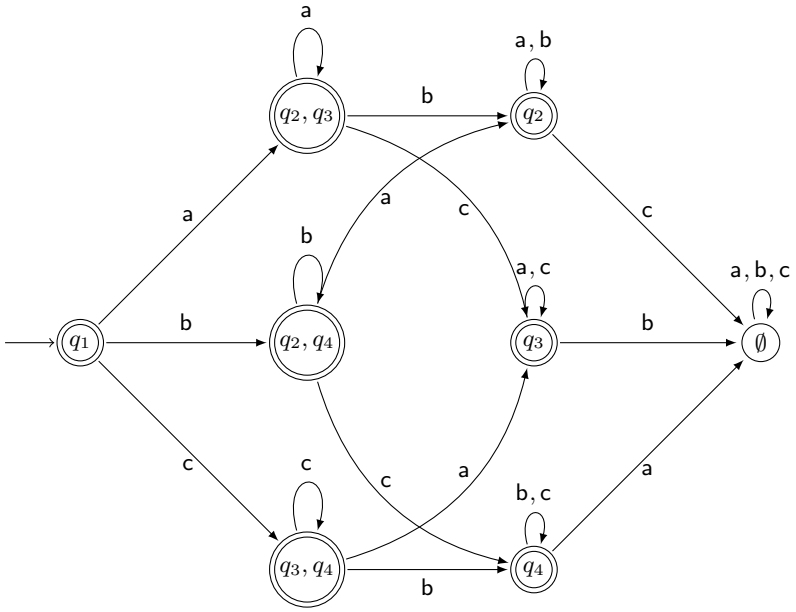
Hinweis: Wir bezeichnen mit $|w|_x$ die Anzahl der Vorkommen des Buchstabens $x \in \Sigma$ in $w \in \Sigma^*$.

Lösungsskizze

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



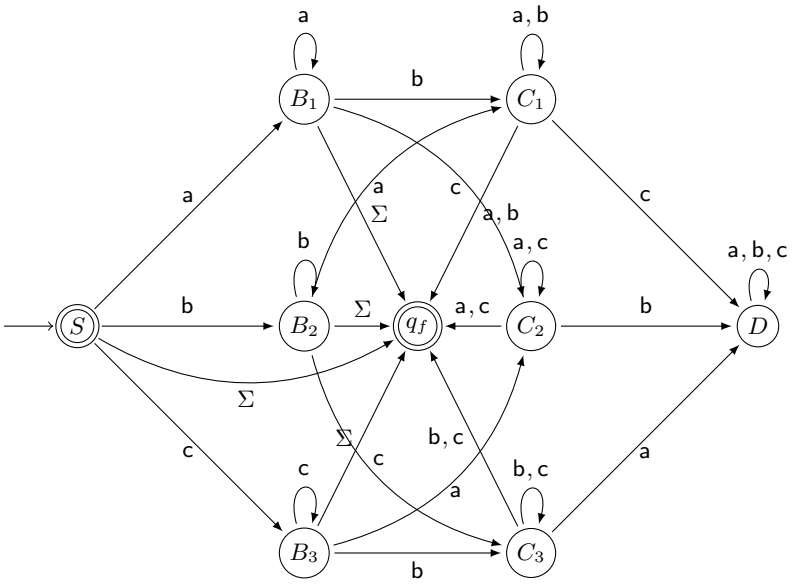
(b) Anwendung der Potenzmengenkonstruktion:



(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

- $\mathsf{S} \to \mathsf{aB}_1 \mid \mathsf{bB}_2 \mid \mathsf{cB}_3 \mid \mathsf{a} \mid \mathsf{b} \mid \mathsf{c} \mid \varepsilon$
- $\mathsf{B}_1 \to \mathsf{a} \mid \mathsf{b} \mid \mathsf{c} \mid \mathsf{aB}_1 \mid \mathsf{bC}_1 \mid \mathsf{cC}_2$
- $\mathsf{B}_2 \to \mathsf{a} \mid \mathsf{b} \mid \mathsf{c} \mid \mathsf{bB}_2 \mid \mathsf{aC}_1 \mid \mathsf{cC}_3$
- $\mathsf{B}_3 \to \mathsf{a} \mid \mathsf{b} \mid \mathsf{c} \mid \mathsf{cB}_3 \mid \mathsf{aC}_2 \mid \mathsf{bC}_3$
- $\mathsf{C}_1 \to \mathsf{a} \mid \mathsf{b} \mid \mathsf{aC}_1 \mid \mathsf{bC}_1 \mid \mathsf{cD}$
- $\mathsf{C}_2 \to \mathsf{a} \mid \mathsf{c} \mid \mathsf{aC}_2 \mid \mathsf{cC}_2 \mid \mathsf{bD}$
- $\mathsf{C}_3 \to \mathsf{b} \mid \mathsf{c} \mid \mathsf{bC}_3 \mid \mathsf{cC}_3 \mid \mathsf{aD}$
- $\mathsf{D} \to \mathsf{aD} \mid \mathsf{bD} \mid \mathsf{cD}$

(d) Wir erhalten folgenden Automaten:



(e) Diese Übersetzung induziert eine Normalform, in der es maximal 2 Endzustände gibt. Die Anzahl der Zustände entspricht die Anzahl der Variablen in der Grammatik (plus den dedizierten q_f Zustand).

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 einen passenden Beweis angeben.

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

(a) der Startzustand hat keine eingehenden Kanten.
(b) kein Endzustand hat eine ausgehende Kante.
(c) für jeden Zustand gilt: alle eingehenden Kanten sind mit demselben Zeichen beschriftet.
(d) für jeden Zustand gilt: alle ausgehenden Kanten sind mit demselben Zeichen beschriftet.

Lösungsskizze

(a) 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: $\mathsf{N}' = (\mathsf{Q}', \Sigma', \delta', \mathsf{q}_0', \mathsf{F}')$, wobei
- $\mathsf{Q}' = \mathsf{Q} \uplus \{\mathsf{q}_0'\}$,
- $\delta' = \delta \uplus \{(\mathsf{q}_0', x, q) \mid (\mathsf{q}_0, x, q) \in \delta\}$
- $\mathsf{F}' = \mathsf{F} \cup \{\mathsf{q}_0' \mid \mathsf{q}_0 \in \mathsf{F}\}$
- $\Sigma' = \Sigma$

(b) Nein: Betrachte einen NFA N für $\mathsf{L} = \{\varepsilon, \mathsf{a}\}$.
(c) Ja: man spaltet jeden Zustand $q \in \mathsf{Q}$ 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 $\delta'((q, x), y, (q', y)) := \delta(q, y, q')$.
(d) Nein: Betrachte einen NFA N mit $\mathsf{L}(\mathsf{N}) = \{\mathsf{a}, \mathsf{b}\} = \Sigma$. Dann gilt $\delta(\mathsf{q}_0, \mathsf{a}) \cap \mathsf{F} \neq \emptyset$ und $\delta(\mathsf{q}_0, \mathsf{b}) \cap \mathsf{F} \neq \emptyset$.

AUFGABE 2.6. Stufe D

Wir erweitern NFAs um die Möglichkeit, mehrere Startzustände $\mathsf{I} \subseteq \mathsf{Q}$ zu besitzen. Ein NFA $\mathsf{N} = (\mathsf{Q}, \Sigma, \delta, \mathsf{I}, \mathsf{F})$ akzeptiert ein Wort $w \in \Sigma^*$ genau dann, wenn

$$\emptyset \neq \mathsf{F} \cap \bigcup_{\mathsf{q}_0 \in \mathsf{I}} \hat{\delta}(\mathsf{q}_0, w),$$

d.h. wenn es mindestens einen akzeptierenden Ablauf von irgendeinem Startzustand aus gibt.

(a) Geben Sie einen NFA $\mathsf{N} = (\mathsf{Q}, \Sigma, \delta, \mathsf{q}_0, \mathsf{F})$ und einen NFA $\mathsf{N}' = (\mathsf{Q}', \Sigma', \delta', \mathsf{I}', \mathsf{F}')$ an, die die Sprache L aus Aufgabe 2.4 erkennen, aber $|\mathsf{Q}'| < |\mathsf{Q}|$.
(b) Entscheiden Sie, ob die folgenden Aussagen korrekt sind oder nicht und begründen Sie Ihre Behauptung, indem Sie entweder ein Gegenbeispiel oder einen passenden Beweis angeben.
Für jeden NFA $\mathsf{N} = (\mathsf{Q}, \Sigma, \delta, \mathsf{I}, \mathsf{F})$ existiert ein NFA $\mathsf{N}' = (\mathsf{Q}', \Sigma', \delta', \mathsf{I}', \mathsf{F}')$ mit $\mathsf{L}(\mathsf{N}) = \mathsf{L}(\mathsf{N}')$, so dass
  (i) $|\mathsf{I}'| = 1$.
  (ii) $|\mathsf{F}'| = 1$.
  (iii) $|\mathsf{I}'| = 1$ und $|\mathsf{F}'| = 1$.

Lösungsskizze

(a) Einen NFA für diese Sprache findet sich in der Lösung zu Aufgabe 2.4. Mit einer Menge von Initialzuständen statt nur einem einzelnen können wir die Zustände q_2, q_3 und q_4 als initial markieren und q_1 weglassen. Damit haben wir separate Initialzustände, um die Sprachen $\{\mathsf{a},\mathsf{b}\}^*$, $\{\mathsf{b},\mathsf{c}\}^*$ und $\{\mathsf{a},\mathsf{c}\}^*$ zu erkennen.

(b) (i) *Kurzversion*: Ja, z.B. Potenzmengenkonstruktion einfach auf den von I aus erreichbaren Teil anwenden. Resultierender DFA N' akzeptiert dieselbe Sprache wie N und jeder DFA ist gleichzeitig auch ein NFA.
Lange Version: Man muss erstmal zeigen, dass ein NFA mit mehreren Startzuständen immer noch eine reguläre Sprache akzeptiert. Das ist, siehe Kurzversion, ziemlich offensichtlich, aber beweisen muss man es trotzdem, daher:
Potenzmengenkonstruktion auf den NFA N anwenden. **GANZ WICHTIG:** intelligent die Konstruktion anwenden, d.h. Tiefen- oder Breitensuchen ausgehend von I ausführen; **NIE**, **NIEMALS** und nochmal **NIE** alle 2^{Q} möglichen Teilmengen konstruieren und dann fröhlich Pfeilchen dazwischen zeichnen.
Sei also $\mathsf{N}' = (\mathsf{Q}', \Sigma', \delta', \mathsf{q}_0', \mathsf{F}')$ mit
- $\mathsf{Q}' \subseteq 2^{\mathsf{Q}}$,
- $\delta' \colon \mathsf{Q}' \times \Sigma \to \mathsf{Q}'$, $\mathsf{I}' = \mathsf{I}$ und
- $\mathsf{F}' = \{S \in \mathsf{Q}' \mid S \cap \mathsf{F} \neq \emptyset\}$,

wobei Q' und δ' die kleinsten Mengen[1] sind, welche folgenden beiden Bedingungen erfüllen:
(1) $\mathsf{I} \in \mathsf{Q}'$
(2) Falls $S \in \mathsf{Q}'$, dann auch $\bigcup_{q \in S} \delta(q,a) =: S' \in \mathsf{Q}'$ und $\delta'(S,a) \in S'$ für jedes $a \in \Sigma$.
Analog zur Vorlesung zeigt man nun, dass $\delta'(\mathsf{I}, w) \in \mathsf{F}'$ gdw. $\exists \mathsf{q}_0 \in \mathsf{Q} \colon \delta(\mathsf{q}_0, w) \cap \mathsf{F} \neq \emptyset$:
Falls $w \in \mathsf{L}(\mathsf{N}')$, dann

$$\mathsf{I} \xrightarrow{w_1} q_1 \xrightarrow{w_2} q_2 \xrightarrow{w_3} \dots \xrightarrow{w_l} S_l$$

mit $S_l \in \mathsf{F}'$, also nach Def. F': $S_l \cap \mathsf{F} \neq \emptyset$. Sei $q_l \in S_l \cap \mathsf{F}$. Dann gibt es nach Def. von δ' ein $q_{l-1} \in S_{l-1}$, sodass $q_l \in \delta(q_{l-1}, w_l)$. Induktiv findet so für alle $0 \leq i \leq l$ ein $q_i \in S_i$ mit $q_{i+1} \in \delta(q_i, w_i)$, womit $w \in \mathsf{L}(\mathsf{N})$ folgt.
Falls $w \in \mathsf{L}(\mathsf{N})$ folgt vollkommen analog, dass auch $w \in \mathsf{L}(\mathsf{N}')$, einfach die obige Argumentation rückwärts lesen.

(ii) **Konstruktion:** Aus dem NFA $\mathsf{N} = (\mathsf{Q}, \Sigma, \delta, \mathsf{I}, \mathsf{F})$ für L konstruieren wir einen NFA $\mathsf{N}' = (\mathsf{Q}', \Sigma', \delta', \mathsf{I}', \mathsf{F}')$ mit
- $\mathsf{Q}' = \mathsf{Q} \cup \{q_f\}$,
- $\mathsf{F}' = \{q_f\}$,
- $\Sigma' = \Sigma$,
- $\delta'(q,a) = \delta(q,a) \cup \{q_f \mid \delta(q,a) \cap \mathsf{F} \neq \emptyset\}$ für alle $a \in \Sigma$ und $q \in \mathsf{Q}$,
- $\delta'(q_f, a) = \emptyset$ für alle $a \in \Sigma$ und
- $\mathsf{I}' = \mathsf{I} \cup \{q_f \mid \mathsf{I} \cap \mathsf{F} \neq \emptyset\}$

Nach Konstruktion ist der neue Zustand q_f der einzige Endzustand von N'.

$\mathsf{L}(\mathsf{N}) \subseteq \mathsf{L}(\mathsf{N}')$: Falls $\varepsilon \in \mathsf{L}(\mathsf{N})$, dann auch $\varepsilon \in \mathsf{L}(\mathsf{N}')$, da q_f dann nach Konstruktion ein Startzustand ist.
Sei daher $w \in \mathsf{L}(\mathsf{N}) \setminus \{\varepsilon\}$. Dann existiert eine akzeptierende Berechnung

$$\mathsf{q}_0 \xrightarrow{w_1} q_1 \xrightarrow{w_2} q_2 \xrightarrow{w_3} \dots \xrightarrow{w_{l-1}} q_{l-1} \xrightarrow{w_l} q_l$$

mit $q_l \in \mathsf{F}$ und $q_{i+1} \in \delta(q_i, w_{i+1})$ und $\mathsf{q}_0 \in \mathsf{I}$ für $w = w_1 \dots w_l$ mit $w_i \in \Sigma$.
Nach Konstruktion existiert in N' damit die akzeptierende Berechnung

$$\mathsf{q}_0 \xrightarrow{w_1} q_1 \xrightarrow{w_2} q_2 \xrightarrow{w_3} \dots \xrightarrow{w_{l-1}} q_{l-1} \xrightarrow{w_l} q_f$$

d.h. $w \in \mathsf{L}(\mathsf{N}')$.

$\mathsf{L}(\mathsf{N}') \subseteq \mathsf{L}(\mathsf{N})$: Sei nun $\varepsilon \in \mathsf{L}(\mathsf{N}')$: Damit muss q_f, da einziger Endzustand, auch Startzustand sein, was nach Konstruktion bedeutet, dass $\varepsilon \in \mathsf{L}(\mathsf{N})$.
Sei daher $w \in \mathsf{L}(\mathsf{N}') \setminus \{\varepsilon\}$, d.h. es existiert eine akzeptierende Berechnung

$$\mathsf{q}_0 \xrightarrow{w_1} q_1 \xrightarrow{w_2} q_2 \xrightarrow{w_3} \dots \xrightarrow{w_{l-1}} q_{l-1} \xrightarrow{w_l} q_f.$$

Nach Konstruktion (1. Schritt) gilt somit $\delta(q_{l-1}, a) \cap \mathsf{F} \neq \emptyset$, womit sich die Berechnung auch zu einem akzeptierenden Berechnung in N übersetzen lässt.

(iii) Nein. Sei $\mathsf{L} = \{\varepsilon, \mathsf{a}\}$. Angenommen es existiert sein NFA $\mathsf{N} = (\mathsf{Q}, \Sigma, \delta, \mathsf{q}_0, \mathsf{F})$ für L mit $\mathsf{F} = \{q_f\}$. Da $\varepsilon \in \mathsf{L}$ und N ein NFA, gilt $\mathsf{q}_0 = q_f$. Da $\mathsf{a} \in \mathsf{L}$, gilt somit $\delta(\mathsf{q}_0, \mathsf{a}) = q_f = \mathsf{q}_0$. Daher gibt es eine a-Schleife am Zustand q_0 und somit auch $\mathsf{aa} \in \mathsf{L}(\mathsf{N}) = \mathsf{L}$. Widerspruch.

[1] Die "kleinsten" Mengen hier formal zu definieren ist technisch aufwendig, aber an dieser Stelle nicht so interessant...

Definition (Substring, Superstring und Superstringsprache)

Sei Σ ein Alphabet und seien v, w Wörter über Σ^*.

(a) v ist ein *Substring* von w genau dann, wenn es Wörter $v_1, v_2 \in \Sigma^*$ gibt mit $w = v_1 v v_2$. Wir schreiben $v \trianglelefteq w$ für v ist ein Substring von w.

(b) Wir nennen w *Superstring* von v, wenn $v \trianglelefteq w$

(c) Die *Superstring-Sprache* für ein Wort $v \in \Sigma^*$ ist definiert als $\mathsf{L}_{v\trianglelefteq} \coloneqq \{u \in \Sigma^* \mid v \trianglelefteq u\}$

AUFGABE 2.7.

Stufe E

Sei $\Sigma = \{\mathsf{m}, \mathsf{i}, \mathsf{a}\}$ das Alphabet.

(a) Sei $w = \mathsf{mia}$ und $v = \mathsf{mammamia}$.

(i) Geben Sie einen NFA an, der die Sprache $\mathsf{L}_{w\trianglelefteq}$ akzeptiert.

(ii) Geben Sie einen DFA an, der die Sprache $\mathsf{L}_{w\trianglelefteq}$ akzeptiert und nicht mehr Zustände als Ihr NFA aus der vorherigen Teilaufgabe hat.

(iii) Geben Sie einen NFA an, der die Sprache $\mathsf{L}_{v\trianglelefteq}$ akzeptiert.

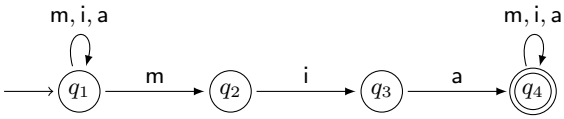
(iv) Geben Sie einen DFA an, der die die Sprache $\mathsf{L}_{v\trianglelefteq}$ akzeptiert und dabei nicht mehr Zustände als Ihr NFA aus der vorherigen Teilaufgabe hat.

(b) Beschreiben Sie die Unterschiede und Gemeinsamkeiten der beiden NFAs und DFAs aus der vorherigen Teilaufgabe. Erklären Sie in möglichst einfacher Sprache, wie Sie es erreicht haben, dass die NFAs und DFAs jeweils die gleiche Anzahl an Zuständen hatten.

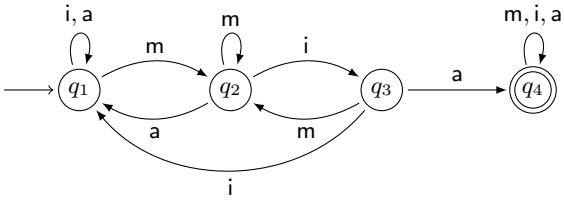
(c) Entwickeln Sie eine allgemeine Konstruktionsvorschrift für NFAs und DFAs, die die Sprache $\mathsf{L}_{w\trianglelefteq}$ für ein beliebiges Wort $w \in \Sigma^*$. Nutzen Sie dafür ihre Beobachtungen aus den vorherigen Aufgabenteilen.

Lösungsskizze

(a) (i) NFA, der $\mathsf{L}_{w\trianglelefteq}$ akzeptiert:



(ii) DFA, der $\mathsf{L}_{w\trianglelefteq}$ akzeptiert:



(iii) Der NFA zur Superstring-Sprache von v hat 9 Zustände und ist ähnlich aufgebaut wie der NFA in Aufgabenteil (i). Es gibt eine Sequenz zum Lesen von v und am Anfangs- und am Endzustand gibt es eine Schleife zum Lesen von m, i und a.

(iv) Ähnlich wie Aufgabenteil (ii) muss man sich überlegen, zu welchem Zustand man zurückgehen muss, wenn man einen Buchstaben gehört, der nicht zu mammamia gehört. Hat man zum Beispiel ein m gelesen, so kann entweder mammamia mit einem a fortgesetzt werden, es könnte ein weiteres m gelesen werden, was wiederum dann der Anfang des Worts v sein könnte oder es wird ein i gelesen, welches zum Startzustand zurückführt, da gerade nicht ein Teil von v gelesen worden sein kann.

(b) Der NFA rät, wann das Wort gelesen wird. Wir müssen daher anders als beim DFA nicht sicherstellen, dass wir bei Buchstaben, die nicht unmittelbar zum Wort gehörten können, ob sie Teil einer anderen Teilsequenz sind.

(c) Die Konstruktionsvorschrift für NFAs $\mathsf{N} = (\mathsf{Q}, \Sigma, \delta, \mathsf{q}_0, \mathsf{F})$ zu einem Wort w über dem Alphabet Σ lautet:

(i) $\mathsf{Q} = \{u \mid \exists v \in \Sigma^*.\ uv = w\}$

(ii) $\delta = \{(\varepsilon, x, \varepsilon), (w, x, w) \mid x \in \Sigma\} \cup \{(u, x, ux) \mid x \in \Sigma \wedge \exists v \in \Sigma^*.\ uxv = w\}$

(iii) $\mathsf{q}_0 = \varepsilon$

(iv) $\mathsf{F} = \{w\}$

Um einen passenden DFA zu erhalten, kann man die Potenzmengenkonstruktion anwenden.

Man kann aber auch den (minimalen) DFA $\mathsf{D} = (\mathsf{Q}, \Sigma, \delta, \mathsf{q}_0, \mathsf{F})$ direkt konstruieren: Die Zustände sind alle Präfixe von w und die Konstruktion versucht das nächste Zeichen zu lesen um aus dem bis jetzt gelesen Präfix ein neues Präfix von w zu konstruieren. Falls dies nicht möglich ist, springt der DFA zu dem Präfix, das das längste Suffix von aktuell gelesenem Wort ist, zurück. Das Verfahren wird in `https://dl.acm.org/citation.cfm?doid=360825.360855` beschrieben.