Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Übungsblatt 2

- **Tipp:** Auf dieser (und dort verlinkten) Website(n) können Sie interaktiv DFAs, NFAs, reguläre Ausdrücke,... erzeugen, simuliereren, umwandeln, ...

  Beachten Sie, dass auf der Website \$ statt ϵ verwendet wird.
- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.
- Für den Rest des Semesters gilt: $\mathbb{N} \coloneqq \mathbb{N}_0 \coloneqq \{0, 1, 2, ...\}$ und $\mathbb{N}_+ \coloneqq \mathbb{N}_0 \setminus \{0\}$.

Vorbereitung (→ vor der Übung selbständig zu bearbeiten)

Vorbereitungsaufgabe Ü2.1. (*Wichtige Begriffe*)

Überprüfen Sie, dass Sie die folgenden Begriffe oder Notationen korrekt definieren können.

- NFA und deren Akzeptanzbedingung
- ϵ-NFA
- Potenzmengenkonstruktion
- Rechtslineare Grammatik
- Regulärer Ausdruck

Vorbereitungsaufgabe Ü2.2. (*Automata Tutor: NFAs und Reguläre Ausdrücke*)

Lösen Sie die Aufgaben Ü2.2 (a–i) auf Automata Tutor. Beachten Sie, dass wir für einen regulären Ausdruck r das folgende Makro definieren: $r^+ \coloneqq rr^*$.

Vorbereitungsaufgabe Ü2.3. (*Konstruieren von NFAs mit Einschränkungen*)

Entscheiden Sie, ob die folgenden Aussagen korrekt sind, 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 ...

(a) der Startzustand hat keine eingehenden Kanten.

(b) kein Endzustand hat eine ausgehende Kante.

(c) für jeden Zustand q gilt: alle eingehenden Kanten von q sind mit demselben Zeichen beschriftet.

(d) für jeden Zustand q gilt: alle ausgehenden Kanten von q sind mit demselben Zeichen beschriftet.

Zu dieser Aufgabe gibt es Video-Lösungen: (a) (b) (c) (d)

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: $N' = (Q', \Sigma, \delta', q_0', F')$, wobei

- $Q' = Q \cup \{q_0'\}$,
- $\delta' = \delta \cup \{(q_0', x, q) : (q_0, x, q) \in \delta\}$
- $F' = F \cup \{q_0' : q_0 \in F\}$

(b) Nein: Betrachte einen NFA N für $L = \{\epsilon, a\}$.

(c) 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 $\delta(q_0, b) \cap F \neq \emptyset$.

Übung und Nachbereitung

Übungsaufgabe Ü2.4. (*Tick Tock Boom, Blow Up*)

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

(a) Konstruieren Sie einen NFA N mit genau 4 Zuständen und $L(N) = L$.

(b) Determinisieren Sie den NFA N aus (a) mittels der Potenzmengenkonstruktion um einen DFA D mit $L(D) = L(N)$ zu erhalten.

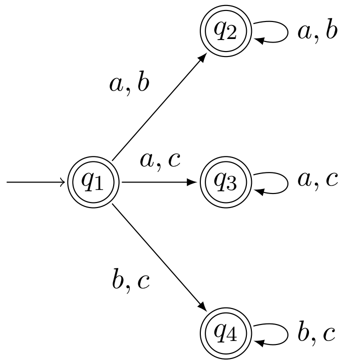
(c) Geben Sie eine rechtslineare Grammatik G (gemäß Satz 3.13) an, sodass $L(G) = L(D)$.

(d) Übersetzen Sie die Grammatik G (gemäß Satz 3.9) in einen NFA N', sodass $L(N') = L(G)$.

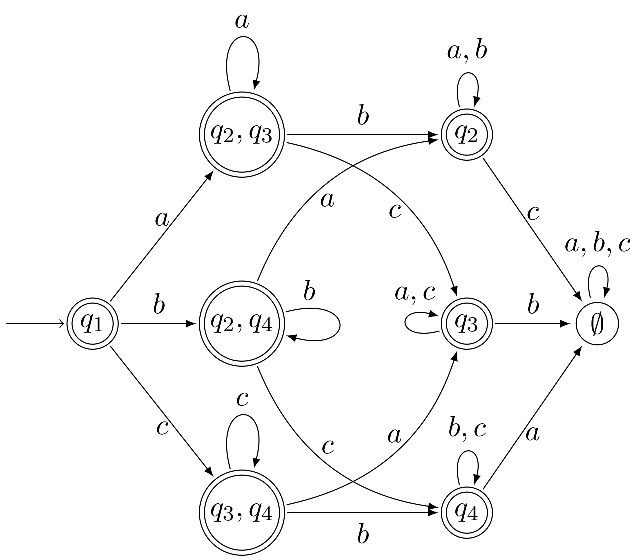
(e) Vergleichen Sie die NFAs N' und N bezüglich Zustands- und Transitionszahl. Diskutieren Sie dann, inwiefern Sie Ihre Beobachtung verallgemeinern können.

Lösungsskizze.

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



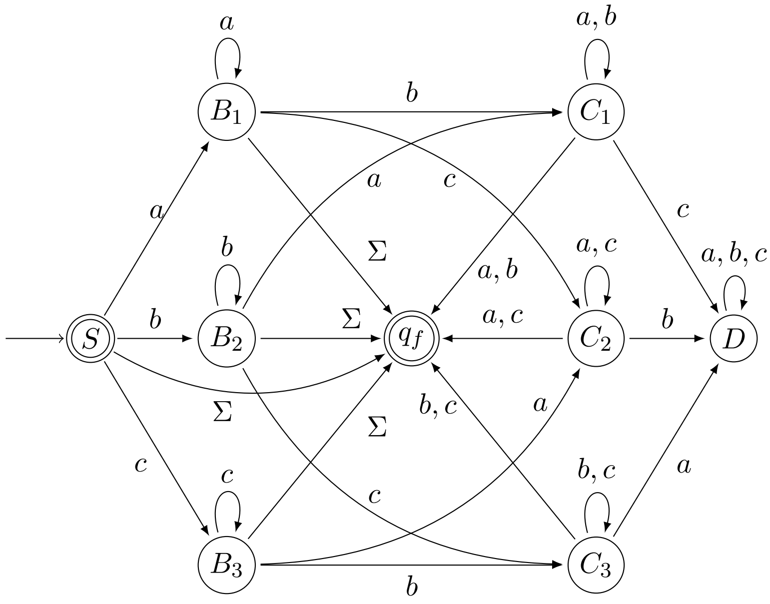
(b) Anwendung der Potenzmengenkonstruktion:



(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

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

(d) Wir erhalten folgenden Automaten:



(e) Diese Übersetzung (NFA → DFA → Grammatik → 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 Zustandzahl 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.

Übungsaufgabe Ü2.5. (*Teilwörter und reguläre Sprachen*)

Ein Teilwort eines Wortes w ist ein zusammenhängendes Wort in w. Wir definieren die Menge aller Teilwörter von w als $\downarrow[w] := \{w' \in \Sigma^* : w \text{ enthält } w'\}$. Also gilt beispielsweise $the, he, theo, \epsilon \in \downarrow[theo]$, aber $to, theoo \notin \downarrow[theo]$. Die Menge aller Wörter, von denen w ein Teilwort ist, ist dann entsprechend definiert als $\uparrow[w] := \{w' \in \Sigma^* : w' \text{ enthält } w\}$; z.B. $thetheotee \in \uparrow[theo]$. Wir erweitern dies auf Sprachen: Für eine beliebige Sprache $L \subseteq \Sigma^*$ definieren wir also $\downarrow[L] := \bigcup_{w \in L} \downarrow[w]$ und $\uparrow[L] := \bigcup_{w \in L} \uparrow[w]$.

Sei nun $\Sigma := \{t, h, e, o\}$ und L eine beliebige reguläre Sprache.

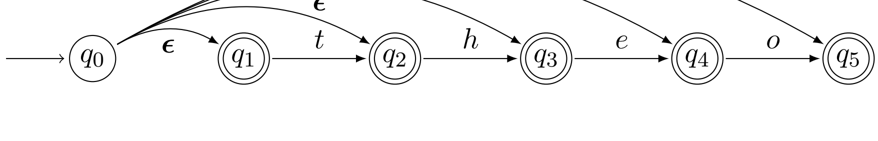
(a) Geben Sie für $\downarrow[theo]$ und $\uparrow[theo]$ einen ϵ-NFA an.

(b) Zeigen Sie, dass $\downarrow[L]$ regulär ist.

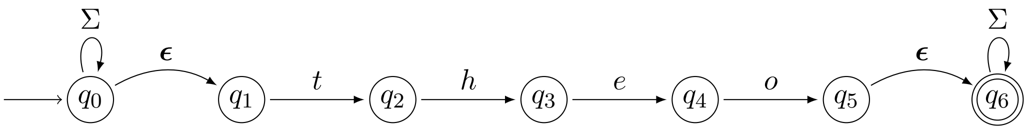
(c) Zeigen Sie, dass $\uparrow[L]$ regulär ist.

Lösungsskizze.

(a) ϵ-NFA für $\downarrow[theo]$:



ϵ-NFA für $\uparrow[theo]$:



(b) Wenn L regulär ist, gibt es einen NFA $N = (Q, \Sigma, \delta, q_0, F)$, der L akzeptiert. Ohne Beschränkung der Allgemeinheit können wir außerdem annehmen, dass in N jeder Zustand von q_0 aus erreichbar ist und aus jedem Zustand ein Endzustand erreicht werden kann. Beweis: Übung!

Wir zeigen nun, dass es einen ϵ-NFA N' gibt, der $\downarrow[L]$ akzeptiert, indem wir N modifizieren.

Idee: Da wir alle Teilwörter akzeptieren wollen, muss es in N' möglich sein, ein Präfix zu überspringen. Dafür führen wir einen neuen Startzustand ein, der jeden anderen Zustand mit einem ϵ-Übergang erreichen kann. Da auch ein beliebiges Suffix gelöscht werden darf, wird jeder Zustand in N akzeptierend (da nach Annahme von jedem Zustand aus ein Endzustand erreicht werden kann).

Wir definieren also den ϵ-NFA $N' := (Q \cup \{q_0'\}, \Sigma, \delta', q_0', Q)$ mit $q_0' \notin Q$. Dabei ist

$$\delta' := \delta \cup \{(q_0', \boldsymbol{\epsilon}, q) \mid q \in Q\}.$$

Im folgenden schreiben wir $q \rightarrow_N^w q'$ für $\hat{\hat{\delta}}(\{q\}, w) \ni q'$ und $q \rightarrow_{N'}^w q'$ für $\hat{\hat{\delta}}'(\{q\}, w) \ni q'$.

Wir müssen zeigen, dass $L(N') = \downarrow[L]$. Sei nun $w = a_1 \cdots a_n \in \Sigma^*$. Es gilt

$$\begin{aligned}
w \in L(N') &\iff \exists q_1, \dots, q_n.\ q_0' \rightarrow_{N'}^{\epsilon} q_1 \rightarrow_{N'}^{a_1} q_2 \cdots \rightarrow_{N'}^{a_n} q_n \in F' && \text{(Def. } \delta')\\
&\iff \exists q_1, \dots, q_n.\ q_1 \rightarrow_{N'}^{a_1} q_2 \cdots \rightarrow_{N'}^{a_n} q_n \in F' && \text{(Def. } \delta')\\
&\iff \exists q_1, \dots, q_n.\ q_1 \rightarrow_{N}^{a_1} q_2 \cdots \rightarrow_{N}^{a_n} q_n \in F' && \text{(Def. } \delta')\\
&\iff \exists q_1, q_n.\ q_1 \rightarrow_N^w q_n \in F'\\
&\iff \exists q_0, q_1, q_n, u.\ q_0 \rightarrow_N^u q_1 \rightarrow_N^w q_n \in F'\\
& && (q_1 \text{ erreichbar von } q_0 \text{ in } N)\\
&\iff \exists q_0, q_1, q_n, q_f, u, v.\ q_0 \rightarrow_N^u q_1 \rightarrow_N^w q_n \rightarrow_N^v q_f \in F\\
& && (F \text{ erreichbar von } q_n \text{ in } N)\\
&\iff \exists u, v.\ uwv \in L(N)\\
&\iff w \in \downarrow[L].
\end{aligned}$$

(c) Wenn L regulär ist, gibt es einen NFA $N = (Q, \Sigma, \delta, q_0, F)$, der L akzeptiert. Wir zeigen, dass es einen ϵ-NFA N' gibt, der $\uparrow[L]$ akzeptiert, indem wir N modifizieren.

Idee: Da wir alle Wörter akzeptieren, die ein Wort aus L als Teilwort enthalten, muss es in N' möglich sein ein beliebiges Präfix, dann ein beliebiges Wort aus L, und dann einen beliebiges Suffix zu lesen. Also fügen wir einen neuen Startzustand ein, bei dem beliebige Zeichen gelesen werden kann. Dieser erreicht den alten Startzustand mit einem ϵ-Übergang. Statt zu akzeptieren, haben alle in N

akzeptierenden Zustand einen ϵ-Übergang zum neuen Endzustand. In diesem wird jedes Wort akzeptiert.

Wir definieren also ϵ-NFA $N' := (Q \cup \{q_0', q_f\}, \Sigma, \delta', q_0', \{q_f\})$ mit $q_0', q_f \notin Q$. Dabei ist

$$\begin{aligned} \delta' := \delta &\cup \{(q_0', x, q_0') \mid x \in \Sigma\} \cup \{(q_0', \epsilon, q_0)\} \\ &\cup \{(q, \epsilon, q_f) \mid q \in F\} \quad \cup \{(q_f, x, q_f) \mid x \in \Sigma\} \end{aligned}$$

Wir müssen zeigen: $L(N') = \uparrow[L]$. Sei nun $w = a_1 \cdots a_n \in \Sigma^*$. Es gilt

$$\begin{aligned} w \in \uparrow[L] &\iff \exists u, w', v.\ w = uw'v \wedge w' \in L \\ &\iff \exists u, w', v.\ w = uw'v \wedge w' \in L(N) \\ &\iff \exists q_0, q_n, u, w', v.\ w = uw'v \wedge q_0 \to_N^{w'} q_n \in F \\ &\iff \exists q_0, q_n, u, w', v.\ w = uw'v \wedge q_0 \to_{N'}^{w'} q_n \in F && \text{(Def. } \delta'\text{)} \\ &\iff \exists q_0, q_n, u, w', v.\ w = uw'v \wedge q_0' \to_{N'}^{\epsilon} q_0 \to_{N'}^{w'} q_n \to_{N'}^{\epsilon} q_f \\ & && \text{(Def. } \delta'\text{)} \\ &\iff \exists q_0, q_n, u, w', v.\ w = uw'v \wedge q_0' \to_{N'}^{u} q_0' \to_{N'}^{\epsilon} q_0 \to_{N'}^{w'} q_n \to_{N'}^{\epsilon} q_f \to_{N'}^{v} q_f \\ & && \text{(Def. } \delta'\text{)} \\ &\iff \exists u, w', v.\ w = uw'v \wedge q_0' \to_{N'}^{uw'v} q_f \\ &\iff \exists u, w', v.\ w = uw'v \wedge q_0' \to_{N'}^{w} q_f \\ &\iff w \in L(N'). && \text{(Def. } \delta'\text{)} \end{aligned}$$

Übungsaufgabe Ü2.6.

In dieser Aufgabe beschäftigen wir uns mit rekursiven Funktionen und Beweisen auf regulären Ausdrücken.

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

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

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

Erinnerung: Da reguläre Ausdrücke induktiv definiert sind, sind sie mit einem Induktionsschema ausgestattet. Um zu beweisen, dass eine Eigenschaft $P(r)$ für alle regulären Ausdrücke r gilt, können wir dieses Induktionsschema verwenden. Bei Anwendung des Schemas ergeben sich dann folgende Beweisaufgaben:

- Zeige $P(\emptyset)$.
- Zeige $P(\epsilon)$.
- Zeige $P(\mathsf{a})$ für alle $\mathsf{a} \in \Sigma$.
- Unter der Annahme $P(\alpha)$ und $P(\beta)$, zeige $P(\alpha\beta)$.
- Unter der Annahme $P(\alpha)$ und $P(\beta)$, zeige $P(\alpha \mid \beta)$.

- Unter der Annahme $P(\alpha)$, zeige $P(\alpha^*)$.

Dieses Prinzip, welches für alle induktiv definierte Strukturen gilt, wird *strukturelle Induktion* genannt.

(b) **(optionale Teilaufgabe)** Gegeben sein nun folgende Funktion, die bestimmen soll, ob die von einem regulärem Ausdruck erzeugte Sprache unendlich viele Wörter enthält. In anderen Worten soll gelten, dass $\mathsf{infinite}(r) \iff |L(r)| = |\mathbb{N}|$.

- $\mathsf{infinite}(\boldsymbol{\emptyset}) = \mathsf{false}$
- $\mathsf{infinite}(\mathsf{a}) = \mathsf{false}$
- $\mathsf{infinite}(\boldsymbol{\epsilon}) = \mathsf{false}$
- $\mathsf{infinite}(\alpha\beta) = \mathsf{infinite}(\alpha) \vee \mathsf{infinite}(\beta)$
- $\mathsf{infinite}(\alpha \,|\, \beta) = \mathsf{infinite}(\alpha) \vee \mathsf{infinite}(\beta)$
- $\mathsf{infinite}(\alpha^*) = \mathsf{true}$

Beweisen Sie, dass die obige Definition korrekt ist oder widerlegen Sie die Korrektheit mit einem Gegenbeispiel.

Lösungsskizze.

(a) **Konstruktion**

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

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

Fall $r = \boldsymbol{\emptyset}$, $r = \mathsf{a}$, $r = \boldsymbol{\epsilon}$. Trivial; Beispiel:

$$\mathsf{empty}(\boldsymbol{\epsilon}) \overset{\text{Def.}}{\iff} \mathsf{false} \iff \{\epsilon\} = \emptyset \iff L(\boldsymbol{\epsilon}) = \emptyset$$

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

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

Es gilt

$$\begin{aligned}
L(\alpha\beta) = \emptyset &\iff L(\alpha)L(\beta) = \emptyset \\
&\iff L(\alpha) = \emptyset \vee L(\beta) = \emptyset \\
&\overset{\text{I.H.}}{\iff} \mathsf{empty}(\alpha) \vee \mathsf{empty}(\beta) = \mathsf{empty}(\alpha\beta).
\end{aligned}$$

Fall $r = \alpha \,|\, \beta$.
Es gelten dieselben Induktionshypothesen wie im vorherigen Fall.

Wir haben

$$\begin{aligned}
L(\alpha \,|\, \beta) = \emptyset &\iff L(\alpha) \cup L(\beta) = \emptyset \\
&\iff L(\alpha) = \emptyset \wedge L(\beta) = \emptyset \\
&\overset{\text{I.H.}}{\iff} \mathsf{empty}(\alpha) \wedge \mathsf{empty}(\beta) = \mathsf{empty}(\alpha \,|\, \beta).
\end{aligned}$$

(b) Die Definition ist nicht korrekt. Konkret haben wir $\mathsf{infinite}(\alpha^*) = \mathsf{true}$ für $\alpha \in \{\boldsymbol{\emptyset}, \boldsymbol{\epsilon}\}$ aber anderseits $|L(\alpha^*)| = |\{\epsilon\}| \neq |\mathbb{N}|$ und $\mathsf{infinite}(\boldsymbol{\emptyset}\mathsf{a}^*) = \mathsf{true}$ aber anderseits $|L(\boldsymbol{\emptyset}\mathsf{a}^*)| = |\emptyset\{\mathsf{a}\}^*| = |\emptyset| \neq |\mathbb{N}|$.

Um die Definition zu reparieren, müssen wir deshalb bestimmen, ob ein Wort $w \in L(\alpha)$ mit $\epsilon \neq w$ existiert. Umgekehrt betrachtet definieren wir eine Funktion nil mit der Eigenschaft $\mathsf{nil}(\alpha) \iff L(\alpha) = \{\epsilon\}$.

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

Wir passen die Definition folgendermaßen an.

- $\mathsf{infinite}(\alpha\beta) = (\neg\mathsf{empty}(\alpha) \wedge \mathsf{infinite}(\beta)) \vee (\neg\mathsf{empty}(\beta) \wedge \mathsf{infinite}(\alpha))$
- $\mathsf{infinite}(\alpha^*) = \neg\mathsf{nil}(\alpha^*)$

Wir beweisen zuerst mittels struktureller Induktion, dass nil korrekt definiert ist, d.h. wir zeigen $\mathsf{nil}(\alpha) \iff L(\alpha) = \{\epsilon\}$.

Fall $r = \boldsymbol{\emptyset}$, $r = \mathsf{a}$, $r = \boldsymbol{\epsilon}$. Trivial.

Fall $r = \alpha^*$.
Als Induktionshypothese erhalten wir, dass $\mathsf{nil}(\alpha) \iff L(\alpha) = \{\epsilon\}$. Es gilt

$$\begin{aligned} L(\alpha^*) = \{\epsilon\} &\iff L(\alpha)^* = \{\epsilon\} \\ &\iff L(\alpha) = \emptyset \vee L(\alpha) = \{\epsilon\} \\ &\iff \mathsf{empty}(\alpha) \vee L(\alpha) = \{\epsilon\} \\ &\overset{\text{I.H.}}{\iff} \mathsf{empty}(\alpha) \vee \mathsf{nil}(\alpha) \overset{\text{Def.}}{\iff} \mathsf{nil}(\alpha^*) \end{aligned}$$

Fall $r = \alpha\beta$.
Als Induktionshypothesen erhalten wir $L(\alpha) = \{\epsilon\} \iff \mathsf{nil}(\alpha)$ und $L(\beta) = \{\epsilon\} \iff \mathsf{nil}(\beta)$.

Es gilt

$$\begin{aligned} L(\alpha\beta) = \{\epsilon\} &\iff L(\alpha)L(\beta) = \{\epsilon\} \\ &\iff L(\alpha) = \{\epsilon\} \wedge L(\beta) = \{\epsilon\} \\ &\overset{\text{I.H.}}{\iff} \mathsf{nil}(\alpha) \wedge \mathsf{nil}(\beta) \overset{\text{Def.}}{\iff} \mathsf{nil}(\alpha\beta). \end{aligned}$$

Fall $r = \alpha \mid \beta$.
Es gelten dieselben Induktionshypothesen wie im vorherigen Fall.

Wir haben

$$\begin{aligned}
L(\alpha \,|\, \beta) = \{\epsilon\} &\iff L(\alpha) \cup L(\beta) = \{\epsilon\} \\
&\iff L(\alpha) = \{\epsilon\} \wedge L(\beta) = \{\epsilon\} \vee L(\alpha) = \{\epsilon\} \wedge L(\beta) = \emptyset \\
&\qquad \vee L(\beta) = \{\epsilon\} \wedge L(\alpha) = \emptyset \\
&\iff L(\alpha) = \{\epsilon\} \wedge L(\beta) = \{\epsilon\} \vee L(\alpha) = \{\epsilon\} \wedge \mathsf{empty}(\beta) \\
&\qquad \vee L(\beta) = \{\epsilon\} \wedge \mathsf{empty}(\alpha) \\
&\overset{\text{I.H.}}{\iff} \mathsf{nil}(\alpha) \wedge \mathsf{nil}(\beta) \vee \mathsf{nil}(\alpha) \wedge \mathsf{empty}(\beta) \vee \mathsf{nil}(\beta) \wedge \mathsf{empty}(\alpha) \\
&\overset{\text{Def.}}{\iff} \mathsf{nil}(\alpha \,|\, \beta).
\end{aligned}$$

Wir beweisen nun mittels struktureller Induktion, dass die Definition von $\mathsf{infinite}$ korrekt ist. Wir zeigen also $\mathsf{infinite}(\alpha) \iff |L(\alpha)| = |\mathbb{N}|$.

Fall $r = \boldsymbol{\emptyset}$, $r = \mathsf{a}$, $r = \boldsymbol{\epsilon}$. Trivial.

Fall $r = \alpha \,|\, \beta$.

Als Induktionshypothesen erhalten wir $|L(\alpha)| = |\mathbb{N}| \iff \mathsf{infinite}(\alpha)$ und $|L(\beta)| = |\mathbb{N}| \iff \mathsf{infinite}(\beta)$.

Es gilt

$$\begin{aligned}
|L(\alpha \,|\, \beta)| = |\mathbb{N}| &\iff |L(\alpha) \cup L(\beta)| = |\mathbb{N}| \\
&\iff |L(\alpha)| = |\mathbb{N}| \vee |L(\beta)| = |\mathbb{N}| \\
&\overset{\text{I.H.}}{\iff} \mathsf{infinite}(\alpha) \vee \mathsf{infinite}(\beta) \\
&\overset{\text{Def.}}{\iff} \mathsf{infinite}(\alpha \,|\, \beta).
\end{aligned}$$

Fall $r = \alpha\beta$.

Es gelten dieselben Induktionshypothesen wie im vorherigen Fall. Wir haben

$$\begin{aligned}
|L(\alpha\beta)| = |\mathbb{N}| &\iff |L(\alpha)L(\beta)| = |\mathbb{N}| \\
&\iff |L(\alpha)| = |\mathbb{N}| \wedge |L(\beta)| \neq 0 \vee |L(\beta)| = |\mathbb{N}| \wedge |L(\alpha)| \neq 0 \\
&\overset{\text{I.H.}}{\iff} \mathsf{infinite}(\alpha) \wedge |L(\beta)| \neq 0 \vee \mathsf{infinite}(\beta) \wedge |L(\alpha)| \neq 0 \\
&\iff \mathsf{infinite}(\alpha) \wedge \neg\mathsf{empty}(\beta) \vee \mathsf{infinite}(\beta) \wedge \neg\mathsf{empty}(\alpha) \\
&\overset{\text{Def.}}{\iff} \mathsf{infinite}(\alpha\beta).
\end{aligned}$$

Fall $r = \alpha^*$. Wir haben

$$\begin{aligned}
|L(\alpha^*)| = |\mathbb{N}| &\iff |L(\alpha)^*| = |\mathbb{N}| \\
&\iff L(\alpha)^* \neq \{\epsilon\} \\
&\iff \neg\mathsf{nil}(\alpha^*) \\
&\overset{\text{Def.}}{\iff} \mathsf{infinite}(\alpha^*).
\end{aligned}$$