Script **generated by TTT**

Title: Nipkow: Theo (02.05.2019)

Date: Thu May 02 14:15:03 CEST 2019

Duration: 86:53 min

Pages: 97

Fazit: Die Automatentypen

- DFA
- NFA
- ϵ-NFA

sind gleich mächtig.

3.5 Rechtslinearen Grammatiken

Erinnerung: rechtslineare Grammatiken haben nur Produktionen der Form $A \to a$, $A \to aB$ und $S \to \epsilon$.

3.5 Rechtslinearen Grammatiken

Erinnerung: rechtslineare Grammatiken haben nur Produktionen der Form $A \to a$, $A \to aB$ und $S \to \epsilon$.

57

Satz 3.13

Für jeden DFA M gibt es eine rechtslineare Grammatik G mit $L(M) = L(G)$.

Satz 3.13

Für jeden DFA M gibt es eine rechtslineare Grammatik G mit $L(M) = L(G)$.

Beweis:

Sei $M = (Q, \Sigma, \delta, q_0, F)$.

Satz 3.13

Für jeden DFA M gibt es eine rechtslineare Grammatik G mit $L(M) = L(G)$.

Beweis:

Sei $M = (Q, \Sigma, \delta, q_0, F)$. Die Grammatik $G = (V, T, P, S)$ mit

- $V = Q$, $T = \Sigma$, $S = q_0$,

Satz 3.13

Für jeden DFA M gibt es eine rechtslineare Grammatik G mit $L(M) = L(G)$.

Beweis:

Sei $M = (Q, \Sigma, \delta, q_0, F)$. Die Grammatik $G = (V, T, P, S)$ mit

- $V = Q$, $T = \Sigma$, $S = q_0$,
- $(q_1 \to a q_2) \in P$ gdw $\delta(q_1, a) = q_2$,

Satz 3.13

Für jeden DFA M gibt es eine rechtslineare Grammatik G mit $L(M) = L(G)$.

Beweis:

Sei $M = (Q, \Sigma, \delta, q_0, F)$. Die Grammatik $G = (V, T, P, S)$ mit

- $V = Q$, $T = \Sigma$, $S = q_0$,
- $(q_1 \to aq_2) \in P$ gdw $\delta(q_1, a) = q_2$,
- $\underline{(q_1 \to a)} \in P$ gdw $\underline{\delta(q_1, a) \in F}$, und

Satz 3.13

Für jeden DFA M gibt es eine rechtslineare Grammatik G mit $L(M) = L(G)$.

Beweis:

Sei $M = (Q, \Sigma, \delta, q_0, F)$. Die Grammatik $G = (V, T, P, S)$ mit

- $V = Q$, $T = \Sigma$, $S = q_0$,
- $\underline{(q_1 \to aq_2)} \in P$ gdw $\delta(q_1, a) = q_2$,
- $(q_1 \to a) \in P$ gdw $\delta(q_1, a) \in F$, und
- $\underline{(q_0 \to \epsilon)} \in P$ gdw $q_0 \in F$,

$p \to ap \mid bq \mid b$

$q \to ap \mid bp$

Satz 3.13

Für jeden DFA M gibt es eine rechtslineare Grammatik G mit $L(M) = L(G)$.

Beweis:

Sei $M = (Q, \Sigma, \delta, q_0, F)$. Die Grammatik $G = (V, T, P, S)$ mit

- $V = Q$, $T = \Sigma$, $S = q_0$,
- $(q_1 \to aq_2) \in P$ gdw $\delta(q_1, a) = q_2$,

Satz 3.13

Für jeden DFA M gibt es eine rechtslineare Grammatik G mit $L(M) = L(G)$.

Satz 3.13

Für jeden DFA M gibt es eine rechtslineare Grammatik G mit $L(M) = L(G)$.

Beweis:

Sei $M = (Q, \Sigma, \delta, q_0, F)$. Die Grammatik $G = (V, T, P, S)$ mit

- $V = Q$, $T = \Sigma$, $S = q_0$,
- $(q_1 \to a q_2) \in P$ gdw $\delta(q_1, a) = q_2$,
- $(q_1 \to a) \in P$ gdw $\delta(q_1, a) \in F$, und
- $(q_0 \to \epsilon) \in P$ gdw $q_0 \in F$,

ist vom Typ 3 und erfüllt $L(G) = L(M)$:
Es gilt (Induktion über n):

$$q_0 \to_G^* a_1 \cdots a_n \quad \text{gdw} \quad \hat{\delta}(q_0, a_1 \ldots a_n) \in F$$

□

Satz 3.14

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

Satz 3.14

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

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine rechtslineare Grammatik.

Satz 3.14

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

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine rechtslineare Grammatik.

Der ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ mit

- $Q = V \cup \{q_f\}$ (wobei q_f neu: $q_f \notin V$)

Satz 3.14

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

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine rechtslineare Grammatik.
Der ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ mit

- $Q = V \cup \{q_f\}$ (wobei q_f neu: $q_f \notin V$)
- $Y \in \delta(X, a)$ gdw $(X \to aY) \in P$

Satz 3.14

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

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine rechtslineare Grammatik.
Der ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ mit

- $Q = V \cup \{q_f\}$ (wobei q_f neu: $q_f \notin V$)
- $Y \in \delta(X, a)$ gdw $(X \to aY) \in P$
- $q_f \in \delta(X, u)$ gdw $(X \to u) \in P$ wobei $u \in \Sigma \cup \{\epsilon\}$

Satz 3.14

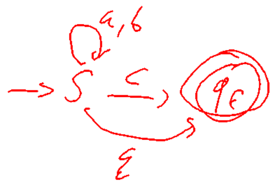
Für jede rechtslineare Grammatik G gibt es einen NFA M mit $L(G) = L(M)$.

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine rechtslineare Grammatik.
Der ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ mit

- $Q = V \cup \{q_f\}$ (wobei q_f neu: $q_f \notin V$)
- $Y \in \delta(X, a)$ gdw $(X \to aY) \in P$
- $q_f \in \delta(X, u)$ gdw $(X \to u) \in P$ wobei $u \in \Sigma \cup \{\epsilon\}$
- $q_0 = S$

$S \to aS \mid bS \mid c \mid \varepsilon$



Satz 3.14

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

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine rechtslineare Grammatik.
Der ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ mit

- $Q = V \cup \{q_f\}$ (wobei q_f neu: $q_f \notin V$)
- $Y \in \delta(X, a)$ gdw $(X \to aY) \in P$
- $q_f \in \delta(X, u)$ gdw $(X \to u) \in P$ wobei $u \in \Sigma \cup \{\epsilon\}$
- $q_0 = S$
- $F = \{q_f\}$

erfüllt $L(N) = L(G)$:

Satz 3.14

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

Beweis:

Sei $G = (V, \Sigma, P, S)$ eine rechtslineare Grammatik.

Der ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ mit

- $Q = V \cup \{q_f\}$ (wobei q_f neu: $q_f \notin V$)
- $Y \in \delta(X, a)$ gdw $(X \to aY) \in P$
- $q_f \in \delta(X, u)$ gdw $(X \to u) \in P$ wobei $u \in \Sigma \cup \{\boldsymbol{\epsilon}\}$
- $q_0 = S$
- $F = \{q_f\}$

erfüllt $L(N) = L(G)$:

Es gilt (Induktion über n):

$$q_f \in \hat{\delta}(S, a_1 \ldots a_n) \quad \text{gdw} \quad S \to_G^* a_1 \ldots a_n$$

$\square$

3.6 Reguläre Ausdrücke

Reguläre Ausdrücke sind eine weitere Notation für die Definition von formalen Sprachen.

3.6 Reguläre Ausdrücke

Reguläre Ausdrücke sind eine weitere Notation für die Definition von formalen Sprachen.

Definition 3.15

Reguläre Ausdrücke (*regular expressions*, REs) sind induktiv definiert:

3.6 Reguläre Ausdrücke

Reguläre Ausdrücke sind eine weitere Notation für die Definition von formalen Sprachen.

Definition 3.15

Reguläre Ausdrücke (*regular expressions*, REs) sind induktiv definiert:

- $\boldsymbol{\emptyset}$ ist ein regulärer Ausdruck.
- $\boldsymbol{\epsilon}$ ist ein regulärer Ausdruck.

3.6 Reguläre Ausdrücke

Reguläre Ausdrücke sind eine weitere Notation für die Definition von formalen Sprachen.

Definition 3.15

Reguläre Ausdrücke (*regular expressions*, REs) sind induktiv definiert:

- $\emptyset$ ist ein regulärer Ausdruck.
- ϵ ist ein regulärer Ausdruck.
- Für jedes $a \in \Sigma$ ist a ist ein regulärer Ausdruck.
- Wenn α und β reguläre Ausdrücke sind, dann auch
  - $\alpha\beta$
  - $\alpha \mid \beta$ (oft $\alpha + \beta$ geschrieben)
  - α^*

3.6 Reguläre Ausdrücke

Reguläre Ausdrücke sind eine weitere Notation für die Definition von formalen Sprachen.

Definition 3.15

Reguläre Ausdrücke (*regular expressions*, REs) sind induktiv definiert:

- $\emptyset$ ist ein regulärer Ausdruck.
- ϵ ist ein regulärer Ausdruck.
- Für jedes $a \in \Sigma$ ist a ist ein regulärer Ausdruck.
- Wenn α und β reguläre Ausdrücke sind, dann auch
  - $\alpha\beta$
  - $\alpha \mid \beta$ (oft $\alpha + \beta$ geschrieben)
  - α^*

Nichts sonst ist ein regulärer Ausdruck.

Notation:

- Reguläre Ausdrücke können bzw. müssen geklammert werden.

Notation:

- Reguläre Ausdrücke können bzw. müssen geklammert werden.
- Bindungsstärke: * stärker als Konkatenation stärker als $\mid$
- $ab^* = a(b^*) \neq (ab)^*$

Notation:

- Reguläre Ausdrücke können bzw. müssen geklammert werden.
- Bindungsstärke: * stärker als Konkatenation stärker als |
- $ab^* = a(b^*) \neq (ab)^*$
- $ab \mid c = (ab) \mid c \neq a(b \mid c)$

Definition 3.16

Die Sprache $L(\gamma)$ eines regulären Ausdrucks γ ist rekursiv definiert:

- $L(\boldsymbol{\emptyset}) = \emptyset$

Definition 3.16

Die Sprache $L(\gamma)$ eines regulären Ausdrucks γ ist rekursiv definiert:

- $L(\boldsymbol{\emptyset}) = \emptyset$
- $L(\boldsymbol{\epsilon}) = \{\epsilon\}$

Definition 3.16

Die Sprache $L(\gamma)$ eines regulären Ausdrucks γ ist rekursiv definiert:

- $L(\boldsymbol{\emptyset}) = \emptyset$
- $L(\boldsymbol{\epsilon}) = \{\epsilon\}$
- $L(a) = \{a\}$ $\quad a \in \Sigma$

Definition 3.16

Die Sprache $L(\gamma)$ eines regulären Ausdrucks γ ist rekursiv definiert:

- $L(\boldsymbol{\emptyset}) = \emptyset$
- $L(\boldsymbol{\epsilon}) = \{\epsilon\}$
- $L(a) = \{a\}$
- $L(\alpha\beta) = L(\alpha)L(\beta)$
- $L(\alpha \mid \beta) = L(\alpha) \cup L(\beta)$

Beispiel 3.17

Sei das zugrunde liegende Alphabet $\Sigma = \{0, 1\}$.

- Alle Wörter, die mit 00 enden:

$$(0|1)^*00$$

Beispiel 3.17

Sei das zugrunde liegende Alphabet $\Sigma = \{0, 1\}$.

- Alle Wörter, die mit 00 enden:

$$(0|1)^*00$$

- Alle Wörter gerader Länge, in denen 0 und 1 alternieren:

$$(01)^* \mid (10)^*$$

Beispiel 3.17

Sei das zugrunde liegende Alphabet $\Sigma = \{0, 1\}$.

- Alle Wörter, die mit 00 enden:

$$(0|1)^*00$$

- Alle Wörter gerader Länge, in denen 0 und 1 alternieren:

$$(01)^* \mid (10)^*$$

- Alle Wörter, die eine gerade Anzahl von 1'en enthalten:

$$(0^*10^*1)^*0^*$$

0^*

Beispiel 3.17

Sei das zugrunde liegende Alphabet $\Sigma = \{0, 1\}$.

- Alle Wörter, die mit 00 enden:

$$(0|1)^*00$$

- Alle Wörter gerader Länge, in denen 0 und 1 alternieren:

$$(01)^* \mid (10)^*$$

- Alle Wörter, die eine gerade Anzahl von 1'en enthalten:

$$(0^*10^*1)^*0^*$$

- Alle Wörter, die die Binärdarstellung einer durch 3 teilbaren Zahl darstellen, also

$$0, 11, 110, 1001, 1100, 1111, 10010, \ldots$$

Beispiel 3.17

Sei das zugrunde liegende Alphabet $\Sigma = \{0, 1\}$.

- Alle Wörter, die mit 00 enden:

$$(0|1)^*00$$

- Alle Wörter gerader Länge, in denen 0 und 1 alternieren:

$$(01)^* \mid (10)^*$$

- Alle Wörter, die eine gerade Anzahl von 1'en enthalten:

$$(0^*10^*1)^*0^*$$

- Alle Wörter, die die Binärdarstellung einer durch 3 teilbaren Zahl darstellen, also

$$0, 11, 110, 1001, 1100, 1111, 10010, \ldots$$

Hausaufgabe!

Erweiterte reguläre Ausdrücke in UNIX:

Erweiterte reguläre Ausdrücke in UNIX:

$$. \quad = \quad a_1|\ldots|a_n \text{ wobei } \Sigma = \{a_1, \ldots a_n\}$$

Erweiterte reguläre Ausdrücke in UNIX:

$$\begin{aligned} \texttt{.} &= a_1|\ldots|a_n \text{ wobei } \Sigma = \{a_1, \ldots a_n\} \\ \texttt{[}a_1 \ldots a_n\texttt{]} &= a_1|\ldots|a_n \end{aligned}$$

Erweiterte reguläre Ausdrücke in UNIX:

$$\begin{aligned} \texttt{.} &= a_1|\ldots|a_n \text{ wobei } \Sigma = \{a_1, \ldots a_n\} \\ \texttt{[}a_1 \ldots a_n\texttt{]} &= a_1|\ldots|a_n \\ \texttt{[\^{}}a_1 \ldots a_n\texttt{]} &= b_1|\ldots|b_m \text{ wobei } \{b_1, \ldots, b_m\} = \Sigma \setminus \{a_1, \ldots a_n\} \end{aligned}$$

Erweiterte reguläre Ausdrücke in UNIX:

$$\begin{aligned} \texttt{.} &= a_1|\ldots|a_n \text{ wobei } \Sigma = \{a_1, \ldots a_n\} \\ \texttt{[}a_1 \ldots a_n\texttt{]} &= a_1|\ldots|a_n \\ \texttt{[\^{}}a_1 \ldots a_n\texttt{]} &= b_1|\ldots|b_m \text{ wobei } \{b_1, \ldots, b_m\} = \Sigma \setminus \{a_1, \ldots a_n\} \\ \alpha\texttt{?} &= \boldsymbol{\epsilon}|\alpha \end{aligned}$$

Erweiterte reguläre Ausdrücke in UNIX:

$$\begin{aligned} \texttt{.} &= a_1|\ldots|a_n \text{ wobei } \Sigma = \{a_1, \ldots a_n\} \\ \texttt{[}a_1 \ldots a_n\texttt{]} &= a_1|\ldots|a_n \\ \texttt{[\^{}}a_1 \ldots a_n\texttt{]} &= b_1|\ldots|b_m \text{ wobei } \{b_1, \ldots, b_m\} = \Sigma \setminus \{a_1, \ldots a_n\} \\ \alpha\texttt{?} &= \boldsymbol{\epsilon}|\alpha \\ \alpha\texttt{+} &= \alpha\alpha^* \end{aligned}$$

Erweiterte reguläre Ausdrücke in UNIX:

$$
\begin{aligned}
\texttt{.} &= a_1|\dots|a_n \text{ wobei } \Sigma = \{a_1,\dots a_n\} \\
\texttt{[}a_1\dots a_n\texttt{]} &= a_1|\dots|a_n \\
\texttt{[\^{}}a_1\dots a_n\texttt{]} &= b_1|\dots|b_m \text{ wobei } \{b_1,\dots,b_m\} = \Sigma \setminus \{a_1,\dots a_n\} \\
\alpha\texttt{?} &= \boldsymbol{\epsilon}|\alpha \\
\alpha\texttt{+} &= \alpha\alpha^* \\
\alpha\{n\} &= \alpha\dots\alpha \ (n \text{ copies})
\end{aligned}
$$

Strukturelle Induktion

Da die regulären Ausdrücke induktiv definiert sind, gilt für sie das Prinzip der strukturellen Induktion:

Strukturelle Induktion

Da die regulären Ausdrücke induktiv definiert sind, gilt für sie das Prinzip der strukturellen Induktion:

Um zu beweisen, dass Eigenschaft P für alle regulären Ausdrücke γ gilt, also $P(\gamma)$, beweise

Strukturelle Induktion

Da die regulären Ausdrücke induktiv definiert sind, gilt für sie das Prinzip der strukturellen Induktion:

Um zu beweisen, dass Eigenschaft P für alle regulären Ausdrücke γ gilt, also $P(\gamma)$, beweise

- $P(\boldsymbol{\emptyset})$
- $P(\boldsymbol{\epsilon})$
- $P(a)$ für alle $a \in \Sigma$
- $P(\alpha) \wedge P(\beta) \Rightarrow P(\alpha\beta)$

Strukturelle Induktion
Da die regulären Ausdrücke induktiv definiert sind, gilt für sie das Prinzip der strukturellen Induktion:

Um zu beweisen, dass Eigenschaft P für alle regulären Ausdrücke γ gilt, also $P(\gamma)$, beweise

- $P(\emptyset)$
- $P(\epsilon)$
- $P(a)$ für alle $a \in \Sigma$
- $P(\alpha) \wedge P(\beta) \Rightarrow P(\alpha\beta)$
- $P(\alpha) \wedge P(\beta) \Rightarrow P(\alpha \mid \beta)$
- $P(\alpha) \Rightarrow P(\alpha^*)$

Komfortabler Spezialfall der Induktion über die Länge von γ.

Satz 3.19 (Kleene 1956)
Eine Sprache $L \subseteq \Sigma^*$ ist genau dann durch einen regulären Ausdruck darstellbar, wenn sie regulär ist.

Satz 3.19 (Kleene 1956)
Eine Sprache $L \subseteq \Sigma^*$ ist genau dann durch einen regulären Ausdruck darstellbar, wenn sie regulär ist.

Beweis:
"$\Longrightarrow$":
Sei $L = L(\gamma)$.

Satz 3.19 (Kleene 1956)
Eine Sprache $L \subseteq \Sigma^*$ ist genau dann durch einen regulären Ausdruck darstellbar, wenn sie regulär ist.

Beweis:
"$\Longrightarrow$":
Sei $L = L(\gamma)$.
Wir konstruieren einen ϵ-NFA N mit $L = L(N)$ mit Hilfe struktureller Induktion über γ.

Satz 3.19 (Kleene 1956)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann durch einen regulären Ausdruck darstellbar, wenn sie regulär ist.

Beweis:

“$\Longrightarrow$”:
Sei $L = L(\gamma)$.
Wir konstruieren einen $\boldsymbol{\epsilon}$-NFA N mit $L = L(N)$ mit Hilfe struktureller Induktion über γ.

Die Basisfälle $\gamma = \emptyset$, $\gamma = \boldsymbol{\epsilon}$, und $\gamma = a \in \Sigma$ sind offensichtlich.

Fall $\gamma = \alpha\beta$:

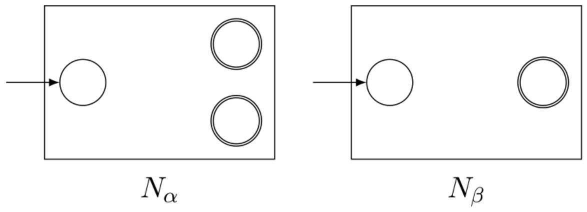
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.

Satz 3.19 (Kleene 1956)

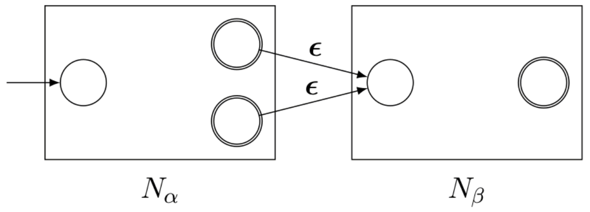
Eine Sprache $L \subseteq \Sigma^*$ ist genau dann durch einen regulären Ausdruck darstellbar, wenn sie regulär ist.

Fall $\gamma = \alpha\beta$:

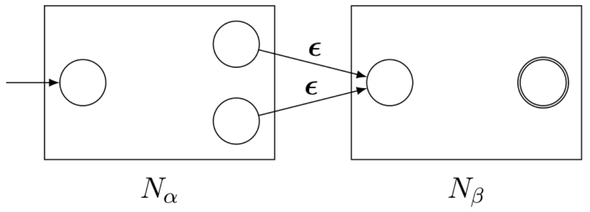
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



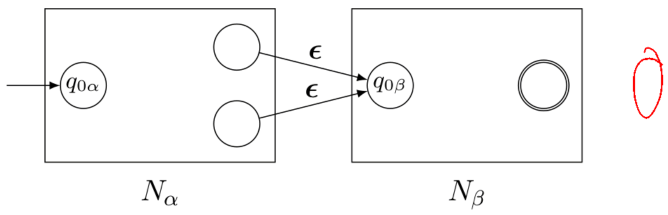
Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



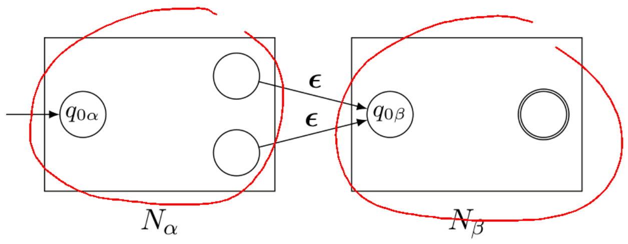
Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



Formal:

$$\begin{aligned} N_\alpha &= (Q_\alpha, \Sigma, \delta_\alpha, q_{0\alpha}, F_\alpha) \\ N_\beta &= (Q_\beta, \Sigma, \delta_\beta, q_{0\beta}, F_\beta) \end{aligned}$$

Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.

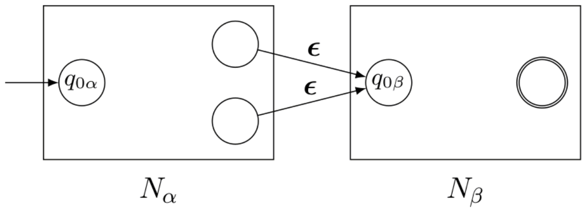


Formal:

$$\begin{aligned} N_\alpha &= (Q_\alpha, \Sigma, \delta_\alpha, q_{0\alpha}, F_\alpha) \\ N_\beta &= (Q_\beta, \Sigma, \delta_\beta, q_{0\beta}, F_\beta) \qquad \text{(mit } Q_\alpha \cap Q_\beta = \emptyset\text{)} \end{aligned}$$

Fall $\gamma = \alpha\beta$:

Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.

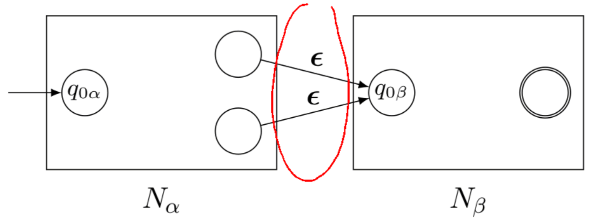


Formal:

$$
\begin{aligned}
N_\alpha &= (Q_\alpha, \Sigma, \delta_\alpha, q_{0\alpha}, F_\alpha) \\
N_\beta &= (Q_\beta, \Sigma, \delta_\beta, q_{0\beta}, F_\beta) \qquad (\text{mit } Q_\alpha \cap Q_\beta = \emptyset) \\
N_{\alpha\beta} &:= (Q_\alpha \cup Q_\beta, \Sigma, \delta, q_{0\alpha}, F_\beta)
\end{aligned}
$$

Fall $\gamma = \alpha\beta$:

Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



Formal:

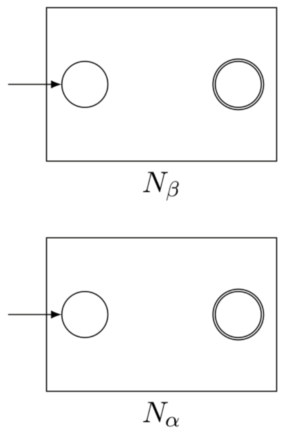
$$
\begin{aligned}
N_\alpha &= (Q_\alpha, \Sigma, \delta_\alpha, q_{0\alpha}, F_\alpha) \\
N_\beta &= (Q_\beta, \Sigma, \delta_\beta, q_{0\beta}, F_\beta) \qquad (\text{mit } Q_\alpha \cap Q_\beta = \emptyset) \\
N_{\alpha\beta} &:= (Q_\alpha \cup Q_\beta, \Sigma, \delta, q_{0\alpha}, F_\beta) \\
\delta &:= \underline{\delta_\alpha} \cup \delta_\beta \cup \{\underline{(f, \epsilon) \mapsto \{q_{0\beta}\}} \mid f \in F_\alpha\}
\end{aligned}
$$

Fall $\gamma = \alpha \mid \beta$:

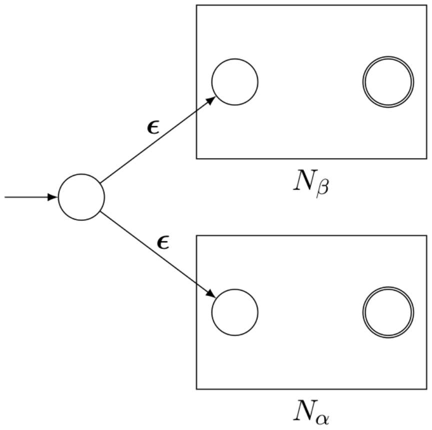
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.

Fall $\gamma = \alpha \mid \beta$:

Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



Fall $\gamma = \alpha \mid \beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.

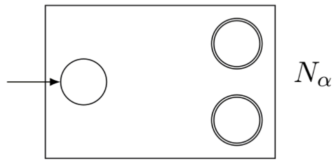


Fall $\gamma = \alpha^*$:
Nach Induktionsannahme können wir einen ϵ-NFA N_α konstruieren mit

$$L(N_\alpha) = L(\alpha) .$$

Fall $\gamma = \alpha^*$:
Nach Induktionsannahme können wir einen ϵ-NFA N_α konstruieren mit

$$L(N_\alpha) = L(\alpha) .$$



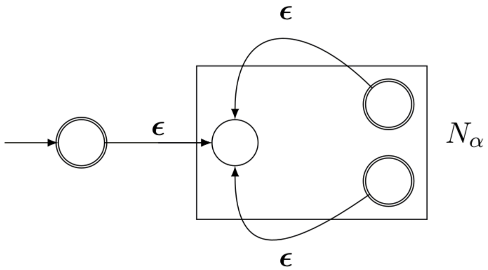
Fall $\gamma = \alpha^*$:
Nach Induktionsannahme können wir einen ϵ-NFA N_α konstruieren mit

$$L(N_\alpha) = L(\alpha) .$$

Fall $\gamma = \alpha^*$:

Nach Induktionsannahme können wir einen ϵ-NFA N_α konstruieren mit

$$L(N_\alpha) = L(\alpha) \ .$$



"$\Longleftarrow$":

Sei $M = (Q, \Sigma, \delta, q_1, F)$ ein DFA.

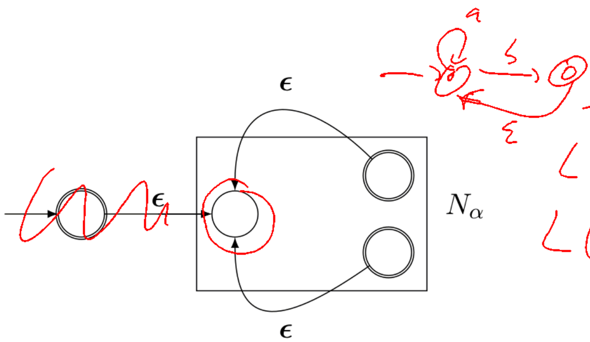
$(a^*b)^*$

$L(\ ) \neq (a^*b$

Fall $\gamma = \alpha^*$:

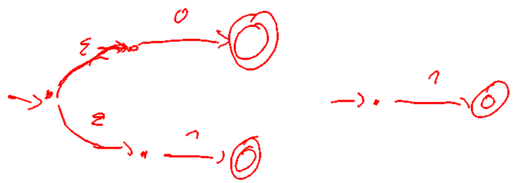
Nach Induktionsannahme können wir einen ϵ-NFA N_α konstruieren mit

$$L(N_\alpha) = L(\alpha) \ .$$



$L(\ ) \neq L((a^*b)^*)$

Bsp $(0|1)^*\,1\,(0|1)$

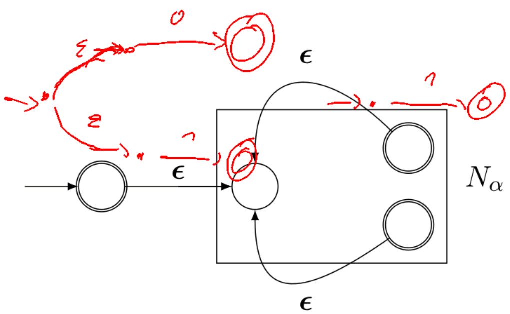


Bsp $(0|1)^* \, 1 \, (0|1)$

Fall $\gamma = \alpha^*$:

Nach Induktionsannahme können wir einen ϵ-NFA N_α konstruieren mit

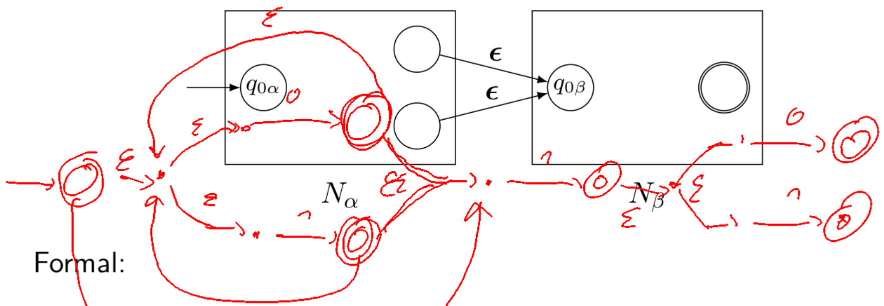
$$L(N_\alpha) = L(\alpha)\ .$$



Bsp $(0|1)^* \, 1 \, (0|1)$

Fall $\gamma = \alpha\beta$:

Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



Formal:

$$\begin{aligned}
N_\alpha &= (Q_\alpha, \Sigma, \delta_\alpha, q_{0\alpha}, F_\alpha) \\
N_\beta &= (Q_\beta, \Sigma, \delta_\beta, q_{0\beta}, F_\beta) \qquad (\text{mit } Q_\alpha \cap Q_\beta = \emptyset) \\
N_{\alpha\beta} &:= (Q_\alpha \cup Q_\beta, \Sigma, \delta, q_{0\alpha}, F_\beta) \\
\delta &:= \delta_\alpha \cup \delta_\beta \cup \{(f, \epsilon) \mapsto \{q_{0\beta}\} \mid f \in F_\alpha\}
\end{aligned}$$

"$\Longleftarrow$":

Sei $M = (Q, \Sigma, \delta, q_1, F)$ ein DFA.
Wir konstruieren einen RE γ mit $L(M) = L(\gamma)$.

"$\Longleftarrow$":

Sei $M = (Q, \Sigma, \delta, q_1, F)$ ein DFA.
Wir konstruieren einen RE γ mit $L(M) = L(\gamma)$.

Sei $Q = \{q_1, \ldots, q_n\}$. Wir setzen

$R^k_{ij} := \{w \in \Sigma^* \mid$ die Eingabe w führt von q_i in q_j, wobei alle Zwischenzustände (ohne ersten und letzten) einen Index $\leq k$ haben $\}$

"$\Longleftarrow$":
Sei $M = (Q, \Sigma, \delta, q_1, F)$ ein DFA.
Wir konstruieren einen RE γ mit $L(M) = L(\gamma)$.

Sei $Q = \{q_1, \dots, q_n\}$. Wir setzen

$R_{ij}^k := \{w \in \Sigma^* \mid$ die Eingabe w führt von q_i in q_j, wobei alle Zwischenzustände (ohne ersten und letzten) einen Index $\leq k$ haben $\}$

Behauptung: Für alle $i, j \in \{1, \dots, n\}$ und $k \in \{0, \dots, n\}$ können wir einen RE α_{ij}^k konstruieren mit $L(\alpha_{ij}^k) = R_{ij}^k$.

Bew.:

Induktion über k:
$k = 0$: Hier gilt

$$R_{ij}^0 = \begin{cases} \{a \in \Sigma \mid \delta(q_i, a) = q_j\}, & \text{falls } i \neq j \\ \{a \in \Sigma \mid \delta(q_i, a) = q_j\} \cup \{\epsilon\}, & \text{falls } i = j \end{cases}$$

Bew.:

Induktion über k:
$k = 0$: Hier gilt

$$R_{ij}^0 = \begin{cases} \{a \in \Sigma \mid \delta(q_i, a) = q_j\}, & \text{falls } i \neq j \\ \{a \in \Sigma \mid \delta(q_i, a) = q_j\} \cup \{\epsilon\}, & \text{falls } i = j \end{cases}$$

Bew.:

Induktion über k:
$k = 0$: Hier gilt

$$R_{ij}^0 = \begin{cases} \{a \in \Sigma \mid \delta(q_i, a) = q_j\}, & \text{falls } i \neq j \\ \{a \in \Sigma \mid \delta(q_i, a) = q_j\} \cup \{\epsilon\}, & \text{falls } i = j \end{cases}$$

Setze

$$\alpha_{ij}^0 := \begin{cases} a_1 | \dots | a_l & \text{falls } i \neq j \\ a_1 | \dots | a_l | \epsilon & \text{falls } i = j \end{cases}$$

wobei $\{a_1 \dots, a_l\} = \{a \in \Sigma \mid \delta(q_i, a) = q_j\}$.

Bew.:

Induktion über k:
$k \Rightarrow k+1$: Hier gilt

$$R_{ij}^{k+1} = \boxed{R_{ij}^{k}} \cup R_{i(k+1)}^{k}(R_{(k+1)(k+1)}^{k})^{*}R_{(k+1)j}^{k}$$

$$i \xrightarrow{\le k} k+1 \xrightarrow{\le k} k+1 \xrightarrow{\le k} j$$

Bew.:

Induktion über k:
$k \Rightarrow k+1$: Hier gilt

$$R_{ij}^{k+1} = R_{ij}^{k} \cup R_{i(k+1)}^{k}(R_{(k+1)(k+1)}^{k})^{*}R_{(k+1)j}^{k}$$

weil man jede Folge von Zahlen $\le k+1$ schreiben kann als

$$F_1, k+1, F_2, k+1, \dots, k+1, F_m$$

wobei jede F_l Folge von Zahlen $\le k$ ist.

Bew.:

Induktion über k:
$k \Rightarrow k+1$: Hier gilt

$$R_{ij}^{k+1} = R_{ij}^{k} \cup R_{i(k+1)}^{k}(R_{(k+1)(k+1)}^{k})^{*}R_{(k+1)j}^{k}$$

weil man jede Folge von Zahlen $\le k+1$ schreiben kann als

$$F_1, k+1, F_2, k+1, \dots, k+1, F_m$$

wobei jede F_l Folge von Zahlen $\le k$ ist.
Wir definieren rekursiv

$$\alpha_{ij}^{k+1} = \alpha_{ij}^{k} \mid \alpha_{i(k+1)}^{k}(\alpha_{(k+1)(k+1)}^{k})^{*}\alpha_{(k+1)j}^{k}$$

Somit gilt

$$L(M) = L(\alpha_{1i_1}^{n} \mid \cdots \mid \alpha_{1i_r}^{n})$$

wobei $F = \{i_1, \dots, i_r\}$. □

Unsere Konversionen auf einen Blick:

DFA NFA ϵ-NFA

RE

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & & \text{NFA} & & \epsilon\text{-NFA} \\ & & & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\to$ ϵ-NFA: RE der Länge $n \leadsto O(n)$ Zustände

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & & \text{NFA} & & \epsilon\text{-NFA} \\ & & & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\to$ ϵ-NFA: RE der Länge $n \leadsto O(n)$ Zustände

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & & \text{NFA} & \leftarrow & \epsilon\text{-NFA} \\ & & & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\to$ ϵ-NFA:	RE der Länge $n \leadsto O(n)$ Zustände
ϵ-NFA $\to$ NFA:	$Q \leadsto Q$

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & \leftarrow & \text{NFA} & \leftarrow & \epsilon\text{-NFA} \\ & & & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\to$ ϵ-NFA:	RE der Länge $n \leadsto O(n)$ Zustände
ϵ-NFA $\to$ NFA:	$Q \leadsto Q$
NFA $\to$ DFA:	n Zustände $\leadsto O(2^n)$ Zustände

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & \leftarrow & \text{NFA} & \leftarrow & \epsilon\text{-NFA} \\ & \searrow & \downarrow & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\rightarrow$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\rightarrow$ NFA:	$Q \rightsquigarrow Q$
NFA $\rightarrow$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\rightarrow$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & \leftarrow & \text{NFA} & \leftarrow & \epsilon\text{-NFA} \\ & \searrow & \downarrow & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\rightarrow$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\rightarrow$ NFA:	$Q \rightsquigarrow Q$
NFA $\rightarrow$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\rightarrow$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

Beweis FA$\rightarrow$RE: $m_k :=$ maximale Länge von α_{ij}^k

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & \leftarrow & \text{NFA} & \leftarrow & \epsilon\text{-NFA} \\ & \searrow & \downarrow & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\rightarrow$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\rightarrow$ NFA:	$Q \rightsquigarrow Q$
NFA $\rightarrow$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\rightarrow$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & \leftarrow & \text{NFA} & \leftarrow & \epsilon\text{-NFA} \\ & \searrow & \downarrow & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\rightarrow$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\rightarrow$ NFA:	$Q \rightsquigarrow Q$
NFA $\rightarrow$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\rightarrow$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

Bew.:

Induktion über k:
$k \Rightarrow k+1$: Hier gilt

$$R_{ij}^{k+1} = R_{ij}^{k} \cup R_{i(k+1)}^{k} (R_{(k+1)(k+1)}^{k})^{*} R_{(k+1)j}^{k}$$

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & \leftarrow & \text{NFA} & \leftarrow & \epsilon\text{-NFA} \\ & \searrow & \downarrow & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\to$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\to$ NFA:	$Q \rightsquigarrow Q$
NFA $\to$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\to$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

Beweis FA$\to$RE: $m_k :=$ maximale Länge von α_{ij}^{k}

- $m_0 = c_1|\Sigma|$

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & & \text{NFA} & & \epsilon\text{-NFA} \\ & & & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\to$ ϵ-NFA: RE der Länge $n \rightsquigarrow O(n)$ Zustände

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & \leftarrow & \text{NFA} & \leftarrow & \epsilon\text{-NFA} \\ & \searrow & \downarrow & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\to$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\to$ NFA:	$Q \rightsquigarrow Q$
NFA $\to$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\to$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

Beweis FA$\to$RE: $m_k :=$ maximale Länge von α_{ij}^{k}

- $m_0 = c_1|\Sigma|$
- $m_{k+1} = 4 * m_k + c_2$
- $m_k \in O(4^k)$

Unsere Konversionen auf einen Blick:

$$\begin{array}{ccccc} \text{DFA} & \leftarrow & \text{NFA} & \leftarrow & \epsilon\text{-NFA} \\ & \searrow & \downarrow & \nearrow & \\ & & \text{RE} & & \end{array}$$

RE $\rightarrow$ ϵ-NFA:	RE der Länge $n \leadsto O(n)$ Zustände
ϵ-NFA $\rightarrow$ NFA:	$Q \leadsto Q$
NFA $\rightarrow$ DFA:	n Zustände $\leadsto O(2^n)$ Zustände
FA $\rightarrow$ RE:	n Zustände $\leadsto$ RE der Länge $O(n4^n)$

Beweis FA→RE: $m_k :=$ maximale Länge von α_{ij}^k

- $m_0 = c_1|\Sigma|$

Unsere Konversionen auf einen Blick:

DFA NFA ϵ-NFA

RE