Script generated by TTT

Title: Seidl: Theoretische_Informatik (26.04.2012)

Date: Thu Apr 26 16:00:14 CEST 2012

Duration: 92:17 min

Pages: 116

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

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

Satz 2.17 (Kleene 1956)

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

Satz 2.17 (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.

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

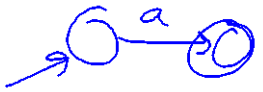
Satz 2.17 (Kleene 1956)

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

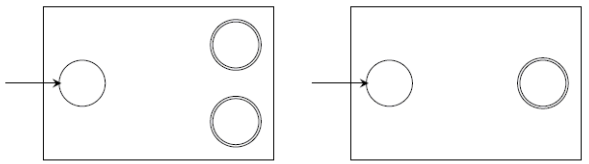
Beweis:

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

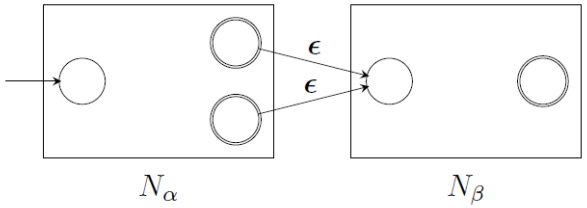
Die Basisfälle $\gamma = \emptyset$, $\gamma = \epsilon$, und $\gamma = a \in \Sigma$ sind offensichtlich.



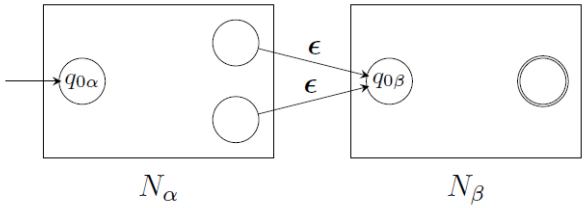
Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



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



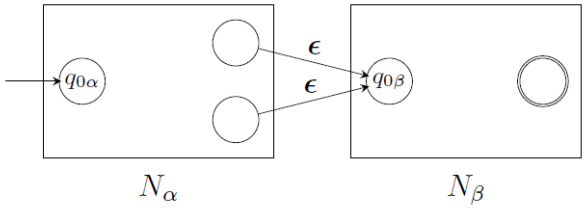
Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



Formal:

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

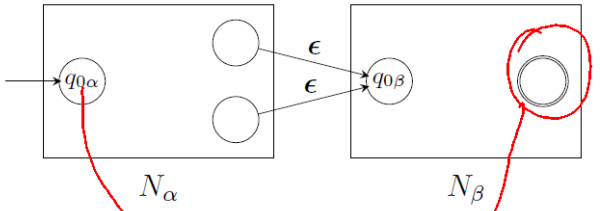
Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



Formal:

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

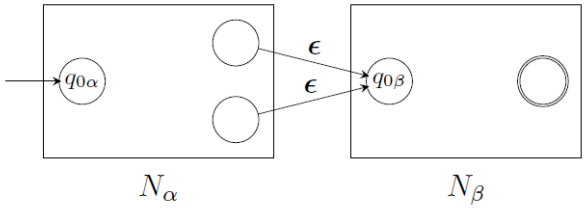
Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



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

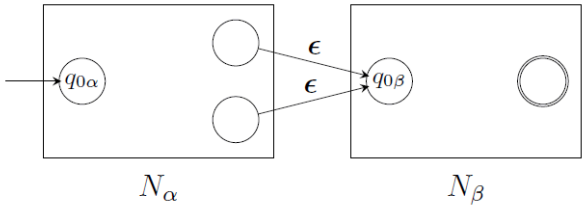


Formal:

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

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

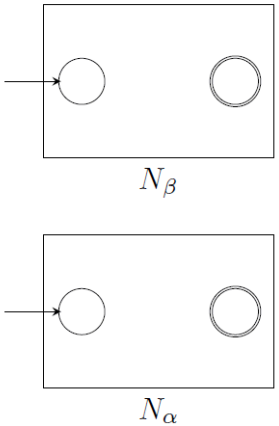
Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir ϵ-NFAs N_α und N_β konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.



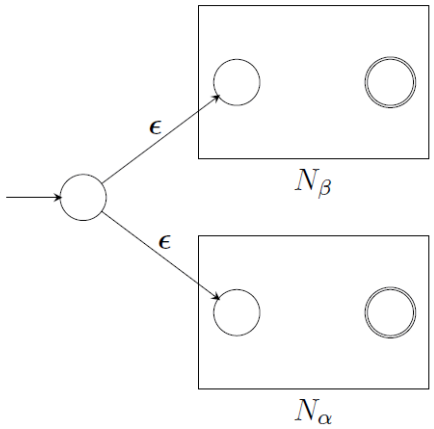
Formal:

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

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

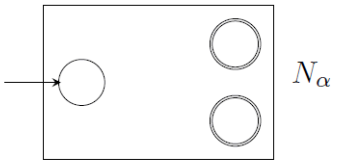


Fall $\gamma = \alpha^*$:
Nach Induktionsannahme können wir einen ϵ-NFA N_α konstruieren mit

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

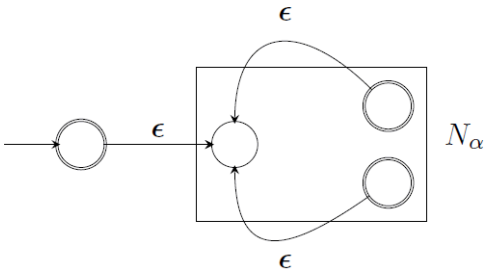
Fall $\gamma = \alpha^*$:
Nach Induktionsannahme können wir einen ϵ-NFA N_α konstruieren mit

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



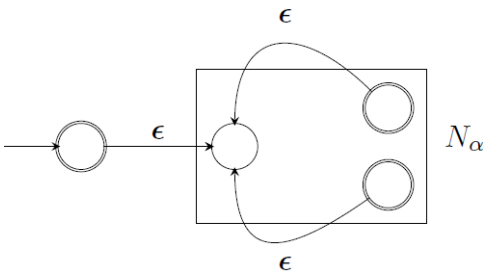
Fall $\gamma = \alpha^*$:
Nach Induktionsannahme können wir einen ϵ-NFA N_α konstruieren mit

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



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

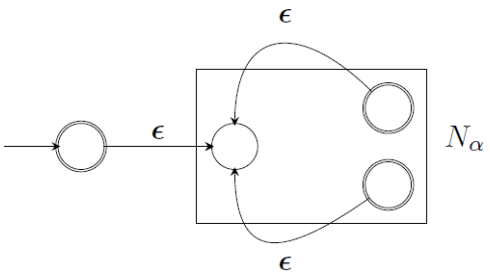
$$L(N_\alpha) = L(\alpha) \ .$$



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

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.
Wir konstruieren einen RE γ mit $L(M) = L(\gamma)$.

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

"$\Longleftarrow$":
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

$$R_{ij}^k := \{w \in \Sigma^* \mid \text{die Eingabe } w \text{ führt von } q_i \text{ in } q_j,$$

"$\Longleftarrow$":
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

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

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

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

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

Behauptung: Für alle $i, j \in \{1, \ldots, n\}$ und $k \in \{0, \ldots, n\}$ 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

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

Setze

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

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

Bew.:

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

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

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

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

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

Bew.:

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

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

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

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

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

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

Bew.:

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

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

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 $\leq k+1$ schreiben kann als

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

wobei jede F_l Folge von Zahlen $\leq k$ 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 $F = \{i_1, \ldots, i_r\}$. □

Fakt 2.18
Alle endlichen Sprachen sind regulär.

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 $\leq k+1$ schreiben kann als

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

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

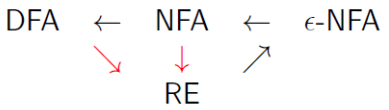
$$\alpha_{ij}^{k+1} = \alpha_{ij}^{k} \mid \alpha_{i(k+1)}^{k}(\alpha_{(k+1)(k+1)}^{k})^{*}\alpha_{(k+1)j}^{k}$$

Somit gilt

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

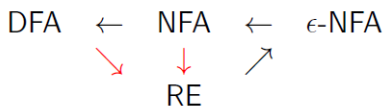
wobei $F = \{i_1, \ldots, i_r\}$. □

Unsere Konversionen auf einen Blick:



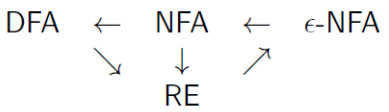
RE $\to$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\to$ NFA:	$Q \rightsquigarrow Q$
NFA $\to$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\to$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

Unsere Konversionen auf einen Blick:



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 $O(n4^n)$

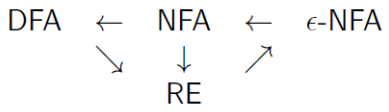
Unsere Konversionen auf einen Blick:



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

Unsere Konversionen auf einen Blick:

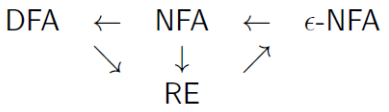


RE $\to$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\to$ NFA:	$Q \rightsquigarrow Q$
NFA $\to$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\to$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

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

- $m_0 \not\leq c_1|\Sigma|$

Unsere Konversionen auf einen Blick:

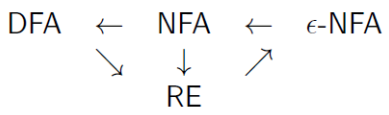


RE $\to$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\to$ NFA:	$Q \rightsquigarrow Q$
NFA $\to$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\to$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

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

- $m_0 = c_1|\Sigma|$
- $m_{k+1} = 4 * m_k + c_2$

Unsere Konversionen auf einen Blick:



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

- $m_0 = c_1|\Sigma|$

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 $\leq k+1$ schreiben kann als

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

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

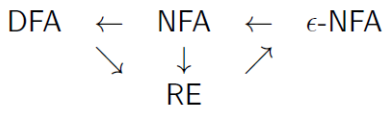
$$\alpha_{ij}^{k+1} = \alpha_{ij}^k \mid \alpha_{i(k+1)}^k (\alpha_{(k+1)(k+1)}^k)^* \alpha_{(k+1)j}^k$$

Somit gilt

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

wobei $F = \{i_1, \ldots, i_r\}$. □

Unsere Konversionen auf einen Blick:

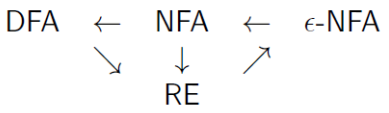


RE $\to$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\to$ NFA:	$Q \rightsquigarrow Q$
NFA $\to$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
FA $\to$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(n4^n)$

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:



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

- $m_0 = c_1|\Sigma|$
- $m_{k+1} = 4 * m_k + c_2$
- $m_k \in O(4^k)$
- Länge des Gesamtausdrucks: $O(n4^n)$

2.6 Abschlusseigenschaften regulärer Sprachen

Satz 2.19
Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch

$$R_1 R_2,$$

reguläre Sprachen.

2.6 Abschlusseigenschaften regulärer Sprachen

Satz 2.19
Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch

$$R_1 R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),$$

reguläre Sprachen.

2.6 Abschlusseigenschaften regulärer Sprachen

Satz 2.19
Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch

$$R_1 R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,$$

reguläre Sprachen.

2.6 Abschlusseigenschaften regulärer Sprachen

Satz 2.19
Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch

$$R_1 R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

reguläre Sprachen.

2.6 Abschlusseigenschaften regulärer Sprachen

Satz 2.19
Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch

$$R_1 R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

reguläre Sprachen.

Beweis:
$R_1 R_2$, $R_1 \cup R_2$, R^* klar.

2.6 Abschlusseigenschaften regulärer Sprachen

Satz 2.19
Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch

$$R_1 R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

reguläre Sprachen.

Beweis:
$R_1 R_2$, $R_1 \cup R_2$, R^* klar.

$\overline{R}$ Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$ DFA.

2.6 Abschlusseigenschaften regulärer Sprachen

Satz 2.19
Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch

$$R_1 R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

reguläre Sprachen.

Beweis:
$R_1 R_2$, $R_1 \cup R_2$, R^* klar.
$\overline{R}$ Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$ DFA.

Betrachte $A' = (Q, \Sigma, \delta, q_0, Q \setminus F)$.
Dann ist $L(A') = \overline{L(A)} = \overline{R}$

2.6 Abschlusseigenschaften regulärer Sprachen

Satz 2.19
Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch

$$R_1 R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

reguläre Sprachen.

Beweis:
$R_1 R_2$, $R_1 \cup R_2$, R^* klar.
$\overline{R}$ Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$ DFA.
Betrachte $A' = (Q, \Sigma, \delta, q_0, Q \setminus F)$.
Dann ist $L(A') = \overline{L(A)} = \overline{R}$

$R_1 \cap R_2 = \overline{\overline{R_1} \cup \overline{R_2}}$ (De Morgan)

2.6 Abschlusseigenschaften regulärer Sprachen

Satz 2.19
Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch

$$R_1 R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

reguläre Sprachen.

Beweis:
$R_1 R_2$, $R_1 \cup R_2$, R^* klar.

$\overline{R}$ Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$ DFA.
Betrachte $A' = (Q, \Sigma, \delta, q_0, Q \setminus F)$.
Dann ist $L(A') = \overline{L(A)} = \overline{R}$

$R_1 \cap R_2 = \overline{\overline{R_1} \cup \overline{R_2}}$ (De Morgan)

$R_1 \setminus R_2 = R_1 \cap \overline{R_2}$ □

Bemerkung
Komplementierung ($\overline{R}$) durch Vertauschen von Endzuständen und Nicht-Endzuständen funktioniert nur bei DFAs, nicht bei NFAs!

Übungsaufgabe: Finde Gegenbeispiel für NFAs

Bemerkung
Komplementierung ($\overline{R}$) durch Vertauschen von Endzuständen und Nicht-Endzuständen funktioniert nur bei DFAs, nicht bei NFAs!

Die Produkt-Konstruktion:
Durchschnitt direkt auf DFAs, ohne Umweg über de Morgan.

Die Produkt-Konstruktion:
Durchschnitt direkt auf DFAs, ohne Umweg über de Morgan.

Beide DFAs laufen synchron parallel, Wort wird akzeptiert wenn *beide* akzeptieren.

Die Produkt-Konstruktion:
Durchschnitt direkt auf DFAs, ohne Umweg über de Morgan.

Beide DFAs laufen synchron parallel, Wort wird akzeptiert wenn *beide* akzeptieren.

Parallelismus = Kreuzprodukt der Zustandsräume

Die Produkt-Konstruktion:
Durchschnitt direkt auf DFAs, ohne Umweg über de Morgan.

Beide DFAs laufen synchron parallel, Wort wird akzeptiert wenn *beide* akzeptieren.

Parallelismus = Kreuzprodukt der Zustandsräume

Satz 2.20
Sind $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_i)$ *und* $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ *DFAs, dann ist der Produkt-Automat*

$$M := (Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), F_1 \times F_2)$$

Die Produkt-Konstruktion:
Durchschnitt direkt auf DFAs, ohne Umweg über de Morgan.

Beide DFAs laufen synchron parallel, Wort wird akzeptiert wenn *beide* akzeptieren.

Parallelismus = Kreuzprodukt der Zustandsräume

Satz 2.20
Sind $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_i)$ *und* $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ *DFAs, dann ist der Produkt-Automat*

$$M := (Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), F_1 \times F_2)$$

$$\delta((q_1, q_2), a) := (\delta_1(q_1, a), \delta_2(q_2, a))$$

Die Produkt-Konstruktion:
Durchschnitt direkt auf DFAs, ohne Umweg über de Morgan.

Beide DFAs laufen synchron parallel, Wort wird akzeptiert wenn *beide* akzeptieren.

Parallelismus = Kreuzprodukt der Zustandsräume

Satz 2.20
Sind $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_i)$ *und* $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ *DFAs, dann ist der Produkt-Automat*

$$M := (Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), F_1 \times F_2)$$
$$\delta((q_1, q_2), a) := (\delta_1(q_1, a), \delta_2(q_2, a))$$

ein DFA der $L(M_1) \cap L(M_2)$ *akzeptiert.*

$F_1 \times Q_2 \cup Q_1 \times F_2$

Die Produkt-Konstruktion:
Durchschnitt direkt auf DFAs, ohne Umweg über de Morgan.

Beide DFAs laufen synchron parallel, Wort wird akzeptiert wenn *beide* akzeptieren.

Parallelismus = Kreuzprodukt der Zustandsräume

Satz 2.20
Sind $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_i)$ *und* $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ *DFAs, dann ist der Produkt-Automat*

$$M := (Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), F_1 \times F_2)$$
$$\delta((q_1, q_2), a) := (\delta_1(q_1, a), \delta_2(q_2, a))$$

ein DFA der $L(M_1) \cap L(M_2)$ *akzeptiert.*

Erinnerung: $|Q_1 \times Q_2| = |Q_1||Q_2|$

Beweis:
Mit Induktion über w:

$$\hat{\delta}((q_1, q_2), w) = (\hat{\delta}_1(q_1, w), \hat{\delta}_2(q_2, w)).$$

Definition 2.21
Die Umkehrung (Spiegelung) von $w = a_1 \dots a_n$ ist $w^R := a_n \dots a_1$.

Definition 2.21
Die Umkehrung (Spiegelung) von $w = a_1 \dots a_n$ ist $w^R := a_n \dots a_1$.
Die Umkehrung einer Sprache A ist $A^R := \{w^R \mid w \in A\}$

Satz 2.22
Ist A eine reguläre Sprache, dann auch A^R.

Definition 2.21
Die Umkehrung (Spiegelung) von $w = a_1 \dots a_n$ ist $w^R := a_n \dots a_1$.
Die Umkehrung einer Sprache A ist $A^R := \{w^R \mid w \in A\}$

Satz 2.22
Ist A eine reguläre Sprache, dann auch A^R.

Beweis:
Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA mit $L(M) = A$.

Definition 2.21
Die Umkehrung (Spiegelung) von $w = a_1 \dots a_n$ ist $w^R := a_n \dots a_1$.
Die Umkehrung einer Sprache A ist $A^R := \{w^R \mid w \in A\}$

Satz 2.22
Ist A eine reguläre Sprache, dann auch A^R.

Beweis:
Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA mit $L(M) = A$.
Wir konstruieren nun einen ϵ-NFA M' mit $L(M') = A^R$:

- Kehre alle Übergänge um: $p \xrightarrow{a} q \rightsquigarrow p \xleftarrow{a} q$

Definition 2.21
Die Umkehrung (Spiegelung) von $w = a_1 \dots a_n$ ist $w^R := a_n \dots a_1$.
Die Umkehrung einer Sprache A ist $A^R := \{w^R \mid w \in A\}$

Satz 2.22
Ist A eine reguläre Sprache, dann auch A^R.

Beweis:
Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA mit $L(M) = A$.
Wir konstruieren nun einen ϵ-NFA M' mit $L(M') = A^R$:

- Kehre alle Übergänge um: $p \xrightarrow{a} q \rightsquigarrow p \xleftarrow{a} q$
- Füge einen neuen Startzustand q_0' hinzu
  mit $q_0' \xrightarrow{\epsilon} f$ für alle $f \in F$.

Definition 2.21
Die Umkehrung (Spiegelung) von $w = a_1 \dots a_n$ ist $w^R := a_n \dots a_1$.
Die Umkehrung einer Sprache A ist $A^R := \{w^R \mid w \in A\}$

Satz 2.22
Ist A eine reguläre Sprache, dann auch A^R.

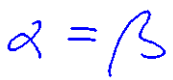
Beweis:
Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA mit $L(M) = A$.
Wir konstruieren nun einen ϵ-NFA M' mit $L(M') = A^R$:

- Kehre alle Übergänge um: $p \xrightarrow{a} q \rightsquigarrow p \xleftarrow{a} q$
- Füge einen neuen Startzustand q_0' hinzu
  mit $q_0' \xrightarrow{\epsilon} f$ für alle $f \in F$.
- Mache q_0 zum (einzigen) Endzustand.

Dann gilt $L(M') = A^R$.

Beweis (Forts.):
Formal: $M' = (Q \cup \{q_0'\}, \Sigma, \delta', q_0', \{q_0\})$ mit

- $q_0' \notin Q$,



2.7 Rechnen mit regulären Ausdrücken

Definition 2.23
Zwei reguläre Ausdrücke sind äquivalent gdw sie die gleiche Sprache darstellen:

$$\alpha \equiv \beta \;:\Leftrightarrow\; L(\alpha) = L(\beta)$$

Beispiel zum Unterschied von $=$ (syntaktische Identität) und $\equiv$ (Bedeutungsäquivalenz): $\epsilon \equiv \emptyset^*$ aber $\epsilon \neq \emptyset^*$.

2.7 Rechnen mit regulären Ausdrücken

Definition 2.23
Zwei reguläre Ausdrücke sind äquivalent gdw sie die gleiche Sprache darstellen:

$$\alpha \equiv \beta \;:\Leftrightarrow\; L(\alpha) = L(\beta)$$

Beispiel zum Unterschied von $=$ (syntaktische Identität) und $\equiv$ (Bedeutungsäquivalenz): $\epsilon \equiv \emptyset^*$ aber $\epsilon \neq \emptyset^*$.
Wird leicht (bewusst oder unbewusst) verwechselt.

Null und Eins:

Lemma 2.24

- $\emptyset \mid \alpha \equiv \alpha \mid \emptyset \equiv \alpha$

Null und Eins:

Lemma 2.24

- $\emptyset \mid \alpha \equiv \alpha \mid \emptyset \equiv \alpha$
- $\emptyset\alpha \equiv \alpha\emptyset \equiv \emptyset$

Null und Eins:

Lemma 2.24

- $\emptyset \mid \alpha \equiv \alpha \mid \emptyset \equiv \alpha$
- $\emptyset\alpha \equiv \alpha\emptyset \equiv \emptyset$
- $\epsilon\alpha \equiv \alpha\epsilon \equiv \alpha$

Null und Eins:

Lemma 2.24

- $\emptyset \mid \alpha \equiv \alpha \mid \emptyset \equiv \alpha$
- $\emptyset\alpha \equiv \alpha\emptyset \equiv \emptyset$
- $\epsilon\alpha \equiv \alpha\epsilon \equiv \alpha$
- $\emptyset^* \equiv \epsilon$
- $\epsilon^* \equiv \epsilon$

Lemma 2.25

Assoziativität:

- $(\alpha \mid \beta) \mid \gamma \equiv \alpha \mid (\beta \mid \gamma)$

Lemma 2.25

Assoziativität:

- $(\alpha \mid \beta) \mid \gamma \equiv \alpha \mid (\beta \mid \gamma)$

- $(\alpha\beta)\gamma \equiv \alpha(\beta\gamma)$

Lemma 2.25

Assoziativität:

- $(\alpha \mid \beta) \mid \gamma \equiv \alpha \mid (\beta \mid \gamma)$
- $(\alpha\beta)\gamma \equiv \alpha(\beta\gamma)$

Kommutativität:

- $\alpha \mid \beta \equiv \beta \mid \alpha$

Distributivität:

- $\alpha(\beta \mid \gamma) \equiv \alpha\beta \mid \alpha\gamma$
- $(\alpha \mid \beta)\gamma \equiv \alpha\gamma \mid \beta\gamma$

Lemma 2.25

Assoziativität:

- $(\alpha \mid \beta) \mid \gamma \equiv \alpha \mid (\beta \mid \gamma)$
- $(\alpha\beta)\gamma \equiv \alpha(\beta\gamma)$

Kommutativität:

- $\alpha \mid \beta \equiv \beta \mid \alpha$

Distributivität:

- $\alpha(\beta \mid \gamma) \equiv \alpha\beta \mid \alpha\gamma$
- $(\alpha \mid \beta)\gamma \equiv \alpha\gamma \mid \beta\gamma$

Idempotenz:

- $\alpha \mid \alpha \equiv \alpha$

Lemma 2.25

Assoziativität:

- $(\alpha \mid \beta) \mid \gamma \equiv \alpha \mid (\beta \mid \gamma)$
- $(\alpha\beta)\gamma \equiv \alpha(\beta\gamma)$

Kommutativität:

- $\alpha \mid \beta \equiv \beta \mid \alpha$

Distributivität:

- $\alpha(\beta \mid \gamma) \equiv \alpha\beta \mid \alpha\gamma$
- $(\alpha \mid \beta)\gamma \equiv \alpha\gamma \mid \beta\gamma$

Idempotenz:

- $\alpha \mid \alpha \equiv \alpha$

Stern:

Lemma 2.26

- $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$

Stern:

Lemma 2.26

- $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$
- $\alpha^*\alpha \equiv \alpha\alpha^*$

Stern:

Lemma 2.26

- $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$
- $\alpha^*\alpha \equiv \alpha\alpha^*$
- $(\alpha^*)^* \equiv \alpha^*$

Stern:

Lemma 2.26

- $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$
- $\alpha^*\alpha \equiv \alpha\alpha^*$
- $(\alpha^*)^* \equiv \alpha^*$

Beispiel 2.27

Herleitung einer Äquivalenz aus obigen Lemmas:

$\epsilon \mid \alpha^*$

Stern:

Lemma 2.26

- $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$
- $\alpha^*\alpha \equiv \alpha\alpha^*$
- $(\alpha^*)^* \equiv \alpha^*$

Beispiel 2.27

Herleitung einer Äquivalenz aus obigen Lemmas:

$\epsilon \mid \alpha^*$

Stern:

Lemma 2.26

- $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$
- $\alpha^*\alpha \equiv \alpha\alpha^*$
- $(\alpha^*)^* \equiv \alpha^*$

Beispiel 2.27

Herleitung einer Äquivalenz aus obigen Lemmas:

$\epsilon \mid \alpha^*$

$\equiv \epsilon \mid (\epsilon \mid \alpha\alpha^*)$ Stern Lemma

Stern:

Lemma 2.26

- $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$
- $\alpha^*\alpha \equiv \alpha\alpha^*$
- $(\alpha^*)^* \equiv \alpha^*$

Beispiel 2.27

Herleitung einer Äquivalenz aus obigen Lemmas:

$\epsilon \mid \alpha^*$
$\equiv \epsilon \mid (\epsilon \mid \alpha\alpha^*)$ Stern Lemma
$\equiv (\epsilon \mid \epsilon) \mid \alpha\alpha^*$ Assoziativität

Stern:

Lemma 2.26

- $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$
- $\alpha^*\alpha \equiv \alpha\alpha^*$
- $(\alpha^*)^* \equiv \alpha^*$

Beispiel 2.27

Herleitung einer Äquivalenz aus obigen Lemmas:

$\epsilon \mid \alpha^*$
$\equiv \epsilon \mid (\epsilon \mid \alpha\alpha^*)$ Stern Lemma
$\equiv (\epsilon \mid \epsilon) \mid \alpha\alpha^*$ Assoziativität
$\equiv \epsilon \mid \alpha\alpha^*$ Idempotenz

Stern:

Lemma 2.26

- $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$
- $\alpha^*\alpha \equiv \alpha\alpha^*$
- $(\alpha^*)^* \equiv \alpha^*$

Beispiel 2.27

Herleitung einer Äquivalenz aus obigen Lemmas:

$\epsilon \mid \alpha^*$
$\equiv \epsilon \mid (\epsilon \mid \alpha\alpha^*)$ Stern Lemma
$\equiv (\epsilon \mid \epsilon) \mid \alpha\alpha^*$ Assoziativität
$\equiv \epsilon \mid \alpha\alpha^*$ Idempotenz
$\equiv \alpha^*$ Stern Lemma

Lässt sich jede gültige Äquivalenz $\alpha \equiv \beta$ aus den obigen Lemmas für $\equiv$ herleiten?

Lässt sich jede gültige Äquivalenz $\alpha \equiv \beta$ aus den obigen Lemmas für $\equiv$ herleiten?

Satz 2.28 (Redko 1964)

Es gibt keine endliche Menge von gültigen Äquivalenzen aus denen sich alle gültigen Äquivalenzen herleiten lassen.

Lässt sich jede gültige Äquivalenz $\alpha \equiv \beta$ aus den obigen Lemmas für $\equiv$ herleiten?

Satz 2.28 (Redko 1964)

Es gibt keine endliche Menge von gültigen Äquivalenzen aus denen sich alle gültigen Äquivalenzen herleiten lassen.

Wenn man mehr als nur Äquivalenzen zulässt:

Arto Salomaa.
Two Complete Axiom Systems for the Algebra of Regular Events. Journal of the ACM, 1966.

Lässt sich jede gültige Äquivalenz $\alpha \equiv \beta$ aus den obigen Lemmas für $\equiv$ herleiten?

Satz 2.28 (Redko 1964)

Es gibt keine endliche Menge von gültigen Äquivalenzen aus denen sich alle gültigen Äquivalenzen herleiten lassen.

Wenn man mehr als nur Äquivalenzen zulässt:

Arto Salomaa.
Two Complete Axiom Systems for the Algebra of Regular Events. Journal of the ACM, 1966.

2.8 Pumping Lemma

Oder: *Wie zeigt man, dass eine Sprache nicht regulär ist?*

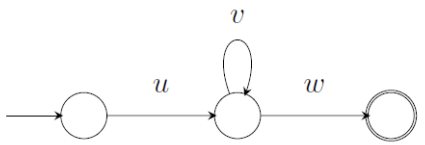
2.8 Pumping Lemma

Oder: *Wie zeigt man, dass eine Sprache nicht regulär ist?*

2.8 Pumping Lemma

Oder: *Wie zeigt man, dass eine Sprache nicht regulär ist?*

Beispiel 2.29



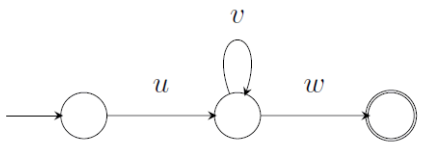
Satz 2.30 (Pumping Lemma für reguläre Sprachen)

Sei $R \subseteq \Sigma^*$ regulär. Dann gibt es ein $n > 0$, so dass sich jedes $z \in R$ mit $|z| \geq n$ so in $z = uvw$ zerlegen lässt, dass

2.8 Pumping Lemma

Oder: *Wie zeigt man, dass eine Sprache nicht regulär ist?*

Beispiel 2.29



Satz 2.30 (Pumping Lemma für reguläre Sprachen)

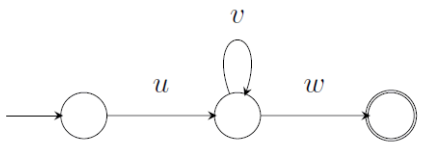
Sei $R \subseteq \Sigma^*$ regulär. Dann gibt es ein $n > 0$, so dass sich jedes $z \in R$ mit $|z| \geq n$ so in $z = uvw$ zerlegen lässt, dass

- $v \neq \epsilon$,
- $|uv| \leq n$, *und*

2.8 Pumping Lemma

Oder: *Wie zeigt man, dass eine Sprache nicht regulär ist?*

Beispiel 2.29



Satz 2.30 (Pumping Lemma für reguläre Sprachen)

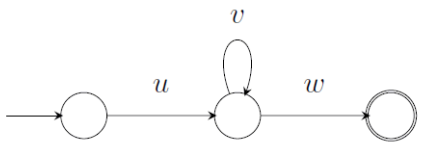
Sei $R \subseteq \Sigma^*$ regulär. Dann gibt es ein $n > 0$, so dass sich jedes $z \in R$ mit $|z| \geq n$ so in $z = uvw$ zerlegen lässt, dass

- $v \neq \epsilon$,
- $|uv| \leq n$, *und*
- $\forall i \geq 0.\ uv^i w \in R$.

2.8 Pumping Lemma

Oder: *Wie zeigt man, dass eine Sprache nicht regulär ist?*

Beispiel 2.29



Satz 2.30 (Pumping Lemma für reguläre Sprachen)

Sei $R \subseteq \Sigma^*$ regulär. Dann gibt es ein $n > 0$, so dass sich jedes $z \in R$ mit $|z| \geq n$ so in $z = uvw$ zerlegen lässt, dass

- $v \neq \epsilon$,
- $|uv| \leq n$, *und*
- $\forall i \geq 0.\ uv^i w \in R.$

Die logische Struktur des Satzes:

$\forall$*reg.* $R.\ \exists n > 0.$

Die logische Struktur des Satzes:

$\forall$*reg.* $R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$

Die logische Struktur des Satzes:

$\forall$*reg.* $R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$

Als Spiel:

- Gib mir eine reguläre Sprache L

Die logische Struktur des Satzes:

$$\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$$

Die logische Struktur des Satzes:

$$\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$$

Als Spiel:

- Gib mir eine reguläre Sprache L

Die logische Struktur des Satzes:

$$\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$$

Als Spiel:

- Gib mir eine reguläre Sprache L
- Dann gebe ich dir ein $n > 0$ (abhängig von L!!!)

Die logische Struktur des Satzes:

$$\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$$

Als Spiel:

- Gib mir eine reguläre Sprache L
- Dann gebe ich dir ein $n > 0$ (abhängig von L!!!)
- Gibst du mir dann ein z mit $|z| \geq n$

Die logische Struktur des Satzes:

$$\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$$

Als Spiel:

- Gib mir eine reguläre Sprache L
- Dann gebe ich dir ein $n > 0$ (abhängig von L!!!)
- Gibst du mir dann ein z mit $|z| \geq n$
- So kann ich z in uvw zerlegen so dass ...

Die logische Struktur des Satzes:

$$\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$$

Als Spiel:

- Gib mir eine reguläre Sprache L
- Dann gebe ich dir ein $n > 0$ (abhängig von L!!!)
- Gibst du mir dann ein z mit $|z| \geq n$
- So kann ich z in uvw zerlegen so dass ...

Sprechweise: n ist eine Pumping-Lemma-Zahl für R,

Beweis:
Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$.

Sei $n = |Q|$. Sei nun $z = a_1 \ldots a_m \in R$ mit $m \geq n$.

Beweis:
Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$.

Sei $n = |Q|$. Sei nun $z = a_1 \ldots a_m \in R$ mit $m \geq n$.

Die beim Lesen von z durchlaufene Zustandsfolge sei

$$q_0 = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \cdots \xrightarrow{a_m} p_m$$

Beweis:
Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$.

Sei $n = |Q|$. Sei nun $z = a_1 \dots a_m \in R$ mit $m \geq n$.

Die beim Lesen von z durchlaufene Zustandsfolge sei

$$q_0 = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \cdots \xrightarrow{a_m} p_m$$

Dann muss es $0 \leq i < j \leq n$ geben mit $p_i = p_j$.

Beweis:
Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$.

Sei $n = |Q|$. Sei nun $z = a_1 \dots a_m \in R$ mit $m \geq n$.

Die beim Lesen von z durchlaufene Zustandsfolge sei

$$q_0 = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \cdots \xrightarrow{a_m} p_m$$

Dann muss es $0 \leq i < j \leq n$ geben mit $p_i = p_j$.

Wir teilen z wie folg auf: $\underbrace{a_1 \dots a_i}_{u} \underbrace{a_{i+1} \dots a_j}_{v} \underbrace{a_{j+1} \dots a_{|z|}}_{w}$

Beweis:
Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$.

Sei $n = |Q|$. Sei nun $z = a_1 \dots a_m \in R$ mit $m \geq n$.

Die beim Lesen von z durchlaufene Zustandsfolge sei

$$q_0 = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \cdots \xrightarrow{a_m} p_m$$

Dann muss es $0 \leq i < j \leq n$ geben mit $p_i = p_j$.

Die logische Struktur des Satzes:

$\forall$*reg.* R.

Beweis:
Sei $R = L(A)$, $A = (Q, \Sigma, \delta, q_0, F)$.

Sei $n = |Q|$. Sei nun $z = a_1 \dots a_m \in R$ mit $m \geq n$.

Die beim Lesen von z durchlaufene Zustandsfolge sei

$$q_0 = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \cdots \xrightarrow{a_m} p_m$$

Dann muss es $0 \leq i < j \leq n$ geben mit $p_i = p_j$.

Wir teilen z wie folg auf: $\underbrace{a_1 \dots a_i}_{u} \underbrace{a_{i+1} \dots a_j}_{v} \underbrace{a_{j+1} \dots a_{|z|}}_{w}$

Damit gilt:

- $|uv| \leq n$,
- $v \neq \epsilon$, und
- $\forall l \geq 0.\ uv^l w \in R$.

□