**Script** **generated by TTT**

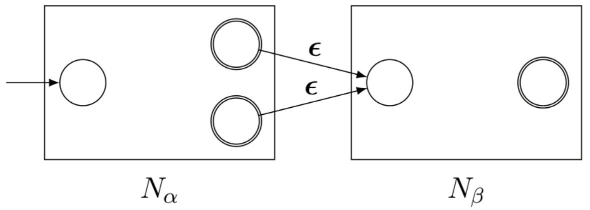
Title: Nipkow: Theo (06.05.2019)

Date: Mon May 06 14:15:12 CEST 2019

Duration: 86:32 min

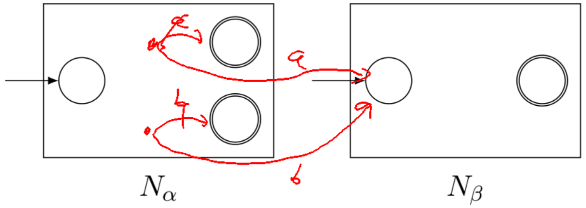
Pages: 96

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

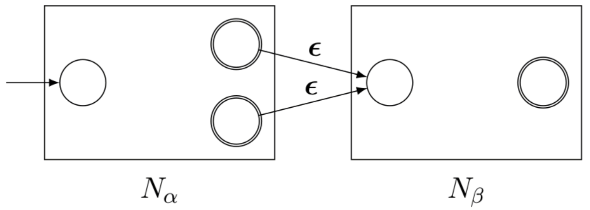





Fall $\gamma = \alpha\beta$:
Nach Induktionsannahme können wir $\epsilon$-NFAs $N_\alpha$ und $N_\beta$ konstruieren mit $L(N_\alpha) = L(\alpha)$ und $L(N_\beta) = L(\beta)$.




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



Beginn $\to \bullet \xrightarrow{\alpha} \circledcirc$

Transformationen:

“$\Longleftarrow$”:

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

$p \xrightarrow{\alpha\beta} q \;\Longrightarrow\; p \xrightarrow{\alpha} \bullet \xrightarrow{\beta} q$

$p \xrightarrow{\alpha|\beta} q \;\Longrightarrow\; p \overset{\alpha}{\underset{\beta}{\rightrightarrows}} q$

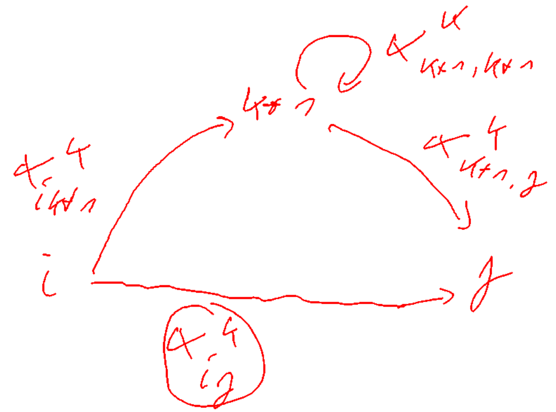
$p \xrightarrow{\alpha^*} q \;\Longrightarrow\; p \xrightarrow{\varepsilon} \bullet \xrightarrow{\varepsilon} q$, loop $\alpha$

$\to \bullet \xrightarrow{(0|1)^*1(0|1)} \circledcirc$

$\to \bullet \xrightarrow{(0|1)^*} \bullet \xrightarrow{1} \bullet \xrightarrow{0|1} \circledcirc$

$\to \bullet \xrightarrow{\varepsilon} \bullet \xrightarrow{\varepsilon} \bullet \xrightarrow{1} \bullet \overset{0}{\underset{1}{\rightrightarrows}} \circledcirc$, loop $0|1$






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

Beginn $\to \bullet \xrightarrow{\alpha} \circledcirc$

Transformationen:

“$\Longleftarrow$”:

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

$p \xrightarrow{\alpha\beta} q \;\Longrightarrow\; p \xrightarrow{\alpha} \bullet \xrightarrow{\beta} q$

$p \xrightarrow{\alpha|\beta} q \;\Longrightarrow\; p \overset{\alpha}{\underset{\beta}{\rightrightarrows}} q$

$p \xrightarrow{\alpha^*} q \;\Longrightarrow\; p \xrightarrow{\varepsilon} \bullet \xrightarrow{\varepsilon} q$, loop $\alpha$

$\to \bullet \xrightarrow{(0|1)^*1(0|1)} \circledcirc$

$\to \bullet \xrightarrow{(0|1)^*} \bullet \xrightarrow{1} \bullet \xrightarrow{0|1} \circledcirc$

$\to \bullet \xrightarrow{\varepsilon} \bullet \xrightarrow{\varepsilon} \bullet \xrightarrow{1} \bullet \overset{0}{\underset{1}{\rightrightarrows}} \circledcirc$, loop $0|1$



Bsp FA $\to$ RE


$\alpha_{11}^0 = \varepsilon \qquad \alpha_{12}^0 = a$

$\alpha_{21}^0 = b \qquad \alpha_{22}^0 = \varepsilon$

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



Bsp FA → RE

### 3.7 Abschlusseigenschaften regulärer Sprachen

**Satz 3.20**
*Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen. Dann sind auch*

$$R_1 R_2, \ R_1 \cup R_2,$$

*reguläre Sprachen.*

$\alpha_{11}^0 = \varepsilon$ $\alpha_{12}^0 = a$

$\alpha_{21}^0 = b$ $\alpha_{22}^0 = \varepsilon$

Verlauf

$\alpha_{11}^1 = \alpha_{11}^0 | \alpha_{11}^0 (\alpha_{11}^0)^* \alpha_{11}^0 = \varepsilon | \varepsilon \varepsilon^* \varepsilon$ — $\varepsilon$

$\alpha_{12}^1 = \alpha_{12}^0 | \alpha_{11}^0 (\alpha_{11}^0)^* \alpha_{12} = a | \varepsilon \varepsilon^* a$ — $a$

$\alpha_{21}^1 = b | b \varepsilon^* \varepsilon$ — $b$

$\alpha_{22} = \varepsilon | b \varepsilon^* a$ — $\varepsilon | ba$

$\alpha_{11}^2 = \alpha_{11}^1 | \alpha_{12}^1 (\alpha_{22}^1)^* \alpha_{21}$ — $\varepsilon | a (\varepsilon | ba)^* b$



### 3.7 Abschlusseigenschaften regulärer Sprachen

**Satz 3.20**
*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.



### 3.7 Abschlusseigenschaften regulärer Sprachen

**Satz 3.20**
*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 $R = L(A)$ für einen $A = (Q, \Sigma, \delta, q_0, F)$ DFA.

## 3.7 Abschlusseigenschaften regulärer Sprachen

Satz 3.20

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

$$R_1R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

*reguläre Sprachen.*

Beweis:

$R_1R_2$, $R_1 \cup R_2$, $R^*$ klar.

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



## 3.7 Abschlusseigenschaften regulärer Sprachen

Satz 3.20

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

$$R_1R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

*reguläre Sprachen.*

Beweis:

$R_1R_2$, $R_1 \cup R_2$, $R^*$ klar.

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



## 3.7 Abschlusseigenschaften regulärer Sprachen

Satz 3.20

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

$$R_1R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

*reguläre Sprachen.*

Beweis:

$R_1R_2$, $R_1 \cup R_2$, $R^*$ klar.

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

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



## 3.7 Abschlusseigenschaften regulärer Sprachen

Satz 3.20

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

$$R_1R_2,\ R_1 \cup R_2,\ R^*,\ \overline{R}\ (:= \Sigma^* \setminus R),\ R_1 \cap R_2,\ R_1 \setminus R_2$$

*reguläre Sprachen.*

Beweis:

$R_1R_2$, $R_1 \cup R_2$, $R^*$ klar.

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

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

$R_1 \setminus R_2 \quad =$

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



**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!

Übungsaufgabe: Finde Gegenbeispiel für NFAs

Bei NFAs:

Komplementierung erzwingt Determinierung.



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

Übungsaufgabe: Finde Gegenbeispiel für NFAs

Bei NFAs:

Komplementierung erzwingt Determinierung.

Komplementierung ist (potenziell) teuer.

**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.

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.



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 3.21
*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*

$$M := (Q_1 \times Q_2, \Sigma,$$

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 3.21

*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 3.21

*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*

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



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 3.21

*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.*



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 3.21

*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:
Durch Induktion über $w$ läßt sich zeigen:

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



Beweis:
Durch Induktion über $w$ läßt sich zeigen:

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

Damit gilt:

$$w \in L(M)$$



Beweis:
Durch Induktion über $w$ läßt sich zeigen:

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

Damit gilt:

$$\begin{aligned} & w \in L(M) \\ \Leftrightarrow \quad & (\hat{\delta}((s_1, s_2), w) \in F_1 \times F_2 \end{aligned}$$



Beweis:
Durch Induktion über $w$ läßt sich zeigen:

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

Damit gilt:

$$\begin{aligned} & w \in L(M) \\ \Leftrightarrow \quad & (\hat{\delta}((s_1, s_2), w) \in F_1 \times F_2 \\ \Leftrightarrow \quad & (\hat{\delta}_1(s_1, w), \hat{\delta}_2(s_2, w)) \in F_1 \times F_2 \end{aligned}$$

Beweis:
Durch Induktion über $w$ läßt sich zeigen:

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

Damit gilt:

$$\begin{aligned} & w \in L(M) \\ \Leftrightarrow \quad & (\hat{\delta}((s_1, s_2), w) \in F_1 \times F_2 \\ \Leftrightarrow \quad & (\hat{\delta}_1(s_1, w), \hat{\delta}_2(s_2, w)) \in F_1 \times F_2 \\ \Leftrightarrow \quad & \hat{\delta}_1(s_1, w) \in F_1 \wedge \hat{\delta}_2(s_2, w) \in F_2 \end{aligned}$$



Beweis:
Durch Induktion über $w$ läßt sich zeigen:

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

Damit gilt:

$$\begin{aligned} & w \in L(M) \\ \Leftrightarrow \quad & (\hat{\delta}((s_1, s_2), w) \in F_1 \times F_2 \\ \Leftrightarrow \quad & (\hat{\delta}_1(s_1, w), \hat{\delta}_2(s_2, w)) \in F_1 \times F_2 \\ \Leftrightarrow \quad & \hat{\delta}_1(s_1, w) \in F_1 \wedge \hat{\delta}_2(s_2, w) \in F_2 \\ \Leftrightarrow \quad & w \in L(M_1) \wedge w \in L(M_2) \\ \Leftrightarrow \quad & w \in L(M_1) \cap L(M_2) \end{aligned}$$

□



Beweis:
Durch Induktion über $w$ läßt sich zeigen:

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



Bsp: FA → RE

Beweis:
Durch Induktion über $w$ läßt sich zeigen:

$\alpha^0_{11} = \varepsilon \quad \alpha^0_{12} = a$


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

$\alpha^0_{21} = b \quad \alpha^0_{22} = \varepsilon$

Damit gilt:

Verlauf

$\alpha^1_{11} = \alpha^0_{11} \mid \alpha^0_{11} (\alpha^0_{11})^* \alpha^0_{11} = \varepsilon \mid \varepsilon\varepsilon^*\varepsilon$ — $\varepsilon$

$\alpha^1_{12} = \alpha^0_{12} \mid \alpha^0_{11} (\alpha^0_{11})^* \alpha^0_{12} = a \mid \varepsilon\varepsilon^* a$ — $a$

$\alpha^1_{21} = b \mid b\varepsilon^*\varepsilon$ — $b$

$\alpha^1_{22} = \varepsilon \mid b\varepsilon^* a$ — $\varepsilon \mid ba$

$$\begin{aligned} & w \in L(M) \\ \Leftrightarrow \quad & (\hat{\delta}((s_1, s_2), w) \in F_1 \times F_2 \\ \Leftrightarrow \quad & (\hat{\delta}_1(s_1, w), \hat{\delta}_2(s_2, w)) \in F_1 \times F_2 \\ \Leftrightarrow \quad & \hat{\delta}_1(s_1, w) \in F_1 \wedge \hat{\delta}_2(s_2, w) \in F_2 \\ \Leftrightarrow \quad & w \in L(M_1) \wedge w \in L(M_2) \\ \Leftrightarrow \quad & w \in L(M_1) \cap L(M_2) \end{aligned}$$

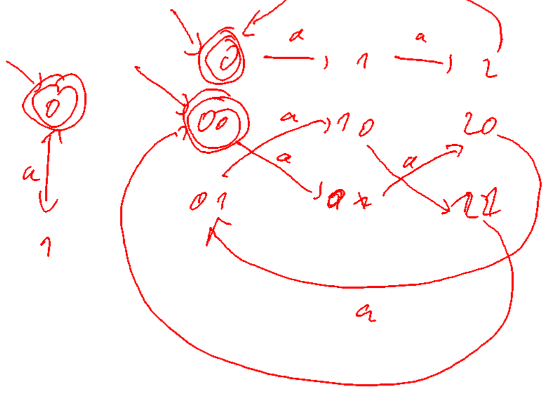
□

$\alpha^2_{11} = \alpha^1_{11} \mid \alpha^1_{12} (\alpha^1_{22})^* \alpha^1_{21}$ — $\varepsilon \mid a(\varepsilon \mid ba)^* b$

Funktioniert Durchschnitt durch Produkt auch für NFAs?

$L_1: (aa)^*$ $L_2: (aaa)^*$

## Definition 3.22
Die Umkehrung (Spiegelung) von $w = a_1 \dots a_n$ ist $w^R := a_n \dots a_1$.
Die Umkehrung einer Sprache $A$ ist $A^R := \{w^R \mid w \in A\}$





## Definition 3.22
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 3.23
*Ist $A$ eine reguläre Sprache, dann auch $A^R$.*



## Definition 3.22
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 3.23
*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 3.22
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 3.23
*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 $\epsilon$-NFA $M'$ mit $L(M') = A^R$:

## Definition 3.22

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 3.23

*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 $\epsilon$-NFA $M'$ mit $L(M') = A^R$:

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



## Definition 3.22

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 3.23

*Ist $A$ eine reguläre Sprache, dann auch $A^R$.*

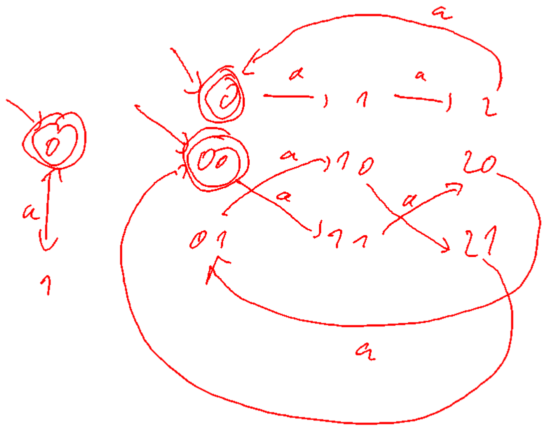
## Beweis:

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein DFA mit $L(M) = A$.
Wir konstruieren nun einen $\epsilon$-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.



$L_1: (aa)^*$ $L_2: (aaa)^*$




## Definition 3.22

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 3.23

*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 $\epsilon$-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.

$a$ $\varepsilon$ $\varepsilon$ $q_0'$

**Definition 3.22**
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 3.23**
*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 $\epsilon$-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.

$\epsilon$ $\epsilon$ $q_0'$



Umkehrung

### 3.8 Rechnen mit regulären Ausdrücken

**Definition 3.24**
Zwei reguläre Ausdrücke sind äquivalent gdw sie die gleiche Sprache darstellen:

Erkennt alle Wörter deren 4-te Bit 1 ist

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

Umkehrung leicht zu erzeugen/schnell

Beispiel zum Unterschied von $=$ (syntaktische Identität) und $\equiv$ (Bedeutungsäquivalenz):
$(\alpha \mid \beta) \equiv (\beta \mid \alpha)$ aber $(\alpha \mid \beta) \neq (\beta \mid \alpha)$.

größer aber prüfen



Null und Eins:

**Lemma 3.25**

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

Null und Eins:

Lemma 3.25

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



Null und Eins:

Lemma 3.25

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



Null und Eins:

Lemma 3.25

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



Null und Eins:

Lemma 3.25

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

## Lemma 3.26

*Assoziativität:*

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



## Lemma 3.26

*Assoziativität:*

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

$\alpha\beta \not\equiv \beta\alpha$



## Lemma 3.26

*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 3.26

*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 3.27

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



Stern:

Lemma 3.27

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



Stern:

Lemma 3.27

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

Beispiel 3.28

Herleitung einer Äquivalenz aus obigen Lemmas:

$\epsilon \mid \alpha^*$



Stern:

Lemma 3.27

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

Beispiel 3.28

Herleitung einer Äquivalenz aus obigen Lemmas:

$\epsilon \mid \alpha^*$

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

Stern: $\varepsilon \mid \alpha^+ \equiv \alpha^*$

### Lemma 3.27

- $\boxed{\epsilon \mid \alpha\alpha^* \equiv \alpha^*}$
- $\alpha^*\alpha \equiv \alpha\alpha^*$
- $(\alpha^*)^* \equiv \alpha^*$

### Beispiel 3.28

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 3.27

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

### Beispiel 3.28

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

$\Longrightarrow \varepsilon \mid \alpha^* \equiv \alpha^*$



Stern:

### Lemma 3.27

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

### Beispiel 3.28

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



$0 \mid \alpha \equiv \alpha$

### Lemma 3.26

$L(0 \mid \alpha) = L(0) \cup L(\alpha) = \emptyset \cup L(\alpha) = L(\alpha)$

*Assoziativität:*

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

**Lemma 3.26**

*Assoziativität:*

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



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

**Satz 3.29 (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 3.29 (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.



## 3.9 Pumping Lemma

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

### 3.9 Pumping Lemma

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

Satz 3.30 (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*



### 3.9 Pumping Lemma

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



### 3.9 Pumping Lemma

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

Satz 3.30 (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*



### 3.9 Pumping Lemma

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

Satz 3.30 (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*

- $v \neq \epsilon$,

### 3.9 Pumping Lemma

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

Satz 3.30 (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*
- $\forall i \geq 0. uv^iw \in R$.



- Die logische Struktur des Satzes:

  $\forall$*reg.* $R$.



- Die logische Struktur des Satzes:

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



- Die logische Struktur des Satzes:

  $\forall$*reg.* $R$. $\exists n > 0$. $\forall z \in R$. $|z| \geq n \Rightarrow$

$uv^iw$
$\in R$

- Die logische Struktur des Satzes:

  $\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$
- Das Pumping Lemma als Spiel:



- Die logische Struktur des Satzes:

  $\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$
- Das Pumping Lemma als Spiel:
  - Spieler I wählt eine reguläre Sprache $R$



- Die logische Struktur des Satzes:

  $\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$
- Das Pumping Lemma als Spiel:
  - Spieler I wählt eine reguläre Sprache $R$
  - Spieler II wählt ein $n > 0$



- Die logische Struktur des Satzes:

  $\forall \textit{reg}.\ R.\ \exists n > 0.\ \forall z \in R.\ |z| \geq n \Rightarrow\ \exists u\ v\ w.\ z = uvw \wedge \ldots$
- Das Pumping Lemma als Spiel:
  - Spieler I wählt eine reguläre Sprache $R$
  - Spieler II wählt ein $n > 0$
  - Spieler I wählt ein $z \in R$ mit $|z| \geq n$

- Die logische Struktur des Satzes:

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

- Das Pumping Lemma als Spiel:
  - Spieler I wählt eine reguläre Sprache $R$
  - Spieler II wählt ein $n > 0$
  - Spieler I wählt ein $z \in R$ mit $|z| \geq n$
  - Spieler II wählt eine Zerlegung $uvw$ von $z$



- Die logische Struktur des Satzes:

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

- Das Pumping Lemma als Spiel:
  - Spieler I wählt eine reguläre Sprache $R$
  - Spieler II wählt ein $n > 0$
  - Spieler I wählt ein $z \in R$ mit $|z| \geq n$
  - Spieler II wählt eine Zerlegung $uvw$ von $z$

Spieler II gewinnt, wenn die Zerlegung die gewünschten Eigenschaften erfüllt, sonst gewinnt Spieler I.



- Die logische Struktur des Satzes:

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

- Das Pumping Lemma als Spiel:
  - Spieler I wählt eine reguläre Sprache $R$
  - Spieler II wählt ein $n > 0$
  - Spieler I wählt ein $z \in R$ mit $|z| \geq n$
  - Spieler II wählt eine Zerlegung $uvw$ von $z$

Spieler II gewinnt, wenn die Zerlegung die gewünschten Eigenschaften erfüllt, sonst gewinnt Spieler I.

Das Pumping Lemma sagt, dass Spieler II eine *Gewinnstrategie* hat: Wenn er richtig spielt, dann gewinnt er jede Partie.



- Die logische Struktur des Satzes:

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

- Das Pumping Lemma als Spiel:
  - Spieler I wählt eine reguläre Sprache $R$
  - Spieler II wählt ein $n > 0$
  - Spieler I wählt ein $z \in R$ mit $|z| \geq n$
  - Spieler II wählt eine Zerlegung $uvw$ von $z$

Spieler II gewinnt, wenn die Zerlegung die gewünschten Eigenschaften erfüllt, sonst gewinnt Spieler I.

Das Pumping Lemma sagt, dass Spieler II eine *Gewinnstrategie* hat: Wenn er richtig spielt, dann gewinnt er jede Partie.

- Sprechweise: $n$ ist eine Pumping-Lemma-Zahl für $R$, falls alle $z \in R$ mit $|z| \geq n$ sich so wie im Pumping-Lemma zerlegen und aufpumpen lassen.

- Die logische Struktur des Satzes:

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

- Das Pumping Lemma als Spiel:
  - Spieler I wählt eine reguläre Sprache $R$
  - Spieler II wählt ein $n > 0$
  - Spieler I wählt ein $z \in R$ mit $|z| \geq n$
  - Spieler II wählt eine Zerlegung $uvw$ von $z$

  Spieler II gewinnt, wenn die Zerlegung die gewünschten Eigenschaften erfüllt, sonst gewinnt Spieler I.

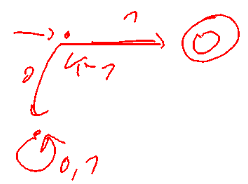
  Das Pumping Lemma sagt, dass Spieler II eine *Gewinnstrategie* hat: Wenn er richtig spielt, dann gewinnt er jede Partie.



Umkehrung

- Die logische Struktur des Satzes:

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



Erkennt alle Wörter deren k-te Bit 1 ist

Umkehrung kann zu exponentiell größeren DFAs führen



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

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

$q_0'$



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

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

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: $\underbrace{a_1 \dots a_i}_{u} \underbrace{a_{i+1} \dots a_j}_{v} \underbrace{a_{j+1} \dots a_{|z|}}_{w}$ 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 $p_i = p_j$.

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:

- $|uv| \leq n$,



## 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: $\underbrace{a_1 \dots a_i}_{u} \underbrace{a_{i+1} \dots a_j}_{v} \underbrace{a_{j+1} \dots a_{|z|}}_{w}$

i

Damit gilt:

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

u v v 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 $p_i = p_j$.

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
- $\forall l \geq 0.\ uv^l w \in R$.

□

[Darf $A$ NFA sein?]

Fazit:

Falls $L(M) = R$ so ist $|Q_M|$ eine Pumping-Lemma-Zahl für $R$.



Fazit:

Falls $L(M) = R$ so ist $|Q_M|$ eine Pumping-Lemma-Zahl für $R$.

Pumping-Lemma-Zahl für $L(ab^*c)$?



Fazit:

Falls $L(M) = R$ so ist $|Q_M|$ eine Pumping-Lemma-Zahl für $R$.

Pumping-Lemma-Zahl für $L(ab^*c)$?

Pumping-Lemma-Zahl für $\{aaaa\}$?



Fazit:

Falls $L(M) = R$ so ist $|Q_M|$ eine Pumping-Lemma-Zahl für $R$.

Pumping-Lemma-Zahl für $L(ab^*c)$?

Pumping-Lemma-Zahl für $\{aaaa\}$?

Ist $|Q_M| + 1$ auch eine Pumping-Lemma-Zahl für $R$?

Beispiel für die Anwendung des Pumping Lemmas:

**Satz 3.31**
*Die Sprache* $\{a^i b^i \mid i \in \mathbb{N}\}$ *ist nicht regulär.*