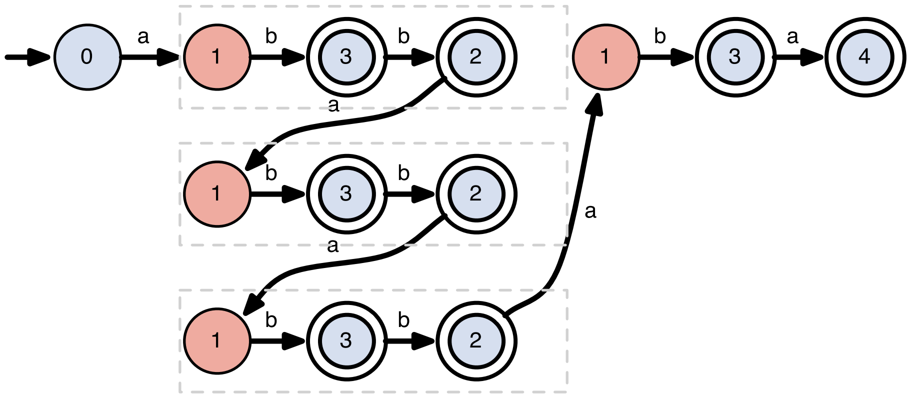
Berechenbarkeitstheorie

3. Vorlesung



Dr. Franziska Jahnke
Institut für Mathematische Logik und Grundlagenforschung
WWU Münster

NEAs

Definition

Ein **Nichtdeterministischer Endlicher Automat (NEA)** ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$, mit:

- $|Q| < \infty$, $|\Sigma| < \infty$,
- $\delta \colon Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ (möglicherweise partiell!),
- $q_0 \in Q$, $F \subseteq Q$.

NEAs

Definition

Ein **Nichtdeterministischer Endlicher Automat (NEA)** ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$, mit:

- $|Q| < \infty$, $|\Sigma| < \infty$,
- $\delta\colon Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ (möglicherweise partiell!),
- $q_0 \in Q$, $F \subseteq Q$.

Ein Wort kann verschiedene Läufe haben !!

NEAs

Definition

Ein **Nichtdeterministischer Endlicher Automat (NEA)** ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$, mit:

- $|Q| < \infty$, $|\Sigma| < \infty$,
- $\delta\colon Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ (möglicherweise partiell!),
- $q_0 \in Q$, $F \subseteq Q$.

Ein Wort kann verschiedene Läufe haben !!

Akzeptanz eines Wortes (NEA)

Ein NEA N **akzeptiert ein Wort** $w \in \Sigma^*$ genau dann, wenn **ein** w-Lauf von N in einem akzeptierenden Zustand endet.

NEAs

Definition

Ein **Nichtdeterministischer Endlicher Automat (NEA)** ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$, mit:

- $|Q| < \infty$, $|\Sigma| < \infty$,
- $\delta \colon Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ (möglicherweise partiell!),
- $q_0 \in Q$, $F \subseteq Q$.

Ein Wort kann verschiedene Läufe haben !!

Akzeptanz eines Wortes (NEA)

Ein NEA N **akzeptiert ein Wort** $w \in \Sigma^*$ genau dann, wenn **ein** w-Lauf von N in einem akzeptierenden Zustand endet.

Erkennen einer Sprache (NEA)

Ein NEA N **erkennt (akzeptiert) die Sprache**

$$L(N) := \{w \in \Sigma^* \mid N \text{ akzeptiert } w\}.$$

Iterierter Übergang beim NEA

Iterierter Übergang beim NEA

ε-Abschluss

(wo komme ich hin ohne ein Zeichen zu lesen?)

$E(p \in Q) := \{q \in Q \mid q$ über ≥ 0 εKanten von p erreichbar$\}$

$E(P \subseteq Q) := \bigcup_{p \in P} E(p)$

Iterierter Übergang beim NEA

ε-Abschluss

(wo komme ich hin ohne ein Zeichen zu lesen?)

$E(p \in Q) := \{q \in Q \mid q$ über ≥ 0 εKanten von p erreichbar$\}$

$E(P \subseteq Q) := \bigcup_{p \in P} E(p)$

Definition

Für einen NEA $N = (Q, \Sigma, \delta, q_0, F)$ ist die **iterierte Übergangsfunktion** $\delta^k \colon Q \times \Sigma^k \to \mathcal{P}(Q)$ definiert durch:

1. $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
2. $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Iterierter Übergang beim NEA

ε-Abschluss

(wo komme ich hin ohne ein Zeichen zu lesen?)

$E(p \in Q) := \{q \in Q \mid q$ über ≥ 0 εKanten von p erreichbar$\}$

$E(P \subseteq Q) := \bigcup_{p \in P} E(p)$

Definition

Für einen NEA $N = (Q, \Sigma, \delta, q_0, F)$ ist die **iterierte Übergangsfunktion** $\delta^k \colon Q \times \Sigma^k \to \mathcal{P}(Q)$ definiert durch:

1. $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
2. $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Schreibweise: $\delta^*(q, w)$ für $\delta^{|w|}(q, w)$

Iterierter Übergang beim NEA

ε-Abschluss

(wo komme ich hin ohne ein Zeichen zu lesen?)

E … r$\}$

E …

$$L(N) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \emptyset\}$$

Definition

Für einen NEA $N = (Q, \Sigma, \delta, q_0, F)$ ist die **iterierte Übergangsfunktion** $\delta^k \colon Q \times \Sigma^k \to \mathcal{P}(Q)$ definiert durch:

1. $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
2. $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Schreibweise: $\delta^*(q, w)$ für $\delta^{|w|}(q, w)$

DEA vs. NEA

Satz 2

$$\{L \mid L \text{ wird durch NEA erkannt}\} = \text{REG}$$

DEA vs. NEA

Satz 2

$$\{L \mid L \text{ wird durch NEA erkannt}\} = \text{REG}$$

Potenzautomat

Für einen NEA $N = (Q, \Sigma, \delta, q_0, F)$ ist sein **Potenzautomat** $\text{Pot}(N) = (\widetilde{Q}, \Sigma, \widetilde{\delta}, \widetilde{q}_0, \widetilde{F})$ ein DEA mit

- Zustandsmenge $\widetilde{Q} = \mathcal{P}(Q)$,
- Übergangsfunktion $\widetilde{\delta}$, mit $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$,
- Startzustand $\widetilde{q}_0 = E(q_0)$,
- akzeptierenden Zuständen $\widetilde{F} = \{P \in \widetilde{Q} \mid P \cap F \neq \emptyset\}$.

DEA vs. NEA

Satz 2

$$\{L \mid L \text{ wird durch NEA erkannt}\} = \text{REG}$$

Potenzautomat

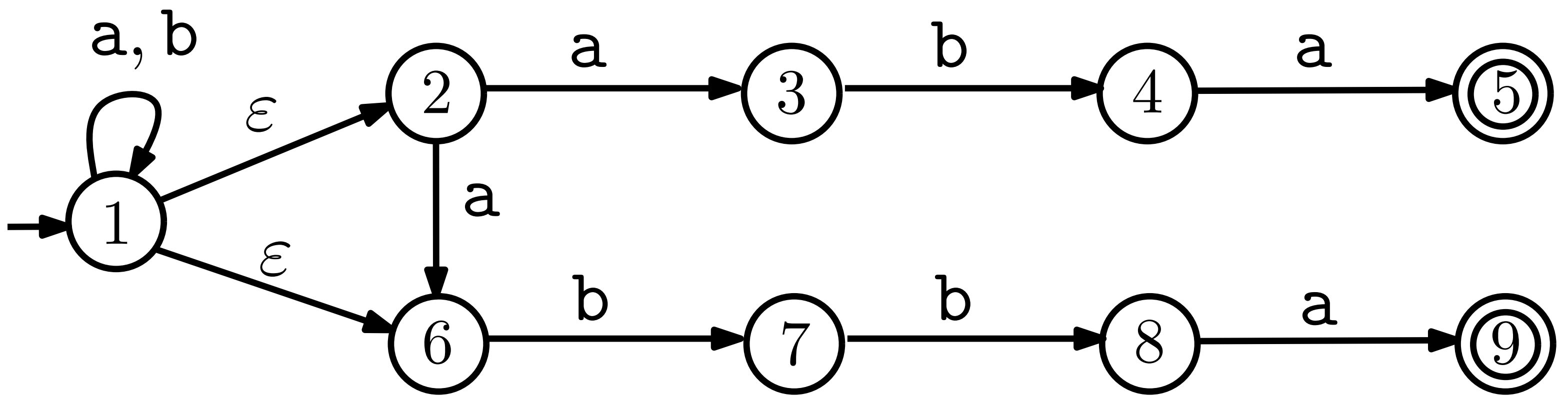
Für einen NEA $N = (Q, \Sigma, \delta, q_0, F)$ ist sein **Potenzautomat** $\text{Pot}(N) = (\widetilde{Q}, \Sigma, \widetilde{\delta}, \widetilde{q}_0, \widetilde{F})$ ein DEA mit

- Zustandsmenge $\widetilde{Q} = \mathcal{P}(Q)$,
- Übergangsfunktion $\widetilde{\delta}$, mit $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$,
- Startzustand $\widetilde{q}_0 = E(q_0)$,
- akzeptierenden Zuständen $\widetilde{F} = \{P \in \widetilde{Q} \mid P \cap F \neq \emptyset\}$.

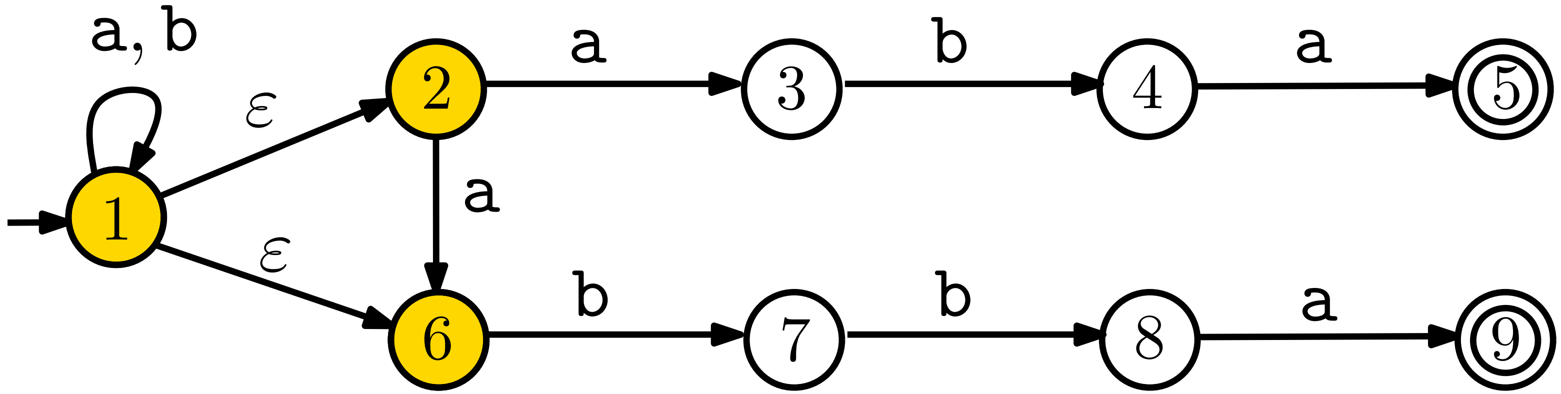
Behauptung

$$L(N) = L(\text{Pot}(N))$$

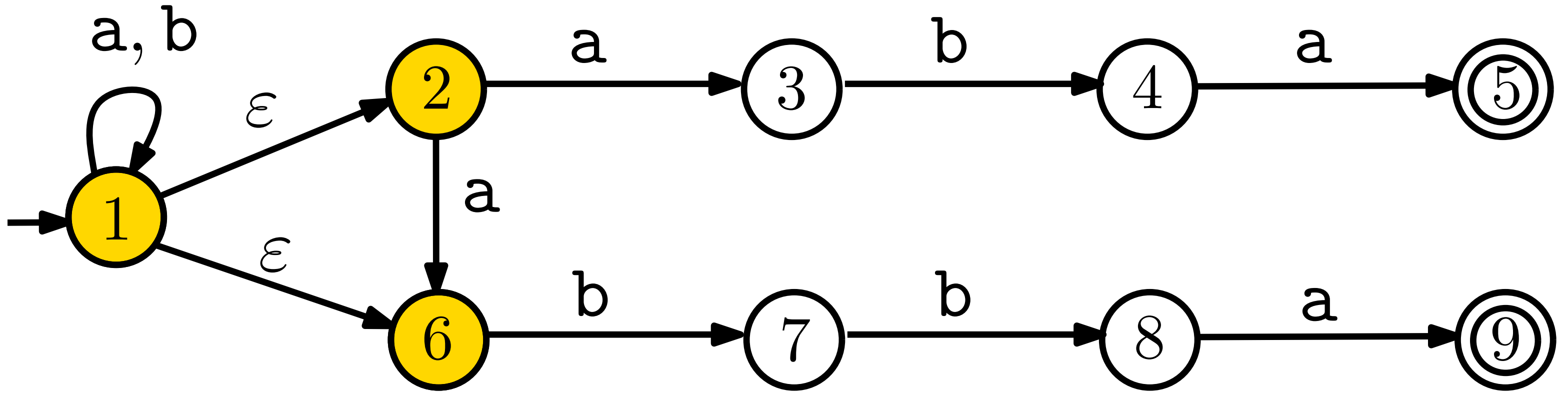
Beispiel Potenzautomat

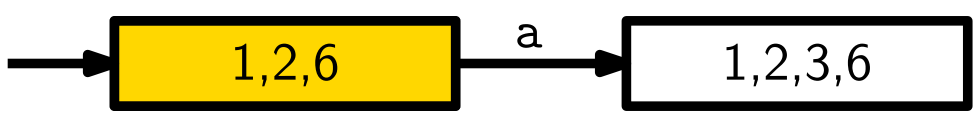


Beispiel Potenzautomat

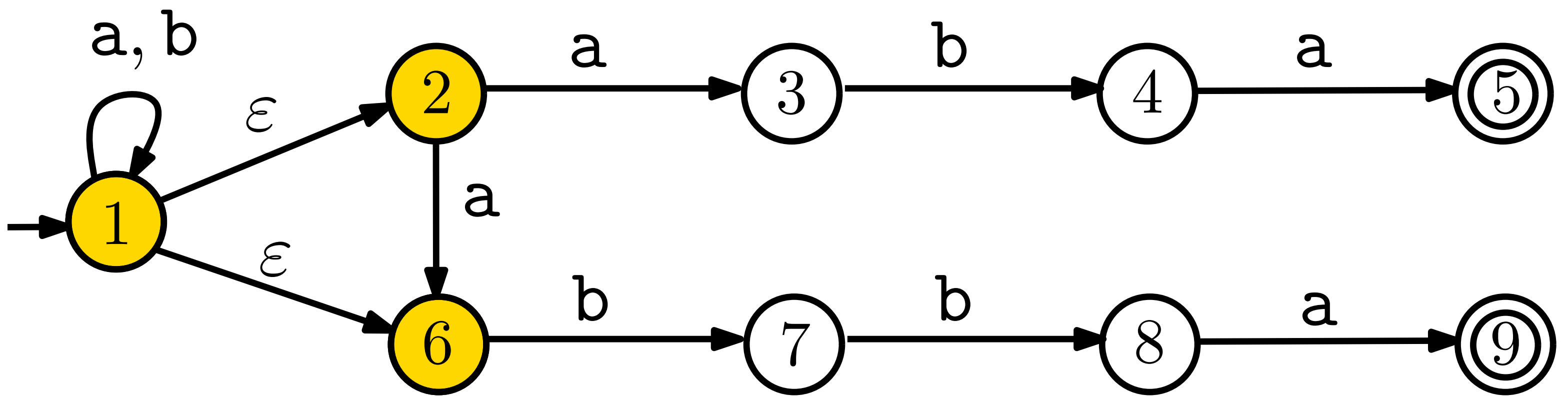


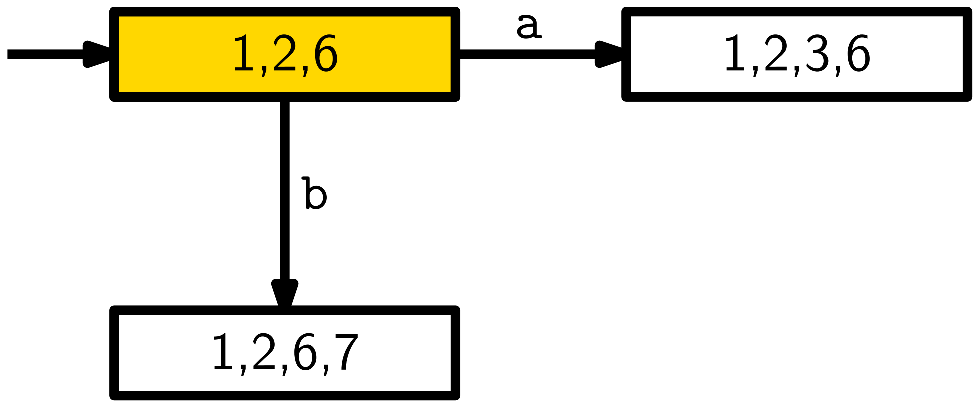
Beispiel Potenzautomat



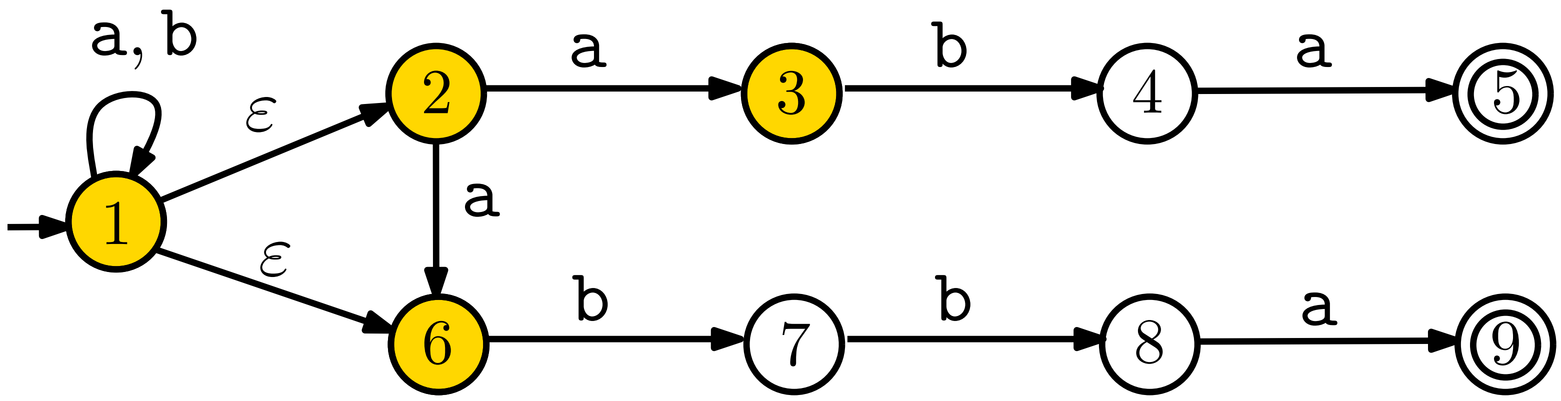


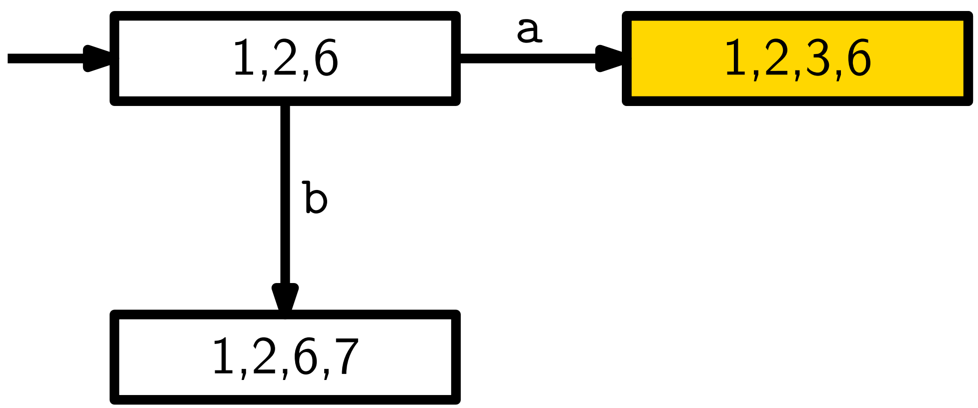
Beispiel Potenzautomat



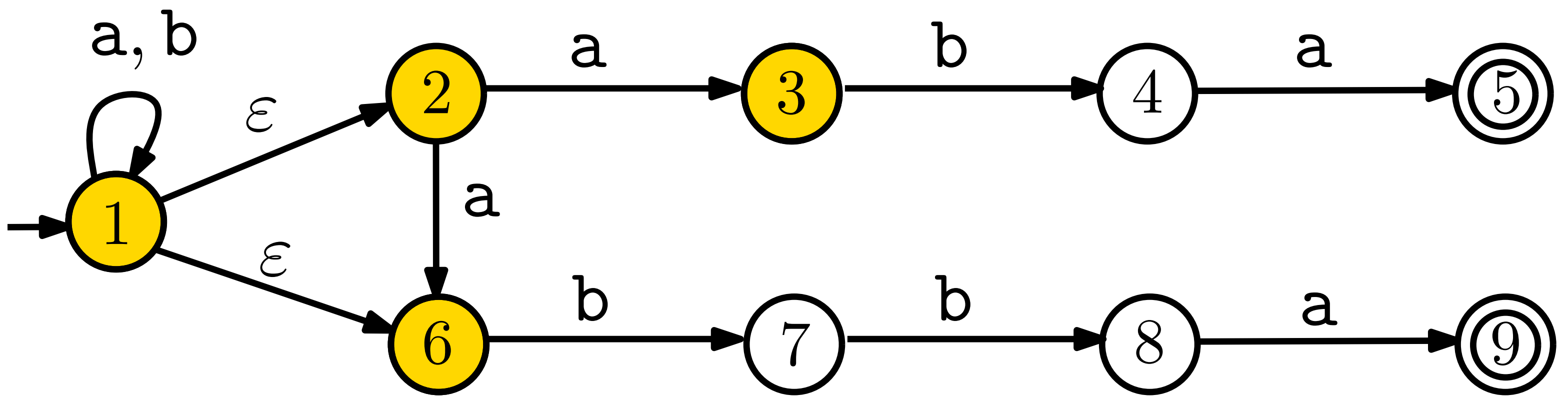


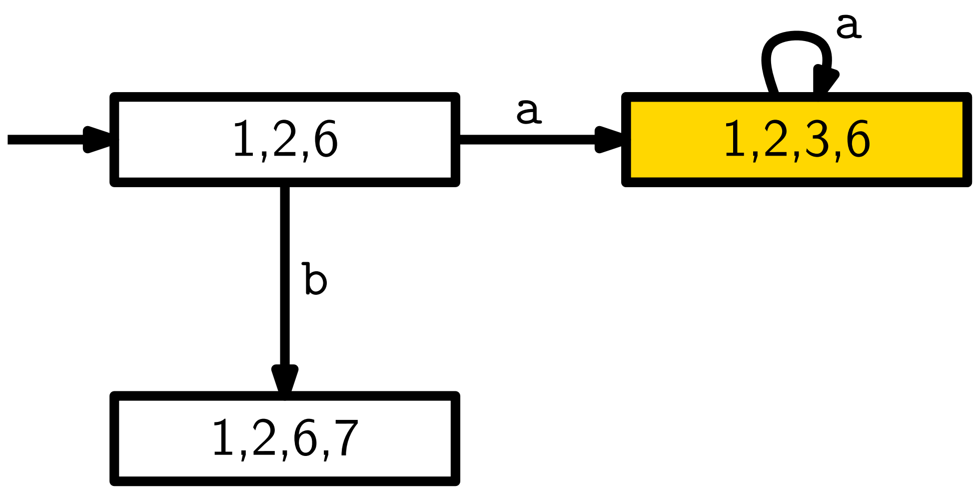
Beispiel Potenzautomat



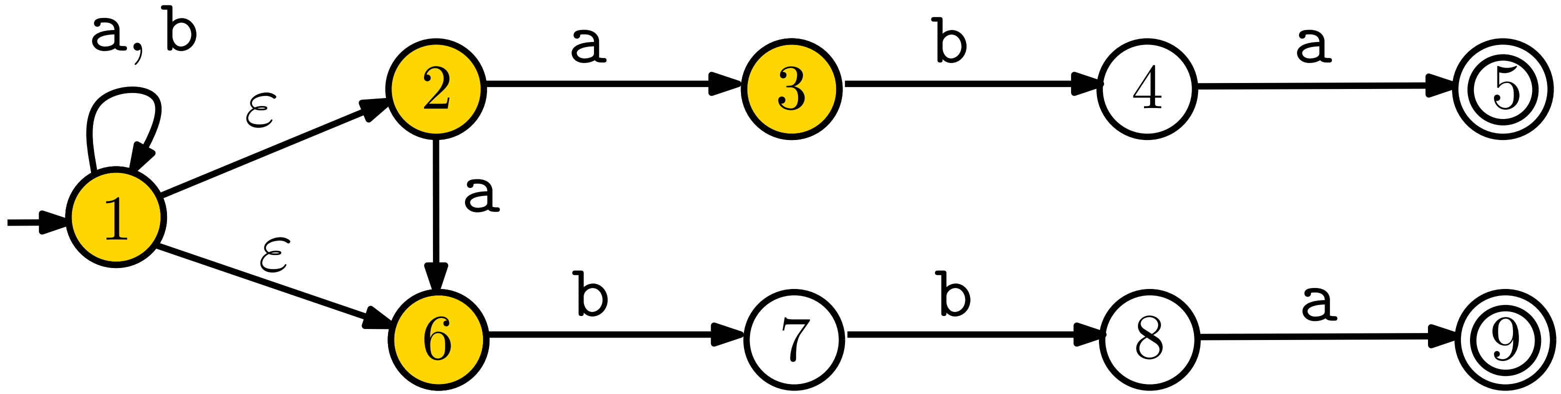


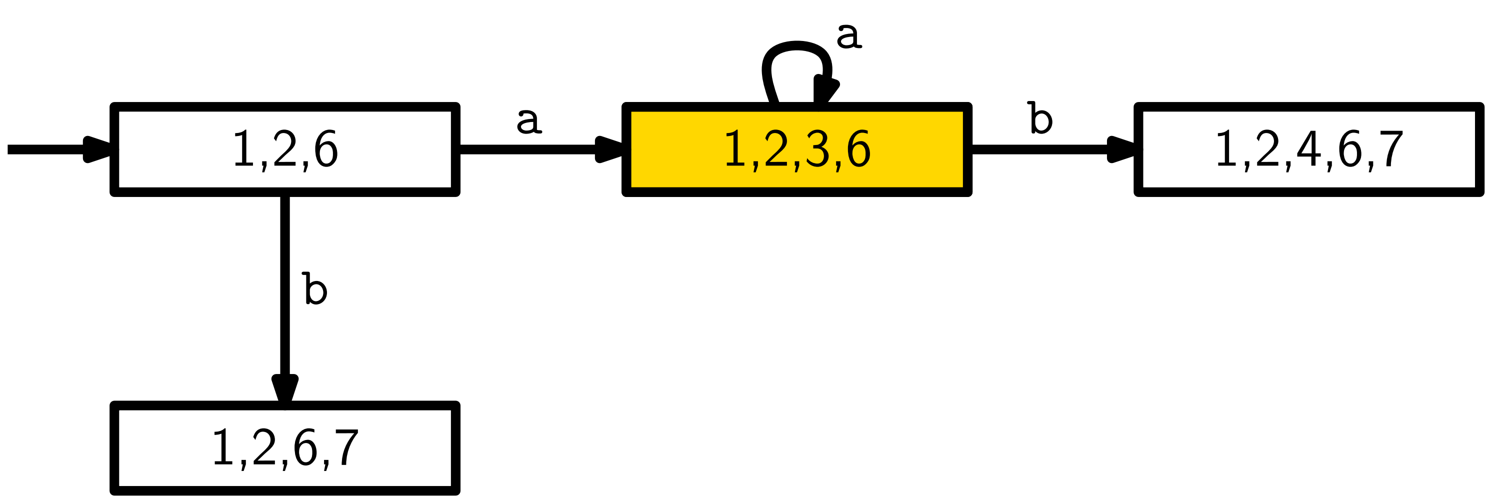
Beispiel Potenzautomat



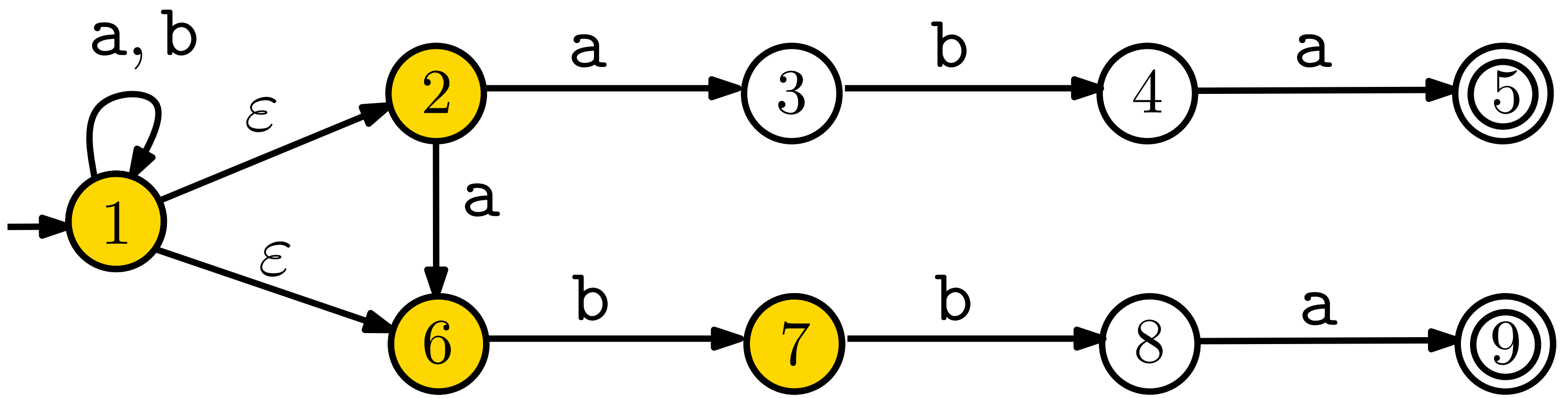


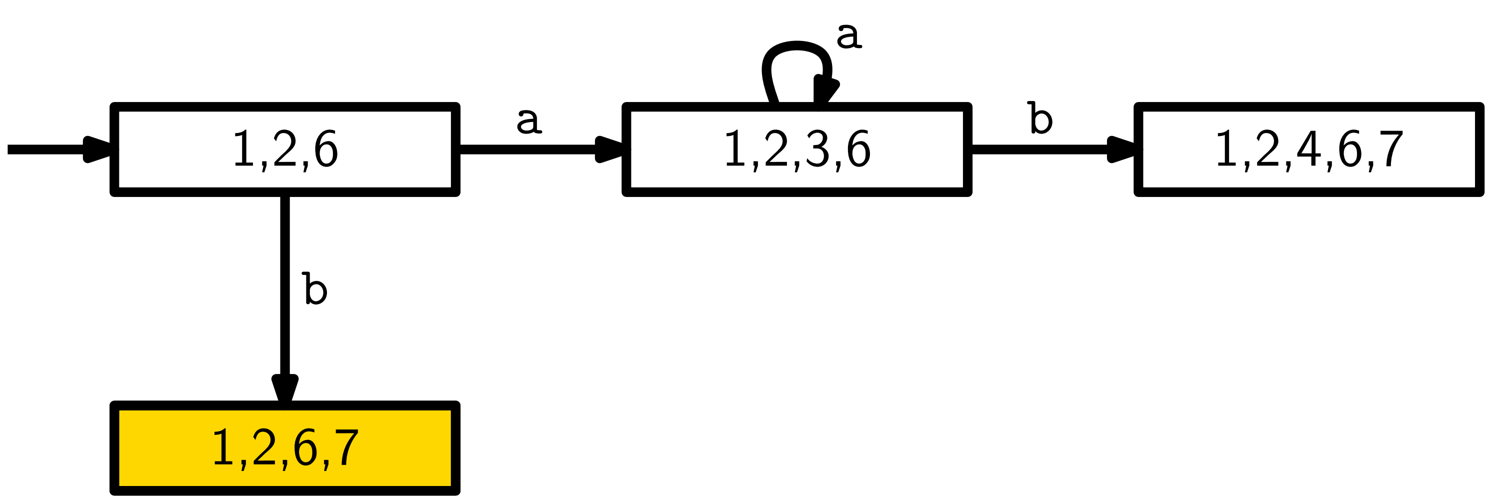
Beispiel Potenzautomat



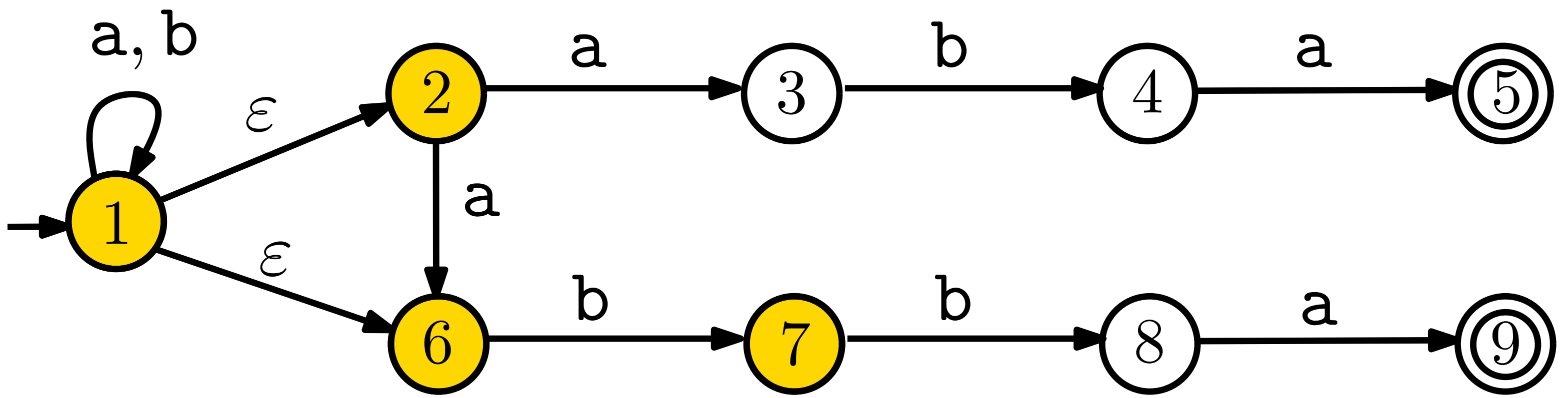


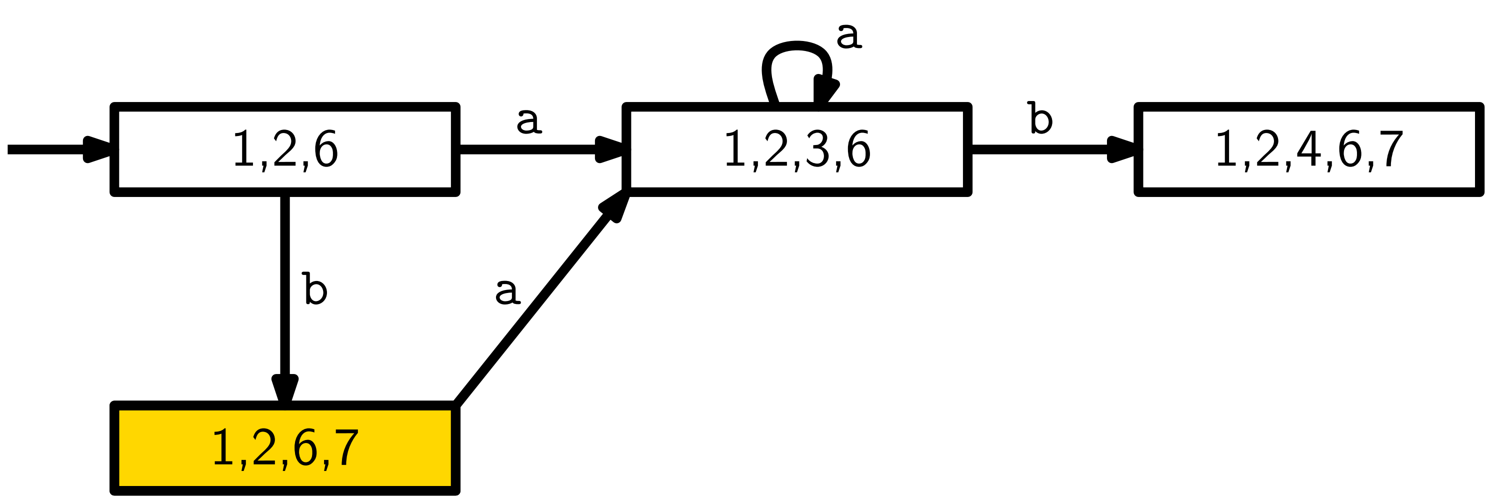
Beispiel Potenzautomat



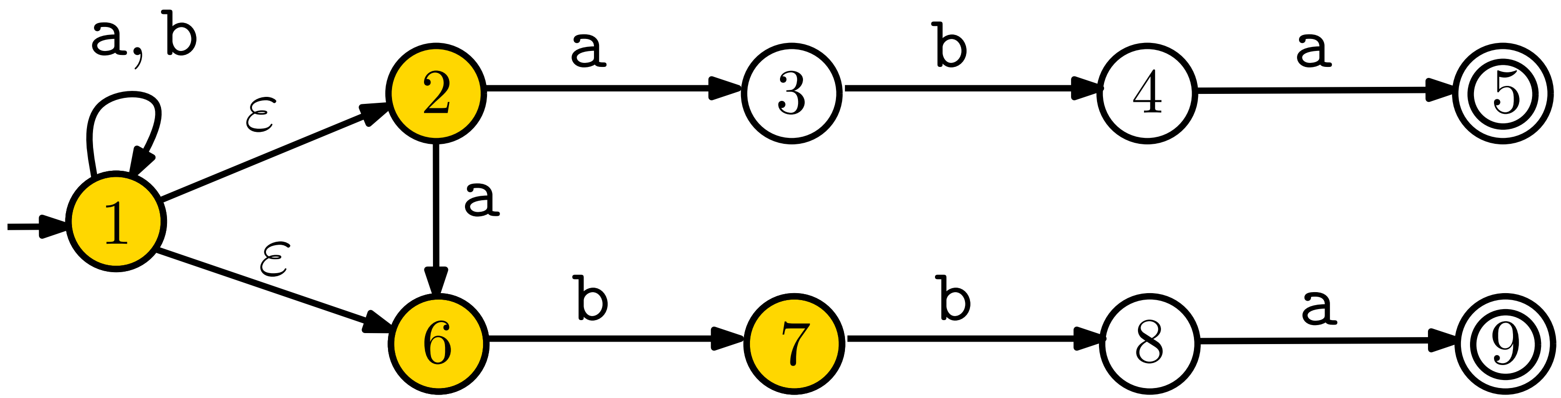


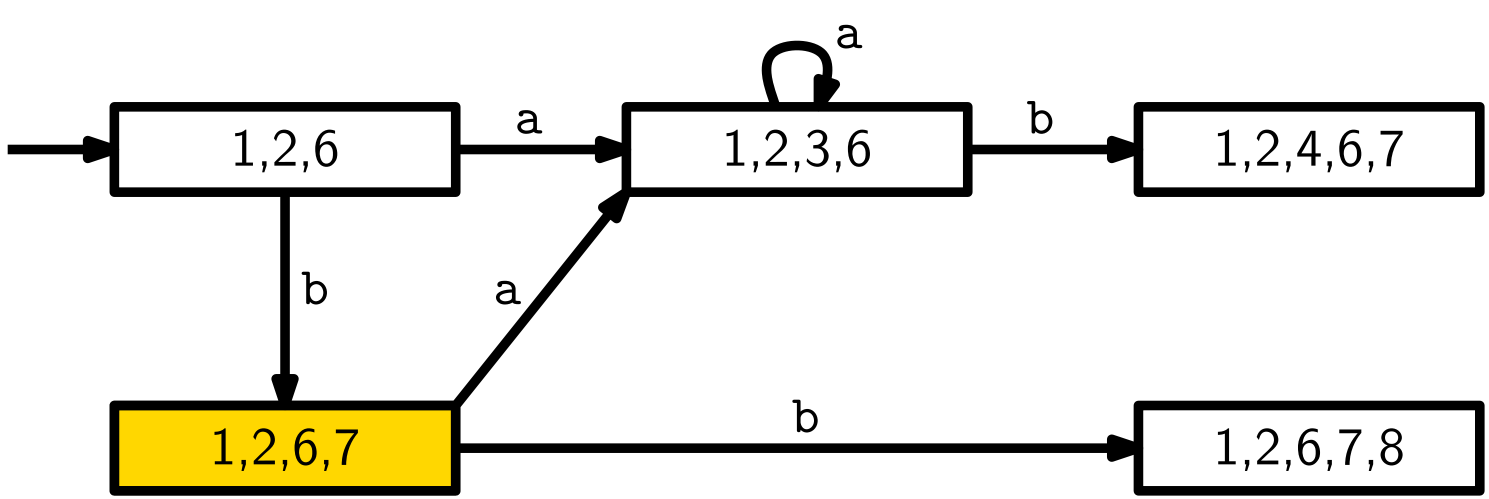
Beispiel Potenzautomat



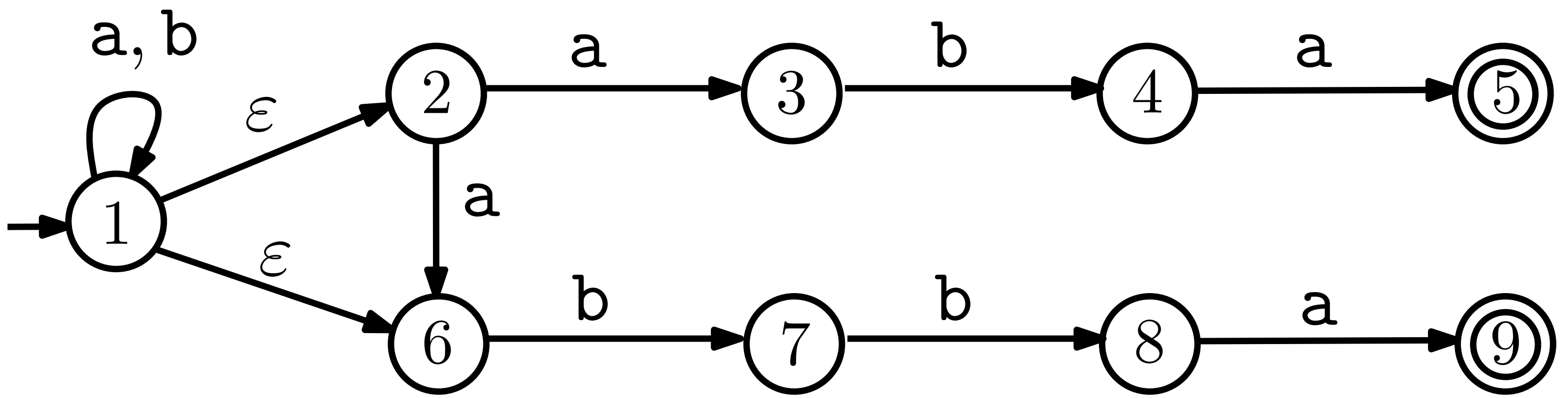


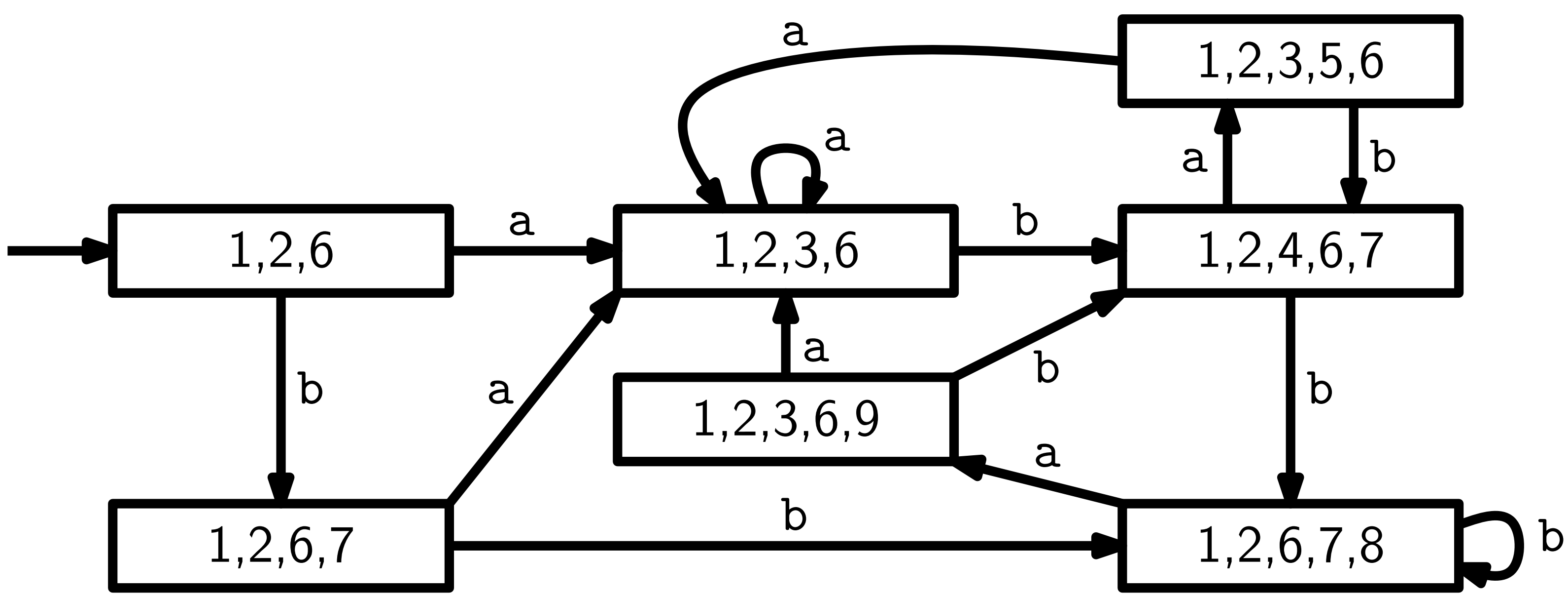
Beispiel Potenzautomat



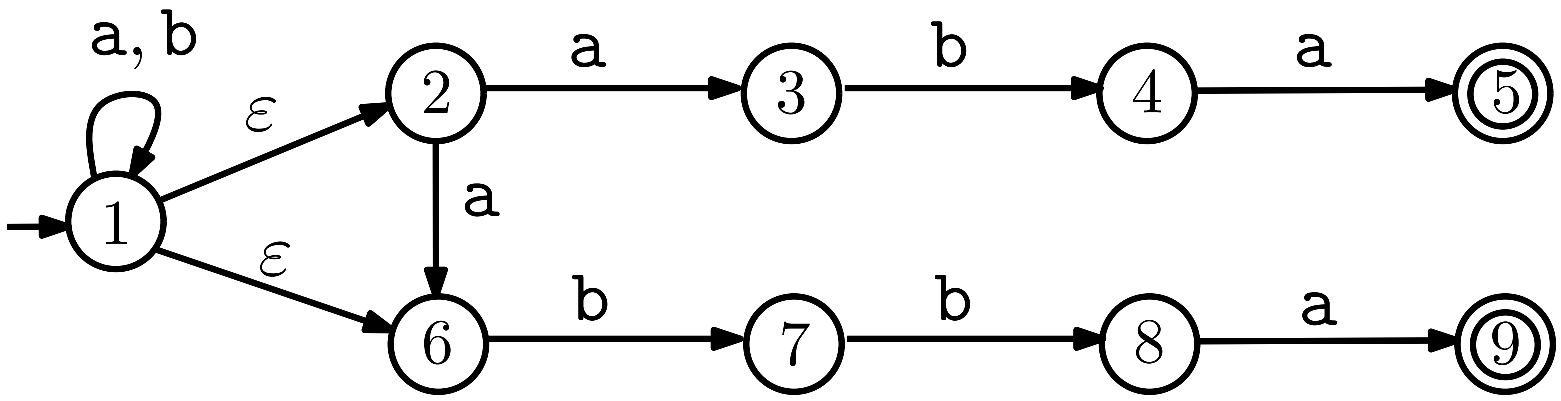


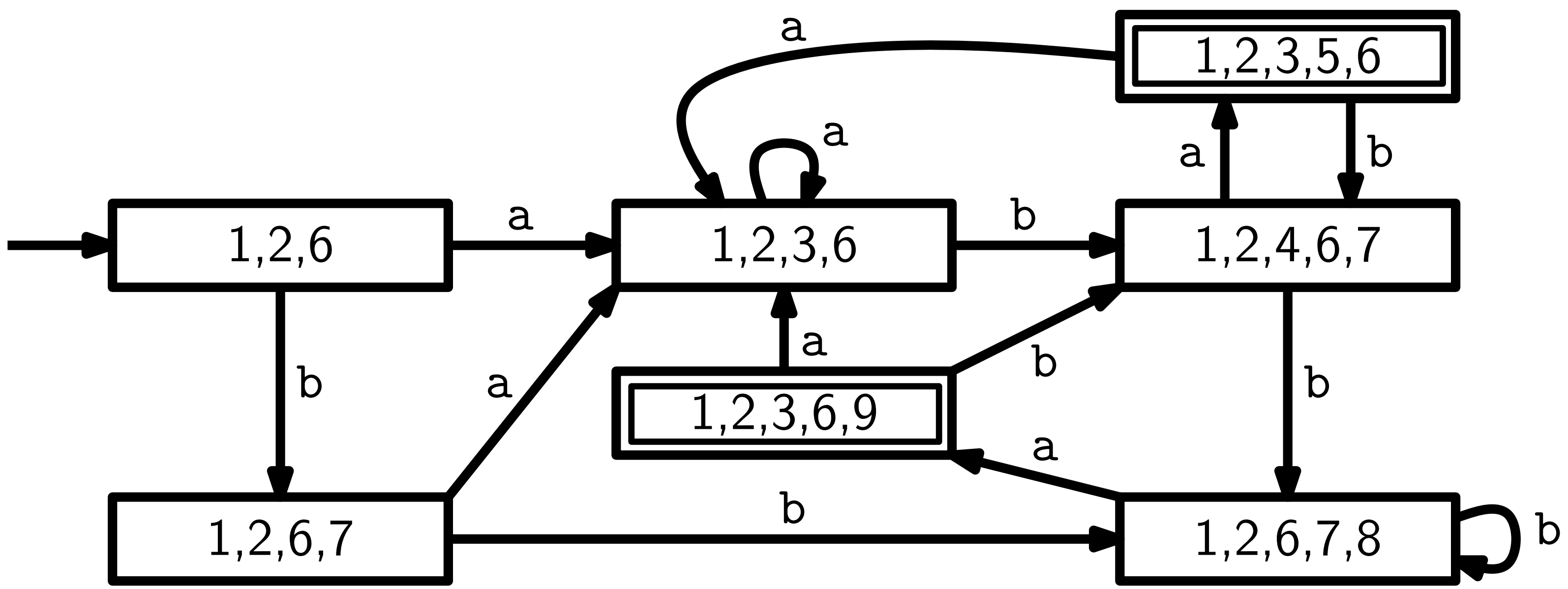
Beispiel Potenzautomat





Beispiel Potenzautomat





Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$,
(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$,
(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$, ← Def. δ^0 für DEA

(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$, ← Def. $\widetilde{\delta}$

(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$, ← Def. δ^i für NEA

(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Ind.anfang: $\widetilde{\delta}^0(E(q), \varepsilon) = E(q) = \delta^0(q, \varepsilon)$

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$,
(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$,
(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Ind.anfang: $\widetilde{\delta}^0(E(q), \varepsilon) \underset{(1)}{=} E(q) \underset{(3)}{=} \delta^0(q, \varepsilon)$

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$,
(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$,
(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Ind.anfang: $\widetilde{\delta}^0(E(q), \varepsilon) = E(q) = \delta^0(q, \varepsilon)$

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$,
(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$,
(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Ind.anfang: $\widetilde{\delta}^0(E(q), \varepsilon) = E(q) = \delta^0(q, \varepsilon)$

Ind.schritt: $\widetilde{\delta}^{k+1}(E(q), ux) = \widetilde{\delta}\left(\widetilde{\delta}^k(E(q), u), x\right)$ für $w = ux$, $x \in \Sigma$

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$,
(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$,
(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Ind.anfang: $\widetilde{\delta}^0(E(q), \varepsilon) = E(q) = \delta^0(q, \varepsilon)$

Ind.schritt: $\widetilde{\delta}^{k+1}(E(q), ux) = \widetilde{\delta}\left(\widetilde{\delta}^k(E(q), u), x\right)$ für $w = ux$, $x \in \Sigma$

$= \widetilde{\delta}\left(\delta^k(q, u), x\right)$ Induktionsannahme

---

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$,
(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$,
(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Ind.anfang: $\widetilde{\delta}^0(E(q), \varepsilon) = E(q) = \delta^0(q, \varepsilon)$

Ind.schritt:

$$\begin{aligned} \widetilde{\delta}^{k+1}(E(q), ux) &= \widetilde{\delta}\left(\widetilde{\delta}^k(E(q), u), x\right) && \text{für } w = ux,\ x \in \Sigma \\ &= \widetilde{\delta}\left(\delta^k(q, u), x\right) && \text{Induktionsannahme} \\ &= E\left(\bigcup_{p \in \delta^k(q,u)} \delta(p, x)\right) && (2) \end{aligned}$$

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$,
(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$,
(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Ind.anfang: $\widetilde{\delta}^0(E(q), \varepsilon) = E(q) = \delta^0(q, \varepsilon)$

Ind.schritt:

$$\begin{aligned}
\widetilde{\delta}^{k+1}(E(q), ux) &= \widetilde{\delta}\left(\widetilde{\delta}^k(E(q), u), x\right) && \text{für } w = ux,\ x \in \Sigma \\
&= \widetilde{\delta}\left(\delta^k(q, u), x\right) && \text{Induktionsannahme} \\
&= E\left(\bigcup_{p \in \delta^k(q,u)} \delta(p, x)\right) && (2) \\
&= \delta^{k+1}(q, ux) && (4)
\end{aligned}$$

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$,
(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$,
(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Lemma 1: $\widetilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Ind.anfang: $\widetilde{\delta}^0(E(q), \varepsilon) = E(q) = \delta^0(q, \varepsilon)$

Ind.schritt: $\widetilde{\delta}^{k+1}(E(q), ux) = \widetilde{\delta}\left(\widetilde{\delta}^k(E(q), u), x\right)$ für $w = ux$, $x \in \Sigma$

$$= \widetilde{\delta}\left(\delta^k(q, u), x\right) \qquad \text{Induktionsannahme}$$

$$= E\left(\bigcup_{p \in \delta^k(q,u)} \delta(p, x)\right) \qquad (2)$$

$$= \delta^{k+1}(q, ux) \qquad (4)$$

$\square$

Erinnerung:

(1) $\widetilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \widetilde{Q}$,
(2) $\widetilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \widetilde{Q}$,
(3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
(4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q,w)} \delta(r, x)\right)$ für alle $q \in Q$, $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Beweis Satz 2

Beweis Satz 2

Wir müssen zeigen $L(N) = L(\mathsf{Pot}(N))$ für jeden NEA N

$w \in \Sigma^*$ wird von N akzeptiert $\Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset$

Beweis Satz 2

Wir müssen zeigen $L(N) = L(\text{Pot}(N))$ für jeden NEA N

$$
\begin{aligned}
w \in \Sigma^* \text{ wird von } N \text{ akzeptiert} &\Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset \\
&\Leftrightarrow \widetilde{\delta}^*(E(q_0), w) \cap F \neq \emptyset
\end{aligned}
$$

(nach Lemma 1)

Beweis Satz 2

Wir müssen zeigen $L(N) = L(\mathsf{Pot}(N))$ für jeden NEA N

$$
\begin{aligned}
w \in \Sigma^* \text{ wird von } N \text{ akzeptiert} &\Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset \\
&\Leftrightarrow \widetilde{\delta}^*(E(q_0), w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad \text{(nach Lemma 1)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad \text{(nach Def. } \widetilde{q}_0\text{)}
\end{aligned}
$$

Beweis Satz 2

Wir müssen zeigen $L(N) = L(\mathsf{Pot}(N))$ für jeden NEA N

$$
\begin{aligned}
w \in \Sigma^* \text{ wird von } N \text{ akzeptiert} &\Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset \\
&\Leftrightarrow \widetilde{\delta}^*(E(q_0), w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad\qquad \text{(nach Lemma 1)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad\qquad \text{(nach Def. } \widetilde{q}_0\text{)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \in \widetilde{F} \\
&\qquad\qquad\qquad\qquad \text{(nach Def. } \widetilde{F}\text{)}
\end{aligned}
$$

Beweis Satz 2

Wir müssen zeigen $L(N) = L(\text{Pot}(N))$ für jeden NEA N

$$
\begin{aligned}
w \in \Sigma^* \text{ wird von } N \text{ akzeptiert} &\Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset \\
&\Leftrightarrow \widetilde{\delta}^*(E(q_0), w) \cap F \neq \emptyset \\
&\qquad\qquad \text{(nach Lemma 1)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \cap F \neq \emptyset \\
&\qquad\qquad \text{(nach Def. } \widetilde{q}_0\text{)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \in \widetilde{F} \\
&\qquad\qquad \text{(nach Def. } \widetilde{F}\text{)} \\
&\Leftrightarrow \text{Pot}(N) \text{ akzeptiert } w
\end{aligned}
$$

Beweis Satz 2

Wir müssen zeigen $L(N) = L(\text{Pot}(N))$ für jeden NEA N

$$
\begin{aligned}
w \in \Sigma^* \text{ wird von } N \text{ akzeptiert} &\Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset \\
&\Leftrightarrow \widetilde{\delta}^*(E(q_0), w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad\qquad \text{(nach Lemma 1)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad\qquad \text{(nach Def. } \widetilde{q}_0\text{)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \in \widetilde{F} \\
&\qquad\qquad\qquad\qquad \text{(nach Def. } \widetilde{F}\text{)} \\
&\Leftrightarrow \text{Pot}(N) \text{ akzeptiert } w
\end{aligned}
$$

Also:
L wird durch NEA erkannt $\rightarrow$ L wird von DEA erkannt $\rightarrow$ $L \in$ REG

Beweis Satz 2

Wir müssen zeigen $L(N) = L(\text{Pot}(N))$ für jeden NEA N

$$
\begin{aligned}
w \in \Sigma^* \text{ wird von } N \text{ akzeptiert} &\Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset \\
&\Leftrightarrow \widetilde{\delta}^*(E(q_0), w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad\qquad \text{(nach Lemma 1)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad\qquad \text{(nach Def. } \widetilde{q}_0\text{)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \in \widetilde{F} \\
&\qquad\qquad\qquad\qquad \text{(nach Def. } \widetilde{F}\text{)} \\
&\Leftrightarrow \text{Pot}(N) \text{ akzeptiert } w
\end{aligned}
$$

Also:
L wird durch NEA erkannt $\rightarrow$ L wird von DEA erkannt $\rightarrow$ $L \in$ REG

Es folgt:

$\{L \mid L \text{ wird durch NEA erkannt}\} = \text{REG}$

Beweis Satz 2

Wir müssen zeigen $L(N) = L(\text{Pot}(N))$ für jeden NEA N

$$
\begin{aligned}
w \in \Sigma^* \text{ wird von } N \text{ akzeptiert} &\Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset \\
&\Leftrightarrow \widetilde{\delta}^*(E(q_0), w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad\qquad \text{(nach Lemma 1)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \cap F \neq \emptyset \\
&\qquad\qquad\qquad\qquad \text{(nach Def. } \widetilde{q}_0\text{)} \\
&\Leftrightarrow \widetilde{\delta}^*(\widetilde{q}_0, w) \in \widetilde{F} \\
&\qquad\qquad\qquad\qquad \text{(nach Def. } \widetilde{F}\text{)} \\
&\Leftrightarrow \text{Pot}(N) \text{ akzeptiert } w
\end{aligned}
$$

Also:
L wird durch NEA erkannt $\rightarrow$ L wird von DEA erkannt $\rightarrow$ $L \in \text{REG}$

Es folgt:

$\{L \mid L \text{ wird durch NEA erkannt}\} = \text{REG}$

□

Zusammenfassung

Reguläre Sprachen

Zusammenfassung

vom **DEA** erkannt (nach Def.)

Reguläre Sprachen

Zusammenfassung

Reguläre Sprachen ← vom **DEA** erkannt (nach Def.)

↕

äquivalent zum **NEA**

Berechnungsmodell mit

- Nichtdeterminismus
- ε-Übergängen

Zusammenfassung

Reguläre Sprachen ← vom **DEA** erkannt (nach Def.)

Potenzautomat

äquivalent zum **NEA**

Berechnungsmodell mit

- Nichtdeterminismus
- ε-Übergängen

Abschlusseigenschaften von REG

Abschlusseigenschaften von REG

$\rightarrow$ die Sprachklasse $\mathcal{L}$ ist abgeschlossen unter der Operation $\bowtie$, wenn für alle $L_1, L_2 \in \mathcal{L}$ auch $L_1 \bowtie L_2 \in \mathcal{L}$.

Abschlusseigenschaften von REG

$\rightarrow$ die Sprachklasse $\mathcal{L}$ ist abgeschlossen unter der Operation $\bowtie$, wenn für alle $L_1, L_2 \in \mathcal{L}$ auch $L_1 \bowtie L_2 \in \mathcal{L}$.

Satz 3

Die regulären Sprachen sind abgeschlossen unter binärer Vereinigung.
$(L_1, L_2 \in \text{REG} \Rightarrow L_1 \cup L_2 \in \text{REG})$

Abschlusseigenschaften von REG

$\rightarrow$ die Sprachklasse $\mathcal{L}$ ist abgeschlossen unter der Operation $\bowtie$, wenn für alle $L_1, L_2 \in \mathcal{L}$ auch $L_1 \bowtie L_2 \in \mathcal{L}$.

Satz 3

Die regulären Sprachen sind abgeschlossen unter binärer Vereinigung.
($L_1, L_2 \in \mathrm{REG} \Rightarrow L_1 \cup L_2 \in \mathrm{REG}$)

Beweis 1 für Satz 3

Abschlusseigenschaften von REG

$\rightarrow$ die Sprachklasse $\mathcal{L}$ ist abgeschlossen unter der Operation $\bowtie$, wenn für alle $L_1, L_2 \in \mathcal{L}$ auch $L_1 \bowtie L_2 \in \mathcal{L}$.

Satz 3

Die regulären Sprachen sind abgeschlossen unter binärer Vereinigung.
($L_1, L_2 \in$ REG $\Rightarrow L_1 \cup L_2 \in$ REG)

Beweis 1 für Satz 3

- wenn $L_1, L_2 \in$ REG, dann existieren DEAs M_1 und M_2 mit $L(M_1) = L_1$ und $L(M_2) = L_2$

Abschlusseigenschaften von REG

$\rightarrow$ die Sprachklasse $\mathcal{L}$ ist abgeschlossen unter der Operation $\bowtie$, wenn für alle $L_1, L_2 \in \mathcal{L}$ auch $L_1 \bowtie L_2 \in \mathcal{L}$.

Satz 3

Die regulären Sprachen sind abgeschlossen unter binärer Vereinigung.
($L_1, L_2 \in$ REG $\Rightarrow L_1 \cup L_2 \in$ REG)

Beweis 1 für Satz 3

- wenn $L_1, L_2 \in$ REG, dann existieren DEAs M_1 und M_2 mit $L(M_1) = L_1$ und $L(M_2) = L_2$
- Konstruiere DEA M_3, der M_1 und M_2 simuliert und genau dann akzeptiert, wenn einer der beiden DEAs akzeptiert

Simulation von L_1 und L_2

Simulation von L_1 und L_2

- sei $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$

Simulation von L_1 und L_2

- sei $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$
- wir konstruieren $M_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$

Simulation von L_1 und L_2

- sei $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$
- wir konstruieren $M_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$
- **Idee:**
  Zustände von M_3 sind Paare von Zuständen $Q_1 \times Q_2$

Simulation von L_1 und L_2

- sei $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$
- wir konstruieren $M_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$
- **Idee:**
  Zustände von M_3 sind Paare von Zuständen $Q_1 \times Q_2$

  $\rightarrow$ M_3 ist genau dann im Zustand (q_a, q_b) wenn M_1 im Zustand q_a und M_2 im Zustand q_b sind (Invariante)

Simulation von L_1 und L_2

- sei $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$
- wir konstruieren $M_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$
- **Idee:**
  Zustände von M_3 sind Paare von Zuständen $Q_1 \times Q_2$

  $\rightarrow$ M_3 ist genau dann im Zustand (q_a, q_b) wenn M_1 im Zustand q_a und M_2 im Zustand q_b sind (Invariante)
- **Formal:**
  - $Q_3 := Q_1 \times Q_2$,
  - $\delta_3((q_1, q_2), a) := (q_1', q_2')$, g.d.w. $\delta_1(q_1, a) = q_1'$ und $\delta_2(q_2, a) = q_2'$
  - $q_{03} := (q_{01}, q_{02})$

Simulation von L_1 und L_2

- sei $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$
- wir konstruieren $M_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$
- **Idee:**
  Zustände von M_3 sind Paare von Zuständen $Q_1 \times Q_2$

  $\rightarrow$ M_3 ist genau dann im Zustand (q_a, q_b) wenn M_1 im Zustand q_a und M_2 im Zustand q_b sind (Invariante)
- **Formal:**
  - $Q_3 := Q_1 \times Q_2$,
  - $\delta_3((q_1, q_2), a) := (q_1', q_2')$, g.d.w. $\delta_1(q_1, a) = q_1'$ und $\delta_2(q_2, a) = q_2'$
  - $q_{03} := (q_{01}, q_{02})$

  $\rightarrow$ garantiert Invariante (ohne formalen Beweis)

Simulation von L_1 und L_2

- sei $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$
- wir konstruieren $M_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$
- **Idee:**
  Zustände von M_3 sind Paare von Zuständen $Q_1 \times Q_2$

  $\rightarrow$ M_3 ist genau dann im Zustand (q_a, q_b) wenn M_1 im Zustand q_a und M_2 im Zustand q_b sind (Invariante)
- **Formal:**
  - $Q_3 := Q_1 \times Q_2$,
  - $\delta_3((q_1, q_2), a) := (q_1', q_2')$, g.d.w. $\delta_1(q_1, a) = q_1'$ und $\delta_2(q_2, a) = q_2'$
  - $q_{03} := (q_{01}, q_{02})$

  $\rightarrow$ garantiert Invariante (ohne formalen Beweis)
- **Akz. Zustände:** akzeptiere, wenn M_1 oder M_2 akzeptiert
  - $F_3 := \{(q_1, q_2) \mid q_1 \in F_1 \text{ oder } q_2 \in F_2\}$

Simulation von L_1 und L_2

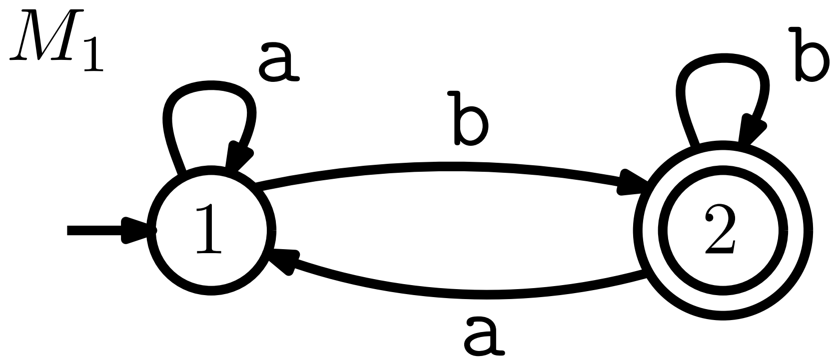
- sei $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$
- wir konstruieren $M_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$
- **Idee:**
  Zustände von M_3 sind Paare von Zuständen $Q_1 \times Q_2$

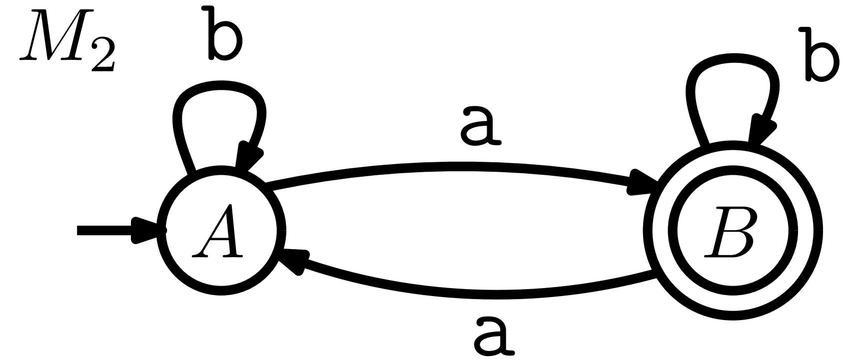
  $\rightarrow$ M_3 ist genau dann im Zustand (q_a, q_b) wenn M_1 im Zustand q_a und M_2 im Zustand q_b sind (Invariante)
- **Formal:**
  - $Q_3 := Q_1 \times Q_2$,
  - $\delta_3((q_1, q_2), a) := (q_1', q_2')$, g.d.w. $\delta_1(q_1, a) = q_1'$ und $\delta_2(q_2, a) = q_2'$
  - $q_{03} := (q_{01}, q_{02})$

  $\rightarrow$ garantiert Invariante (ohne formalen Beweis)
- **Akz. Zustände:** akzeptiere, wenn M_1 oder M_2 akzeptiert
  - $F_3 := \{(q_1, q_2) \mid q_1 \in F_1 \text{ oder } q_2 \in F_2\}$ $\square$

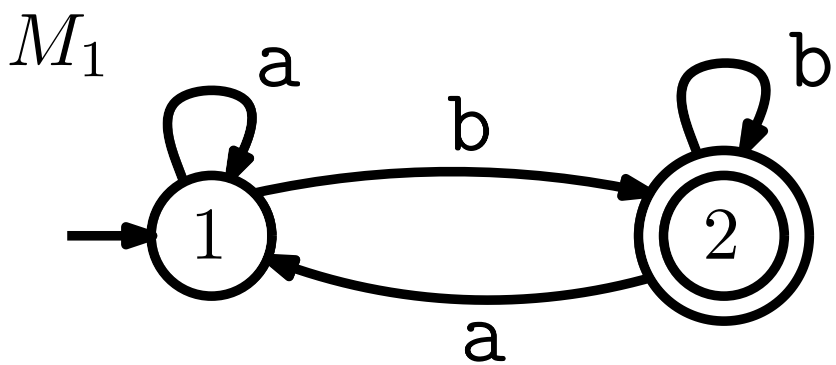
Beispiel

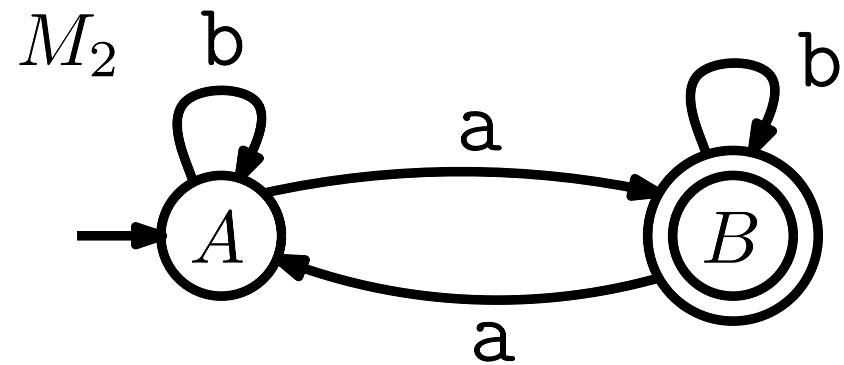
Beispiel

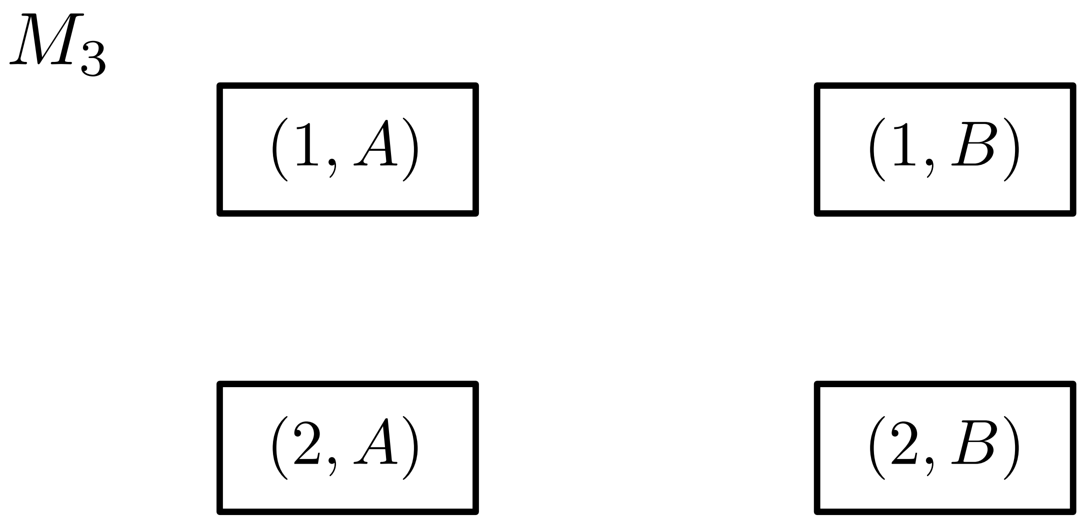




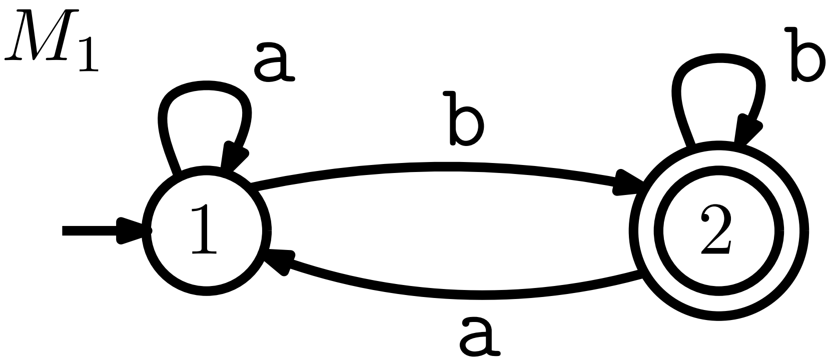
Beispiel

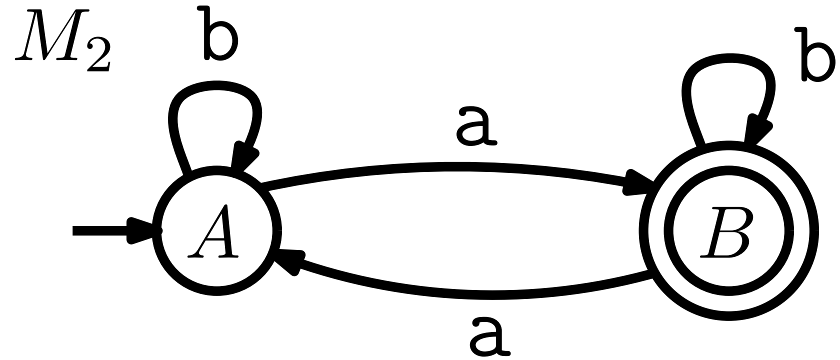


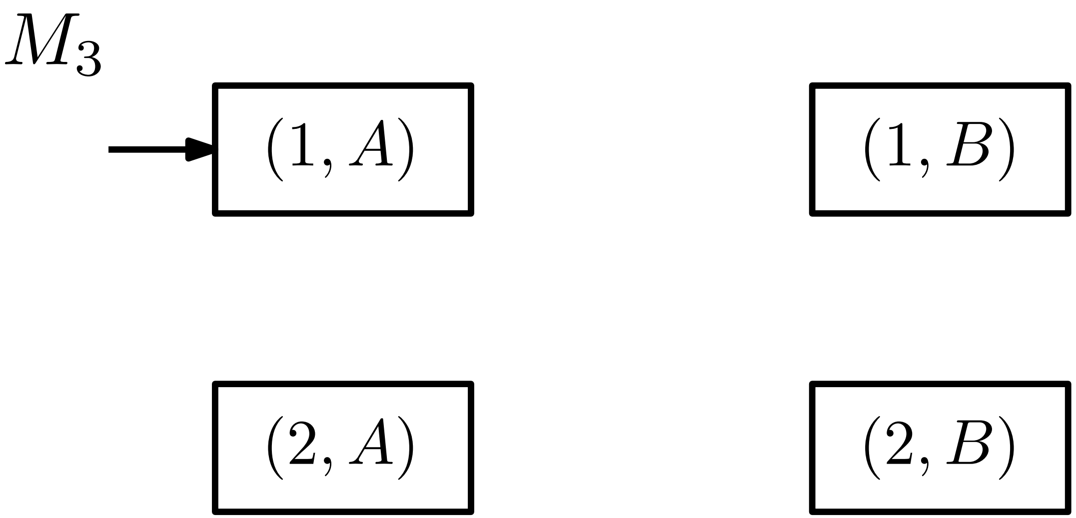




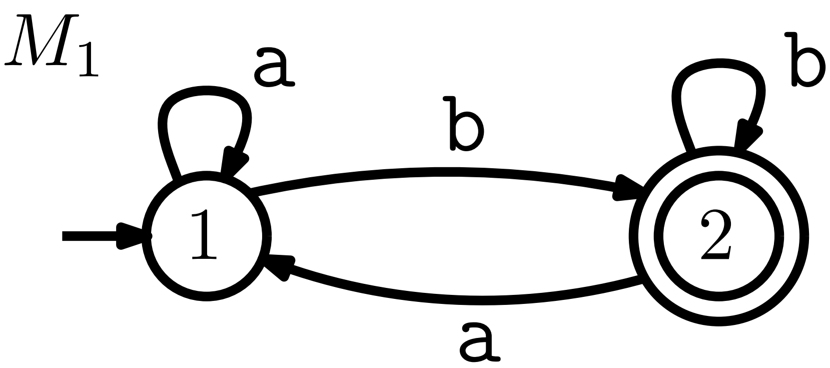
Beispiel

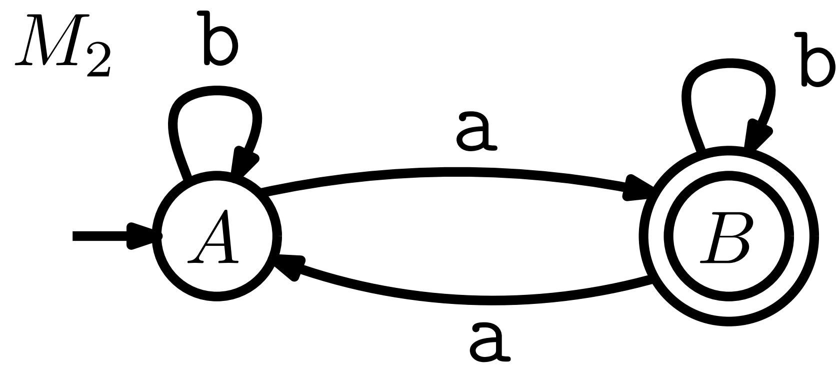


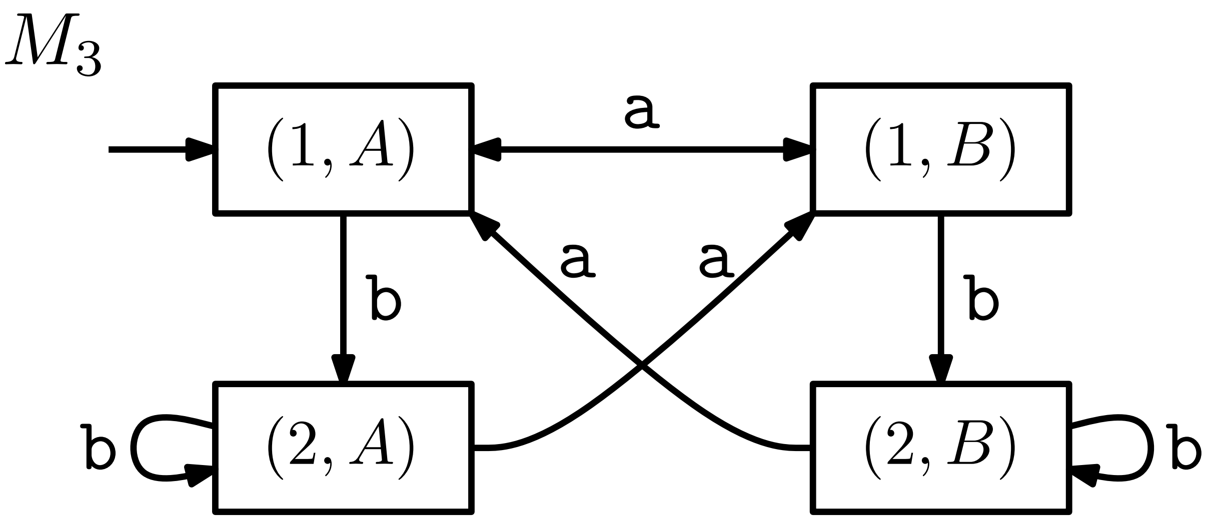




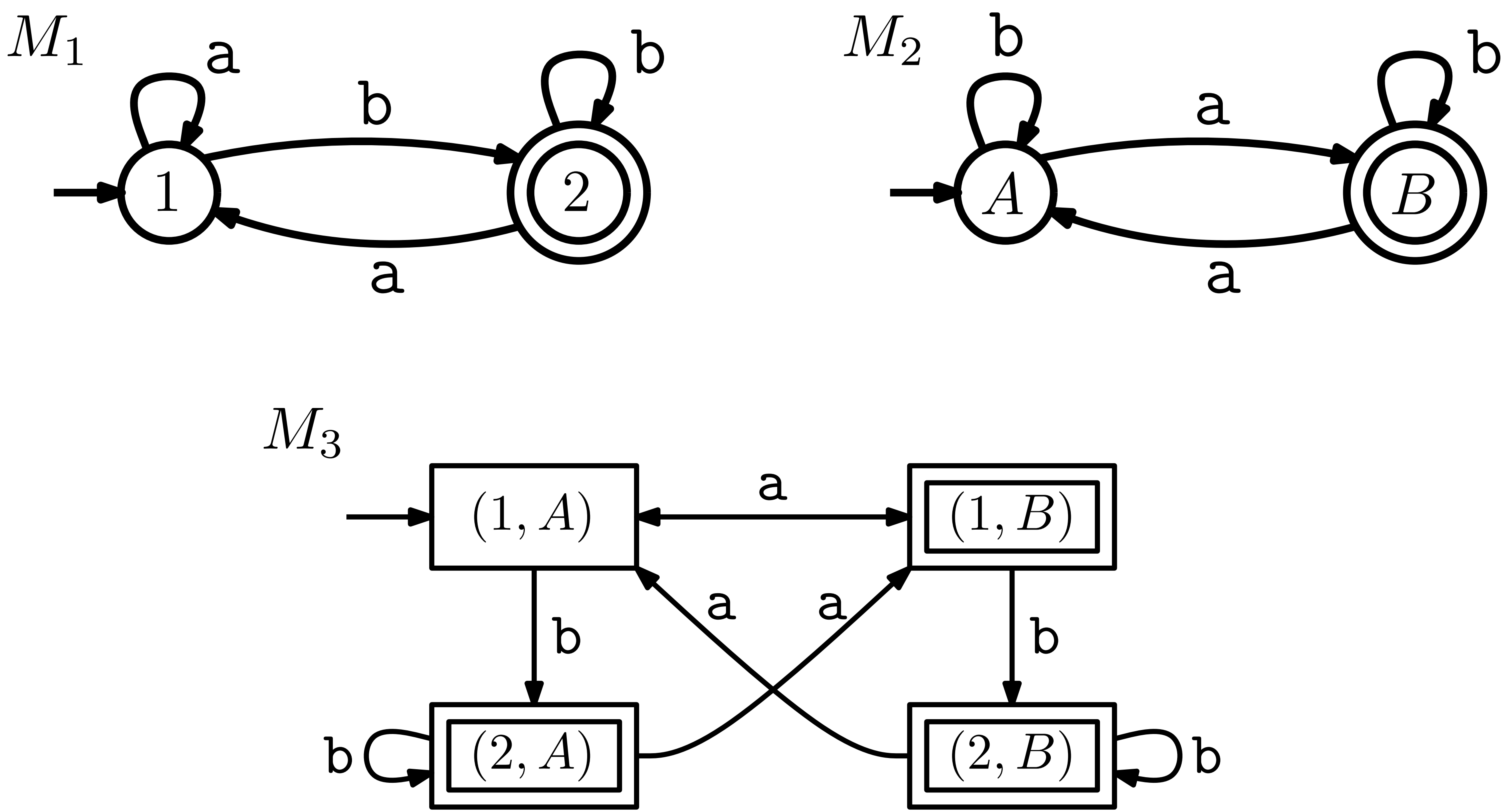
Beispiel







Beispiel



Beweis 2 für Satz 3

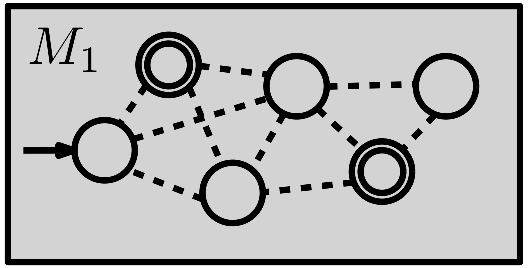
Beweis 2 für Satz 3

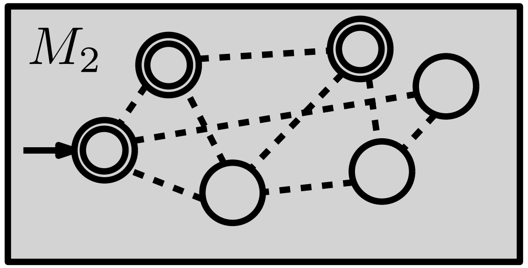
- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \cup L_2$

Beweis 2 für Satz 3

- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \cup L_2$

Schema



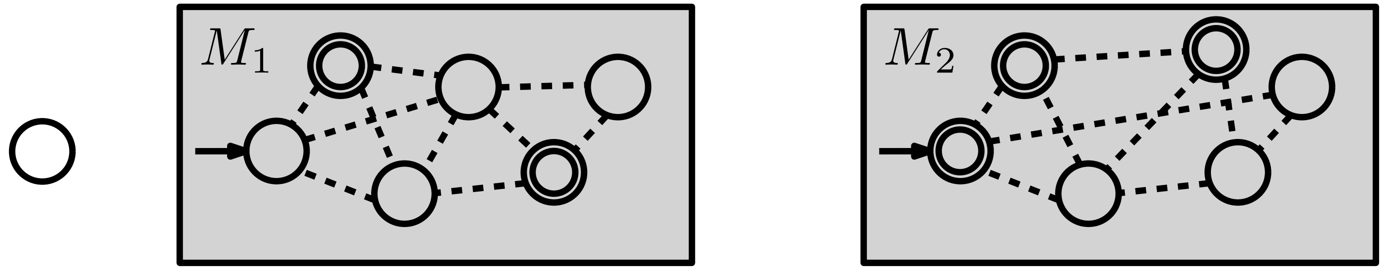


Beweis 2 für Satz 3

- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \cup L_2$

Schema

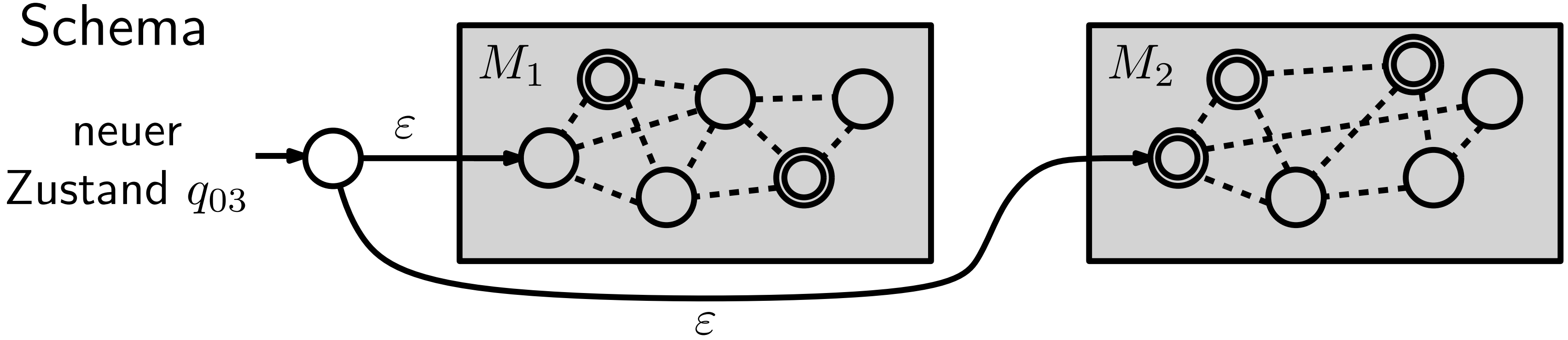
neuer Zustand q_{03}



Beweis 2 für Satz 3

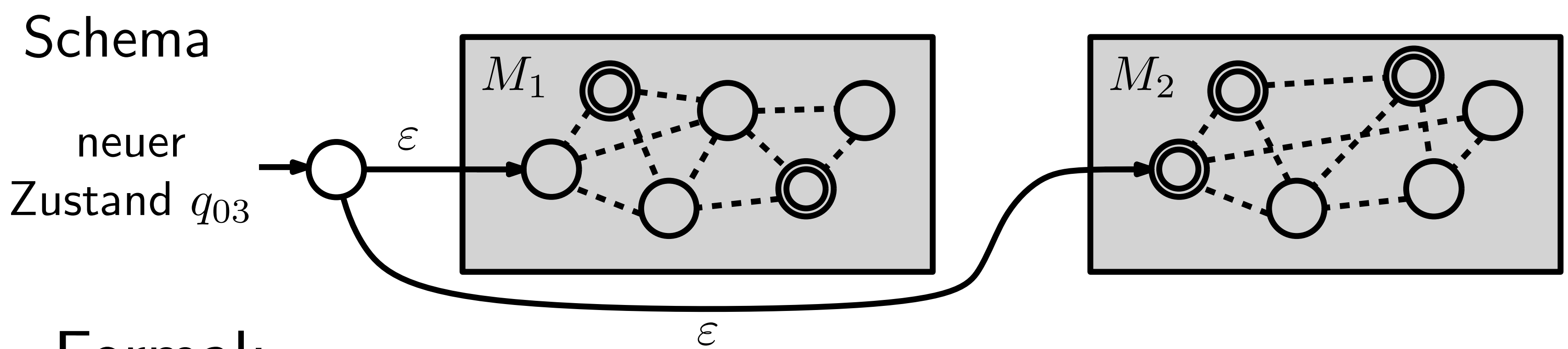
- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \cup L_2$

Schema



Beweis 2 für Satz 3

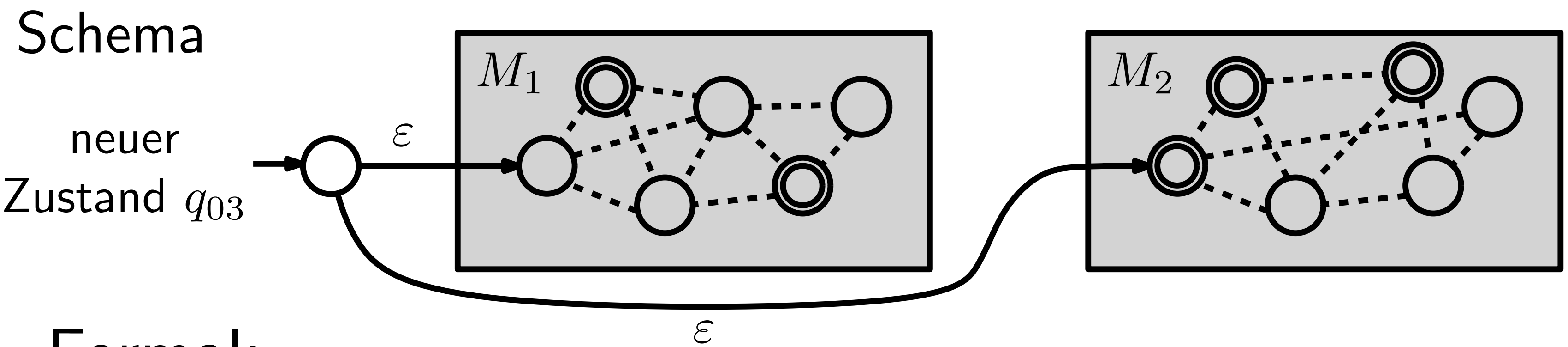
- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \cup L_2$



- Formal:
  - $Q_3 := Q_1 \cup Q_2 \cup \{q_{03}\}$,
  - $\delta_3(q,a) := \begin{cases} \{q_{01}, q_{02}\} & \text{wenn } q = q_{03} \text{ und } a = \varepsilon \\ \emptyset & \text{wenn } q = q_{03} \text{ und } a \neq \varepsilon \\ \delta_1(q,a) & \text{wenn } q \in Q_1 \\ \delta_2(q,a) & \text{wenn } q \in Q_2 \end{cases}$
  - $F_3 := F_1 \cup F_2$

Beweis 2 für Satz 3

- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \cup L_2$



- Formal:
  - $Q_3 := Q_1 \cup Q_2 \cup \{q_{03}\}$,
  - $\delta_3(q,a) := \begin{cases} \{q_{01}, q_{02}\} & \text{wenn } q = q_{03} \text{ und } a = \varepsilon \\ \emptyset & \text{wenn } q = q_{03} \text{ und } a \neq \varepsilon \\ \delta_1(q,a) & \text{wenn } q \in Q_1 \\ \delta_2(q,a) & \text{wenn } q \in Q_2 \end{cases}$
  - $F_3 := F_1 \cup F_2$

Annahme:
Q_1 und Q_2 disjunkt
$q_{03} \notin Q_1 \cup Q_2$

Weitere Abschlusseigenschaften

Weitere Abschlusseigenschaften

Satz 4

Die Regulären Sprachen sind abgeschlossen unter Konkatenation, Kleene Stern und Komplement.

Weitere Abschlusseigenschaften

Satz 4

Die Regulären Sprachen sind abgeschlossen unter Konkatenation, Kleene Stern und Komplement.

Beweisskizzen

- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \circ L_2$

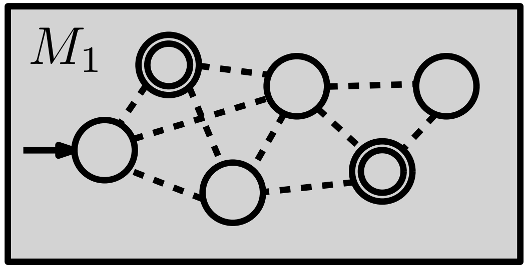
Weitere Abschlusseigenschaften

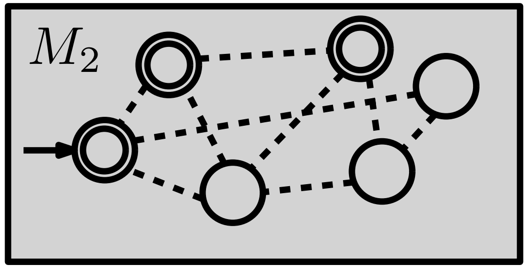
Satz 4

Die Regulären Sprachen sind abgeschlossen unter Konkatenation, Kleene Stern und Komplement.

Beweisskizzen

- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \circ L_2$





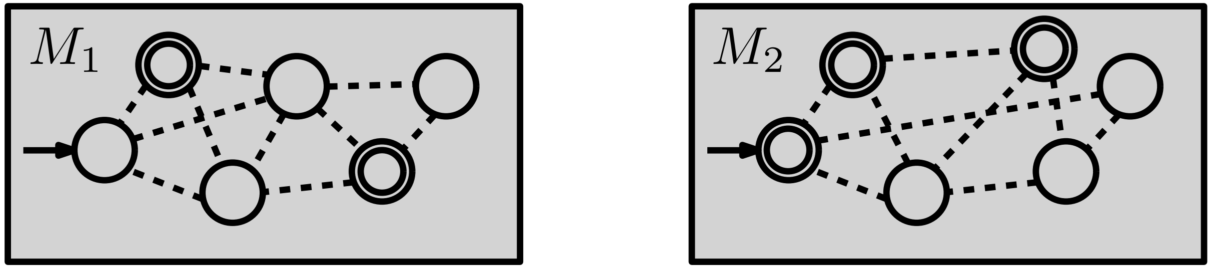
Weitere Abschlusseigenschaften

Satz 4

Die Regulären Sprachen sind abgeschlossen unter Konkatenation, Kleene Stern und Komplement.

Beweisskizzen

- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \circ L_2$



- verbinde alle akz. Zustände von M_1 über ε Kanten mit q_{02} und setzte $q_{03} = q_{01}$ und $F_3 = F_2$.

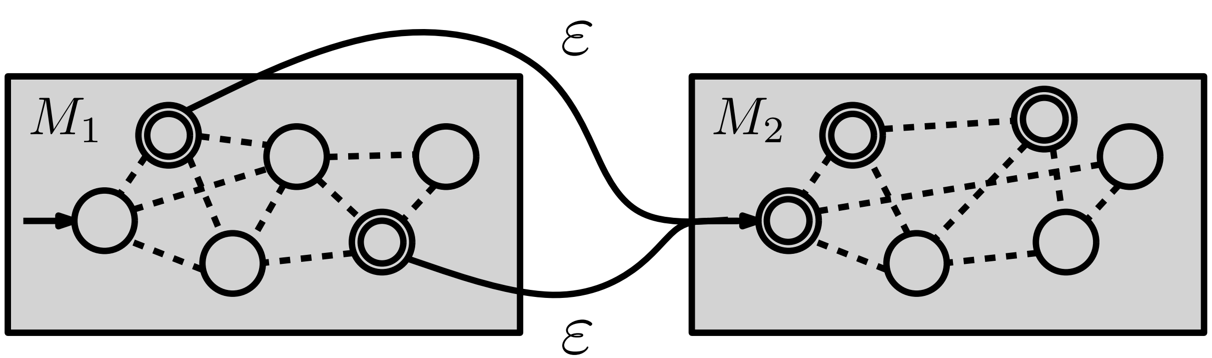
Weitere Abschlusseigenschaften

Satz 4

Die Regulären Sprachen sind abgeschlossen unter Konkatenation, Kleene Stern und Komplement.

Beweisskizzen

- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \circ L_2$



- verbinde alle akz. Zustände von M_1 über ε Kanten mit q_{02} und setzte $q_{03} = q_{01}$ und $F_3 = F_2$.

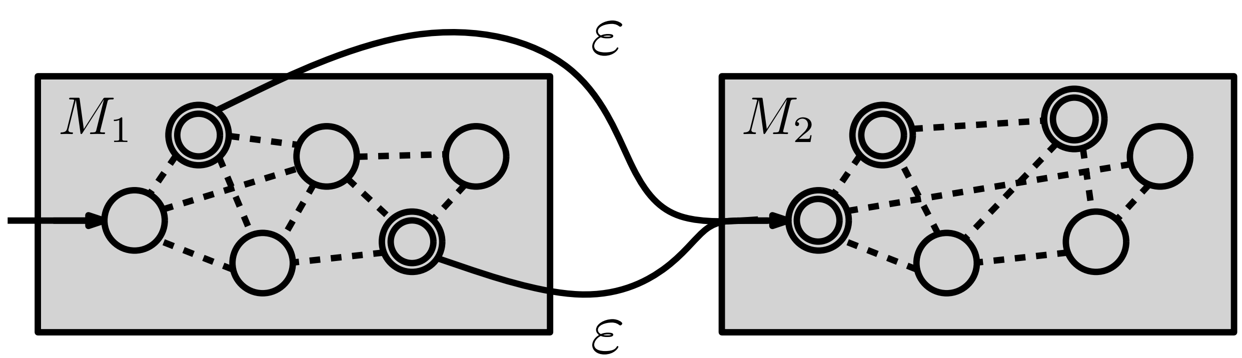
Weitere Abschlusseigenschaften

Satz 4

Die Regulären Sprachen sind abgeschlossen unter Konkatenation, Kleene Stern und Komplement.

Beweisskizzen

- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \circ L_2$



- verbinde alle akz. Zustände von M_1 über ε Kanten mit q_{02} und setzte $q_{03} = q_{01}$ und $F_3 = F_2$.

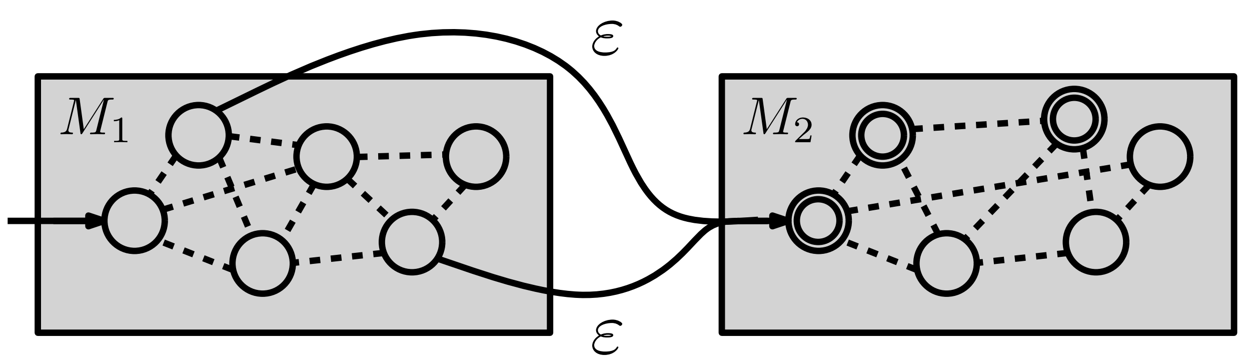
Weitere Abschlusseigenschaften

Satz 4

Die Regulären Sprachen sind abgeschlossen unter Konkatenation, Kleene Stern und Komplement.

Beweisskizzen

- M_1, M_2 wie bisher, wir konstruieren nun **NEA** $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \circ L_2$



- verbinde alle akz. Zustände von M_1 über ε Kanten mit q_{02} und setzte $q_{03} = q_{01}$ und $F_3 = F_2$.

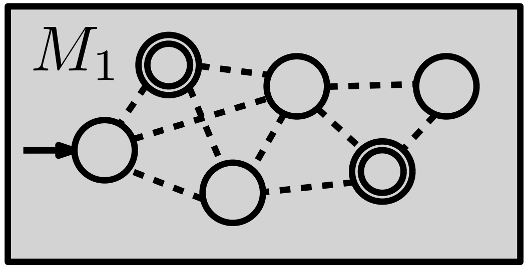
Abschluss Kleene Stern und Komplement[14]

Abschluss Kleene Stern und Komplement

- M_1 wie bisher, wir konstruieren nun **NEA** N_3 für L_1^*

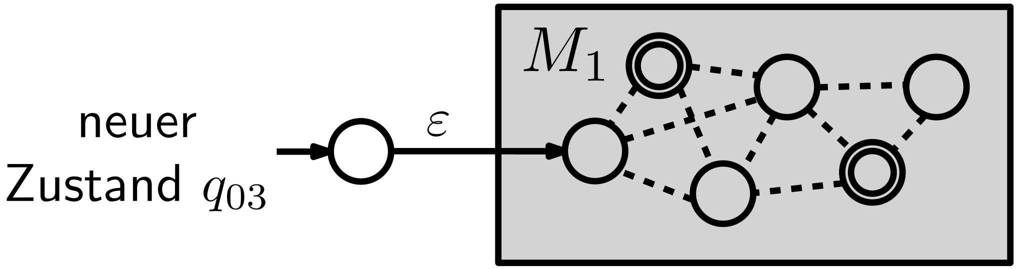
Abschluss Kleene Stern und Komplement

- M_1 wie bisher, wir konstruieren nun **NEA** N_3 für L_1^*



Abschluss Kleene Stern und Komplement

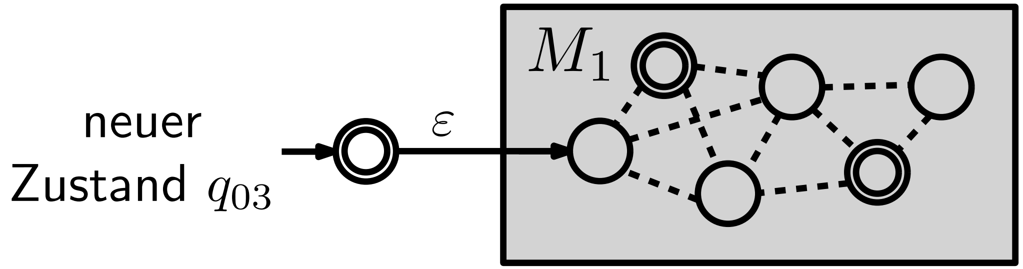
- M_1 wie bisher, wir konstruieren nun **NEA** N_3 für L_1^*



- neuer Startzustand q_{03} über ε-Kante mit q_{01} verbunden

Abschluss Kleene Stern und Komplement

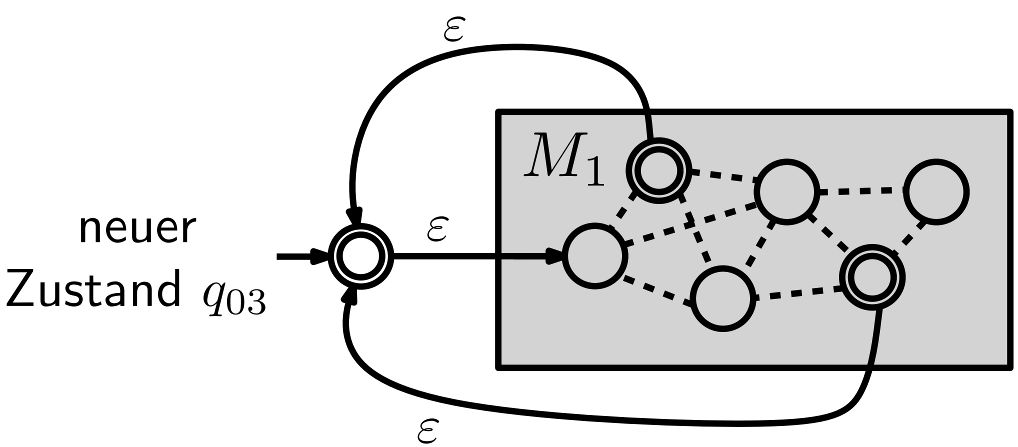
- M_1 wie bisher, wir konstruieren nun **NEA** N_3 für L_1^*



- neuer Startzustand q_{03} über ε-Kante mit q_{01} verbunden
- neuer Startzustand ist akzeptierend, da $\varepsilon \in L(M_1)$

Abschluss Kleene Stern und Komplement

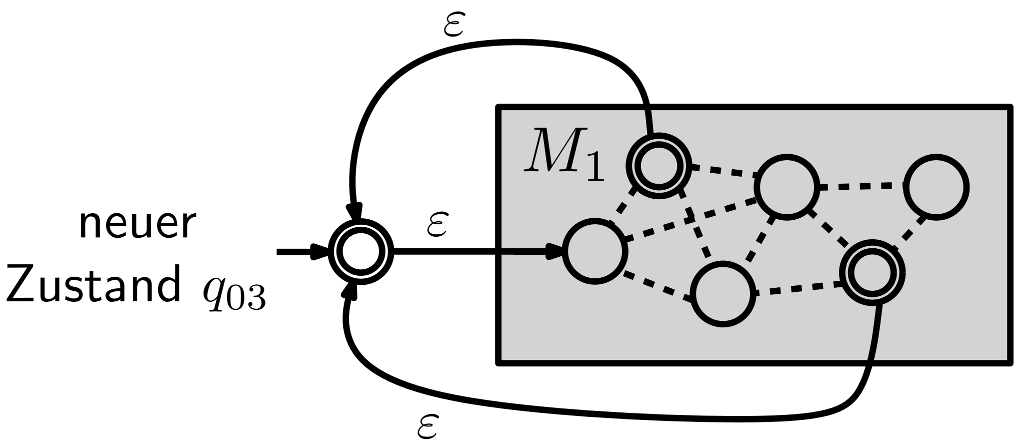
- M_1 wie bisher, wir konstruieren nun **NEA** N_3 für L_1^*



- neuer Startzustand q_{03} über ε-Kante mit q_{01} verbunden
- neuer Startzustand ist akzeptierend, da $\varepsilon \in L(M_1)$
- verbinde alle Zustände aus F_1 über ε-Kanten mit q_{03}

Abschluss Kleene Stern und Komplement

- M_1 wie bisher, wir konstruieren nun **NEA** N_3 für L_1^*

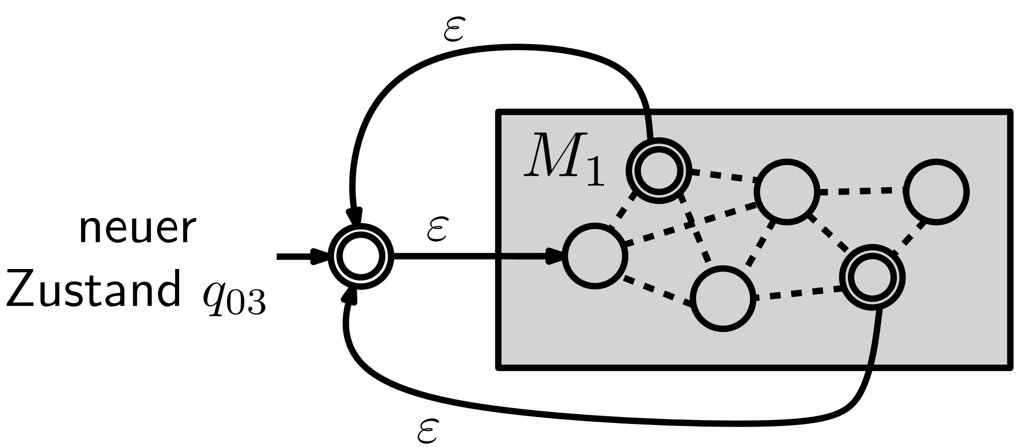


  - neuer Startzustand q_{03} über ε-Kante mit q_{01} verbunden
  - neuer Startzustand ist akzeptierend, da $\varepsilon \in L(M_1)$
  - verbinde alle Zustände aus F_1 über ε-Kanten mit q_{03}

- für **Komplement** tausche akzeptierende Zustände aus

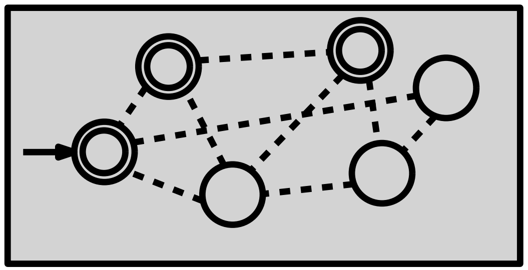
Abschluss Kleene Stern und Komplement

- M_1 wie bisher, wir konstruieren nun **NEA** N_3 für L_1^*



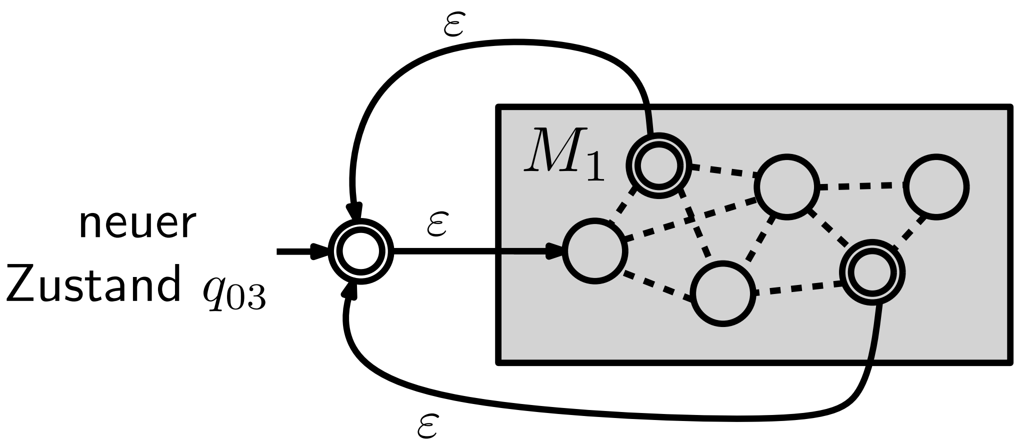
  - neuer Startzustand q_{03} über ε-Kante mit q_{01} verbunden
  - neuer Startzustand ist akzeptierend, da $\varepsilon \in L(M_1)$
  - verbinde alle Zustände aus F_1 über ε-Kanten mit q_{03}

- für **Komplement** tausche akzeptierende Zustände aus



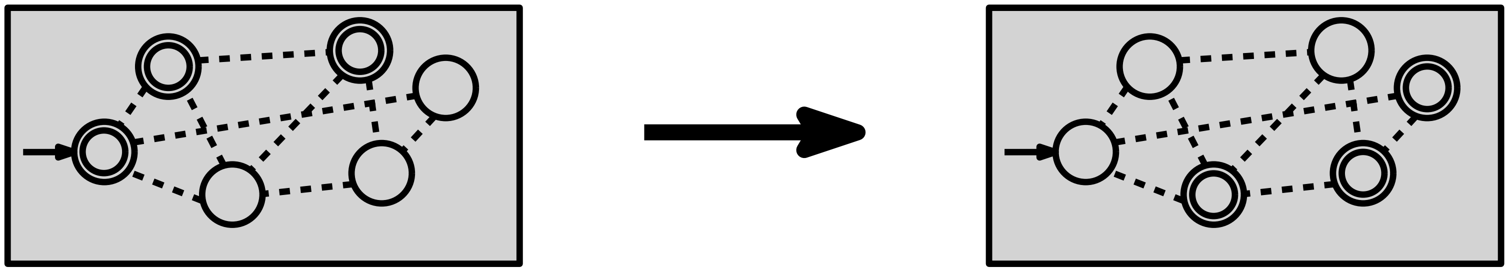
Abschluss Kleene Stern und Komplement

- M_1 wie bisher, wir konstruieren nun **NEA** N_3 für L_1^*



  - neuer Startzustand q_{03} über ε-Kante mit q_{01} verbunden
  - neuer Startzustand ist akzeptierend, da $\varepsilon \in L(M_1)$
  - verbinde alle Zustände aus F_1 über ε-Kanten mit q_{03}

- für **Komplement** tausche akzeptierende Zustände aus



Reguläre Ausdrücke

Reguläre Ausdrücke

- Formalismus zum Beschreiben von formalen Sprachen

Reguläre Ausdrücke

- Formalismus zum Beschreiben von formalen Sprachen
- ein Regulärer Ausdruck (RA) ist ein Wort über dem Alphabet $\Sigma \cup \{\emptyset, \varepsilon, (,), +, \cdot, *\}$

Reguläre Ausdrücke

- Formalismus zum Beschreiben von formalen Sprachen
- ein Regulärer Ausdruck (RA) ist ein Wort über dem Alphabet $\Sigma \cup \{\emptyset, \varepsilon, (,), +, \cdot, *\}$

Syntax

Reguläre Ausdrücke

- Formalismus zum Beschreiben von formalen Sprachen
- ein Regulärer Ausdruck (RA) ist ein Wort über dem Alphabet $\Sigma \cup \{\emptyset, \varepsilon, (,), +, \cdot, *\}$

Syntax
(1) $\forall a \in \Sigma \cup \{\emptyset, \varepsilon\}$: a ist RA
(2) Wenn A, B Reg. Ausdrücke, dann auch $(A + B)$
(3) Wenn A, B Reg. Ausdrücke, dann auch $A \cdot B$ (kurz AB)
(4) Wenn A Reg. Ausdruck, dann auch A^*
(5) Wenn A Reg. Ausdruck, dann auch (A)

Reguläre Ausdrücke

- Formalismus zum Beschreiben von formalen Sprachen
- ein Regulärer Ausdruck (RA) ist ein Wort über dem Alphabet $\Sigma \cup \{\emptyset, \varepsilon, (,), +, \cdot, *\}$

Syntax

(1) $\forall a \in \Sigma \cup \{\emptyset, \varepsilon\}$: a ist RA
(2) Wenn A, B Reg. Ausdrücke, dann auch $(A + B)$
(3) Wenn A, B Reg. Ausdrücke, dann auch $A \cdot B$ (kurz AB)
(4) Wenn A Reg. Ausdruck, dann auch A^*
(5) Wenn A Reg. Ausdruck, dann auch (A)

Bsp

$((\mathtt{aba}^* + \mathtt{abb})^*\mathtt{ab} + \varepsilon)$ ist RA

$(\mathtt{a} + \mathtt{b})\mathtt{b}(^*\mathtt{b})$ ist kein RA

Reguläre Ausdrücke

- Formalismus zum Beschreiben von formalen Sprachen
- ein Regulärer Ausdruck (RA) ist ein Wort über dem Alphabet $\Sigma \cup \{\emptyset, \varepsilon, (,), +, \cdot, *\}$

Syntax
(1) $\forall a \in \Sigma \cup \{\emptyset, \varepsilon\}$: a ist RA
(2) Wenn A, B Reg. Ausdrücke, dann auch $(A + B)$
(3) Wenn A, B Reg. Ausdrücke, dann auch $A \cdot B$ (kurz AB)
(4) Wenn A Reg. Ausdruck, dann auch A^*
(5) Wenn A Reg. Ausdruck, dann auch (A)

Bsp
$((\mathtt{aba}^* + \mathtt{abb})^*\mathtt{ab} + \varepsilon)$ ist RA
$(\mathtt{a} + \mathtt{b})\mathtt{b}(^*\mathtt{b})$ ist kein RA

Induktive Definition korrekt, da RA durch die Anwendung der (umgekehrten) Regeln immer kürzer werden

Semantik Regulärer Ausdrücke

Semantik Regulärer Ausdrücke

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

Semantik Regulärer Ausdrücke

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

(1)
- $R = \emptyset$, dann $L(R) = \emptyset$
- $R = \varepsilon$, dann $L(R) = \{\varepsilon\}$
- $R = \mathsf{a}$, und $\mathsf{a} \in \Sigma$, dann $L(R) = \{\mathsf{a}\}$

Semantik Regulärer Ausdrücke

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

(1) • $R = \emptyset$, dann $L(R) = \emptyset$
• $R = \varepsilon$, dann $L(R) = \{\varepsilon\}$
• $R = \mathsf{a}$, und $\mathsf{a} \in \Sigma$, dann $L(R) = \{\mathsf{a}\}$

(2) $R = (A + B)$ und A, B sind RA, dann $L(R) = L(A) \cup L(B)$

Semantik Regulärer Ausdrücke

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

(1)
- $R = \emptyset$, dann $L(R) = \emptyset$
- $R = \varepsilon$, dann $L(R) = \{\varepsilon\}$
- $R = \mathsf{a}$, und $\mathsf{a} \in \Sigma$, dann $L(R) = \{\mathsf{a}\}$

(2) $R = (A + B)$ und A, B sind RA, dann $L(R) = L(A) \cup L(B)$

(3) $R = (AB)$ und A, B sind RA, dann $L(R) = L(A) \circ L(B)$

Semantik Regulärer Ausdrücke

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

(1)
- $R = \emptyset$, dann $L(R) = \emptyset$
- $R = \varepsilon$, dann $L(R) = \{\varepsilon\}$
- $R = \mathsf{a}$, und $\mathsf{a} \in \Sigma$, dann $L(R) = \{\mathsf{a}\}$

(2) $R = (A + B)$ und A, B sind RA, dann $L(R) = L(A) \cup L(B)$

(3) $R = (AB)$ und A, B sind RA, dann $L(R) = L(A) \circ L(B)$

(4) $R = A^*$ und A ist RA, dann $L(R) = L(A)^*$

Semantik Regulärer Ausdrücke

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

(1)
- $R = \emptyset$, dann $L(R) = \emptyset$
- $R = \varepsilon$, dann $L(R) = \{\varepsilon\}$
- $R = \mathsf{a}$, und $\mathsf{a} \in \Sigma$, dann $L(R) = \{\mathsf{a}\}$

(2) $R = (A + B)$ und A, B sind RA, dann $L(R) = L(A) \cup L(B)$

(3) $R = (AB)$ und A, B sind RA, dann $L(R) = L(A) \circ L(B)$

(4) $R = A^*$ und A ist RA, dann $L(R) = L(A)^*$

(5) $R = (A)$ und A ist RA, dann $L(R) = L(A)$

Semantik Regulärer Ausdrücke

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

(1)
- $R = \emptyset$, dann $L(R) = \emptyset$
- $R = \varepsilon$, dann $L(R) = \{\varepsilon\}$
- $R = \mathsf{a}$, und $\mathsf{a} \in \Sigma$, dann $L(R) = \{\mathsf{a}\}$

(2) $R = (A + B)$ und A, B sind RA, dann $L(R) = L(A) \cup L(B)$

(3) $R = (AB)$ und A, B sind RA, dann $L(R) = L(A) \circ L(B)$

(4) $R = A^*$ und A ist RA, dann $L(R) = L(A)^*$

(5) $R = (A)$ und A ist RA, dann $L(R) = L(A)$

Reihenfolge der Operatoren: $*$ vor $\cdot$ vor $+$

Semantik Regulärer Ausdrücke

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

(1)
- $R = \emptyset$, dann $L(R) = \emptyset$
- $R = \varepsilon$, dann $L(R) = \{\varepsilon\}$
- $R = \mathtt{a}$, und $\mathtt{a} \in \Sigma$, dann $L(R) = \{\mathtt{a}\}$

(2) $R = (A + B)$ und A, B sind RA, dann $L(R) = L(A) \cup L(B)$

(3) $R = (AB)$ und A, B sind RA, dann $L(R) = L(A) \circ L(B)$

(4) $R = A^*$ und A ist RA, dann $L(R) = L(A)^*$

(5) $R = (A)$ und A ist RA, dann $L(R) = L(A)$

Reihenfolge der Operatoren: $*$ vor $\cdot$ vor $+$

Bsp. $R = (\mathtt{a} + \mathtt{b})^*\mathtt{aba}(\mathtt{a} + \mathtt{b})^*$

Semantik Regulärer Ausdrücke

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

(1)
- $R = \emptyset$, dann $L(R) = \emptyset$
- $R = \varepsilon$, dann $L(R) = \{\varepsilon\}$
- $R = \mathsf{a}$, und $\mathsf{a} \in \Sigma$, dann $L(R) = \{\mathsf{a}\}$

(2) $R = (A + B)$ und A, B sind RA, dann $L(R) = L(A) \cup L(B)$

(3) $R = (AB)$ und A, B sind RA, dann $L(R) = L(A) \circ L(B)$

(4) $R = A^*$ und A ist RA, dann $L(R) = L(A)^*$

(5) $R = (A)$ und A ist RA, dann $L(R) = L(A)$

Reihenfolge der Operatoren: $*$ vor $\cdot$ vor $+$

Bsp. $R = (\mathsf{a} + \mathsf{b})^*\mathsf{aba}(\mathsf{a} + \mathsf{b})^*$

$$L(R) = \{w \in \{\mathsf{a}, \mathsf{b}\}^* \mid w \text{ enthält Teilwort aba}\}$$

Satz von Kleene

Satz von Kleene

Satz 5

$$\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} = \text{REG}$$

Satz von Kleene

Satz 5

$$\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} = \text{REG}$$

Beweis (Organisation)

Satz von Kleene

Satz 5

$$\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} = \text{REG}$$

Beweis (Organisation)

1. Teil: Wir konstruieren zu einem RA R einen NEA N mit $L(R) = L(N)$

Satz von Kleene

Satz 5

$$\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} = \text{REG}$$

Beweis (Organisation)

1. Teil: Wir konstruieren zu einem RA R einen NEA N mit $L(R) = L(N)$

das zeigt $\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} \subseteq \text{REG}$

Satz von Kleene

Satz 5

$$\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} = \text{REG}$$

Beweis (Organisation)

1. Teil: Wir konstruieren zu einem RA R einen NEA N mit $L(R) = L(N)$

das zeigt $\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} \subseteq \text{REG}$

2. Teil: Wir konstruieren zu einem DEA M einen RA R mit $L(M) = L(R)$

Satz von Kleene

Satz 5

$$\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} = \text{REG}$$

Beweis (Organisation)

1. Teil: Wir konstruieren zu einem RA R einen NEA N mit $L(R) = L(N)$

das zeigt $\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} \subseteq \text{REG}$

2. Teil: Wir konstruieren zu einem DEA M einen RA R mit $L(M) = L(R)$

das zeigt $\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} \supseteq \text{REG}$

Für RA R ist $L(R)$ regulär

Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

Ind.anfang: Eine Ableitungsregel

Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

Ind.anfang: Eine Ableitungsregel

3 Möglichkeiten

$R = \varepsilon$ $\qquad$ $R = \emptyset$ $\qquad$ $R = x$ für $x \in \Sigma$

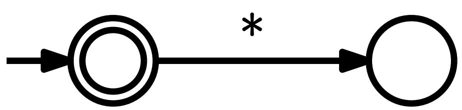
Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

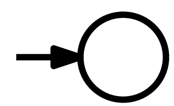
Ind.anfang: Eine Ableitungsregel

3 Möglichkeiten

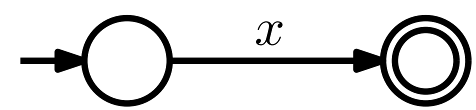
$R = \varepsilon$



$R = \emptyset$



$R = x$ für $x \in \Sigma$



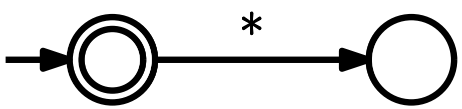
Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

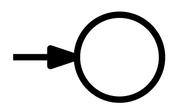
Ind.anfang: Eine Ableitungsregel

3 Möglichkeiten

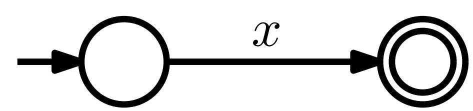
$R = \varepsilon$



$R = \emptyset$



$R = x$ für $x \in \Sigma$

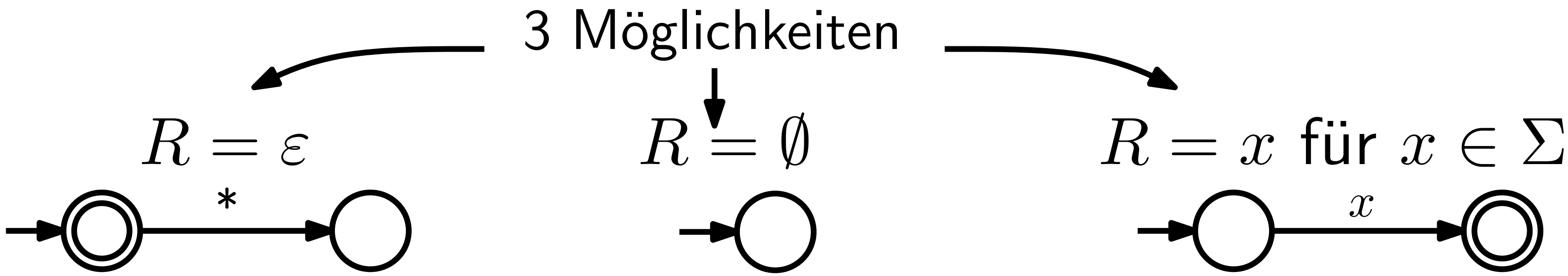


$\rightarrow L(R)$ regulär

Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

Ind.anfang: Eine Ableitungsregel



$\rightarrow L(R)$ regulär

Ind.schritt: k Ableitungsregeln

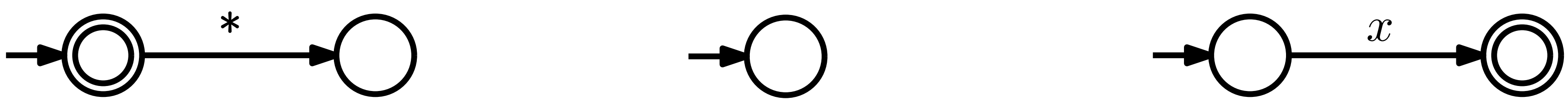
Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

Ind.anfang: Eine Ableitungsregel

3 Möglichkeiten

$R = \varepsilon$ $\qquad$ $R = \emptyset$ $\qquad$ $R = x$ für $x \in \Sigma$



$\rightarrow L(R)$ regulär

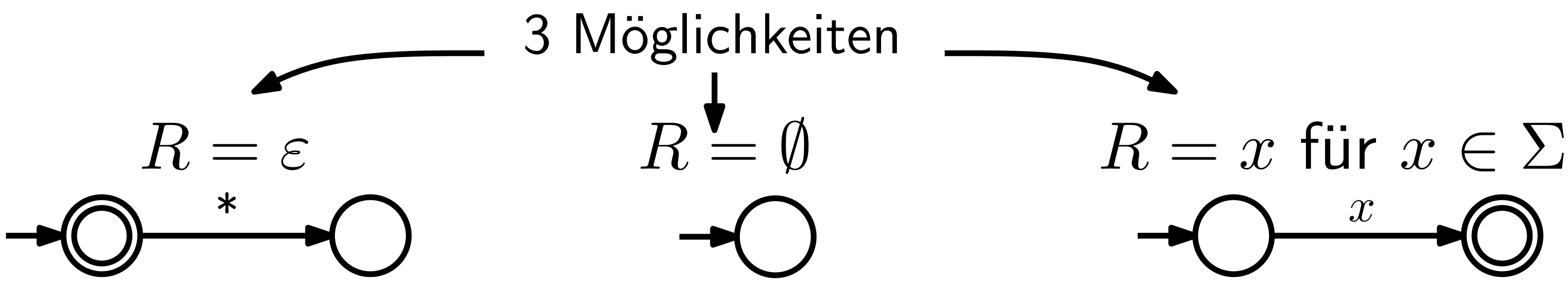
Ind.schritt: k Ableitungsregeln

- Angenommen $R = (A + B)$, und A und B sind RAs.

Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

Ind.anfang: Eine Ableitungsregel



$\rightarrow L(R)$ regulär

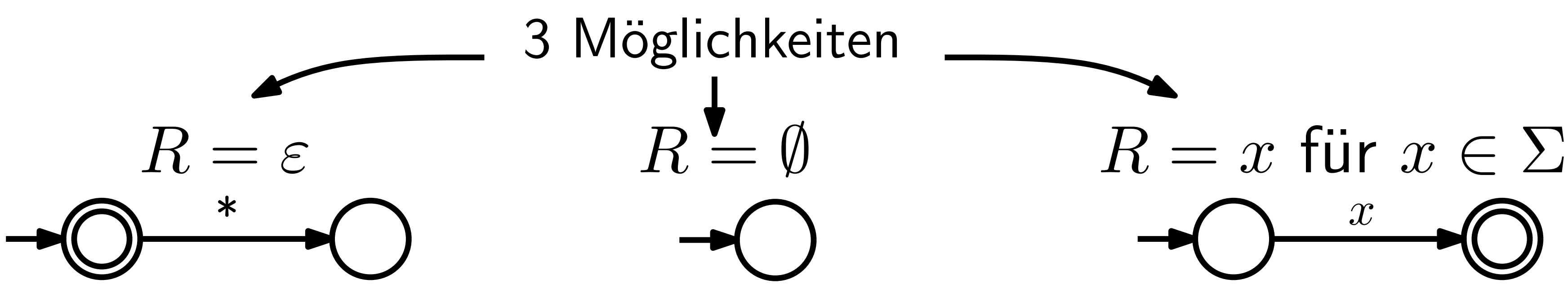
Ind.schritt: k Ableitungsregeln

- Angenommen $R = (A + B)$, und A und B sind RAs.
- Nach Induktionsannahme sind $L(A)$ und $L(B)$ regulär.

Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

Ind.anfang: Eine Ableitungsregel



$\rightarrow L(R)$ regulär

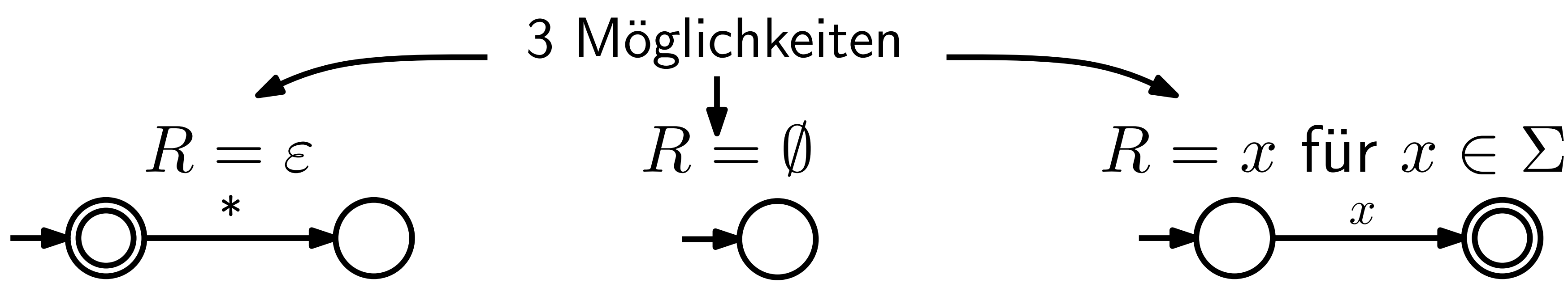
Ind.schritt: k Ableitungsregeln

- Angenommen $R = (A + B)$, und A und B sind RAs.
- Nach Induktionsannahme sind $L(A)$ und $L(B)$ regulär.
- $L(R) = L(A) \cup L(B)$ ist nach Abschlusseigenschaft (Satz 4) auch regulär.

Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

Ind.anfang: Eine Ableitungsregel



$\rightarrow$ $L(R)$ regulär

Ind.schritt: k Ableitungsregeln

- Angenommen $R = (A + B)$, und A und B sind RAs.
- Nach Induktionsannahme sind $L(A)$ und $L(B)$ regulär.
- $L(R) = L(A) \cup L(B)$ ist nach Abschlusseigenschaft (Satz 4) auch regulär.
- für Konkatenation, Stern, Klammern, analog

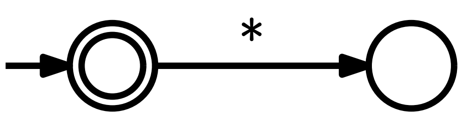
Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

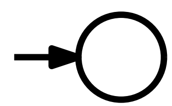
Ind.anfang: Eine Ableitungsregel

3 Möglichkeiten

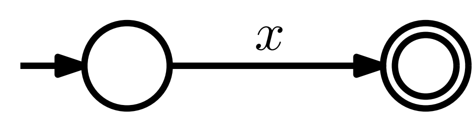
$R = \varepsilon$



$R = \emptyset$



$R = x$ für $x \in \Sigma$



$\rightarrow L(R)$ regulär

Ind.schritt: k Ableitungsregeln

- Angenommen $R = (A + B)$, und A und B sind RAs.
- Nach Induktionsannahme sind $L(A)$ und $L(B)$ regulär.
- $L(R) = L(A) \cup L(B)$ ist nach Abschlusseigenschaft (Satz 4) auch regulär.
- für Konkatenation, Stern, Klammern, analog $\square$

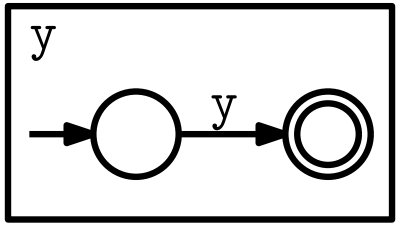
Beispiel

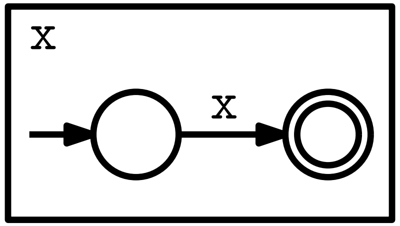
Beispiel

$R = (\mathtt{x} + \mathtt{y})^* \mathtt{x}$

Beispiel

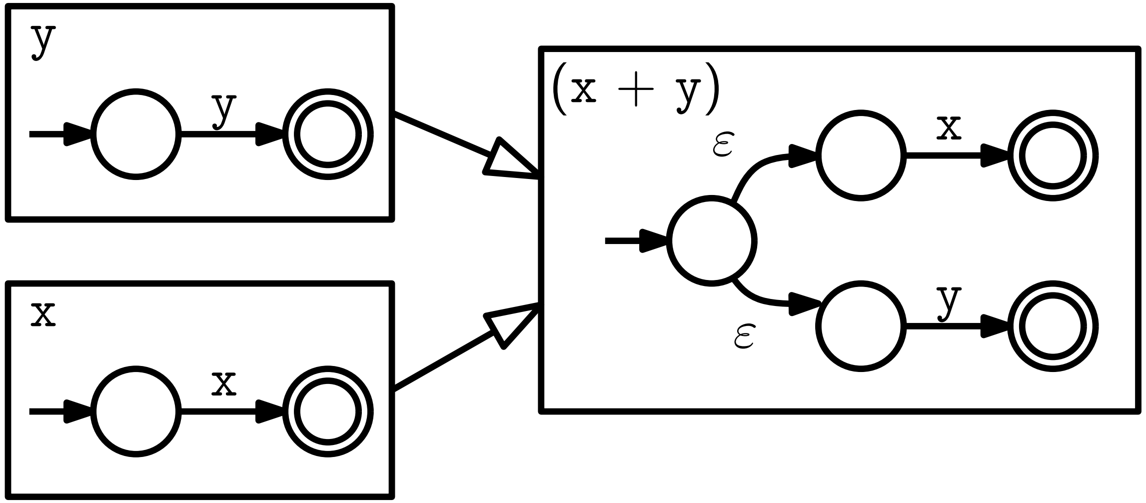
$$R = (\mathtt{x} + \mathtt{y})^* \mathtt{x}$$





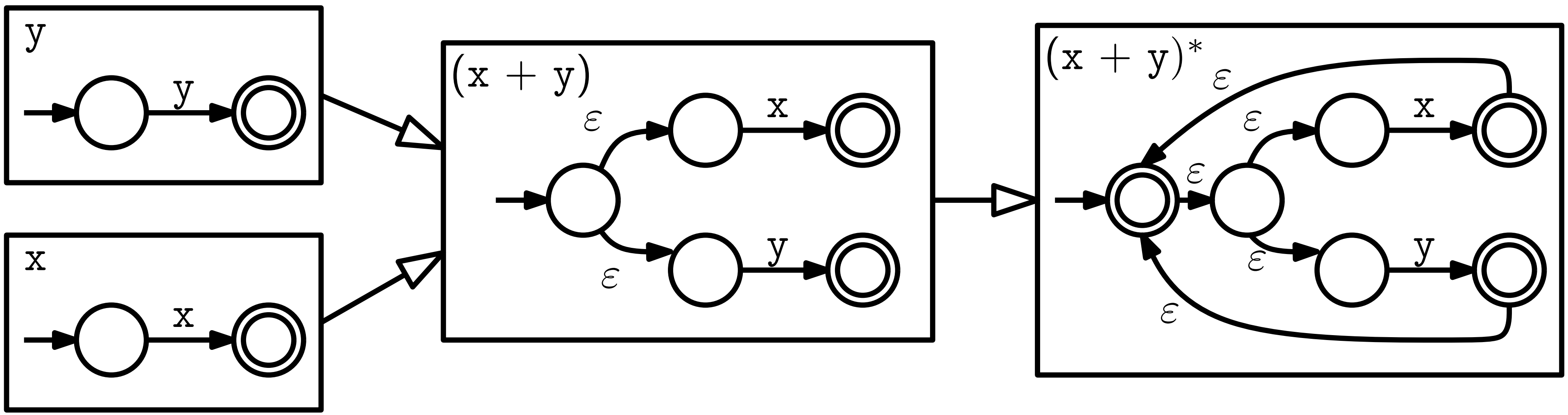
Beispiel

$R = (\mathrm{x} + \mathrm{y})^* \mathrm{x}$



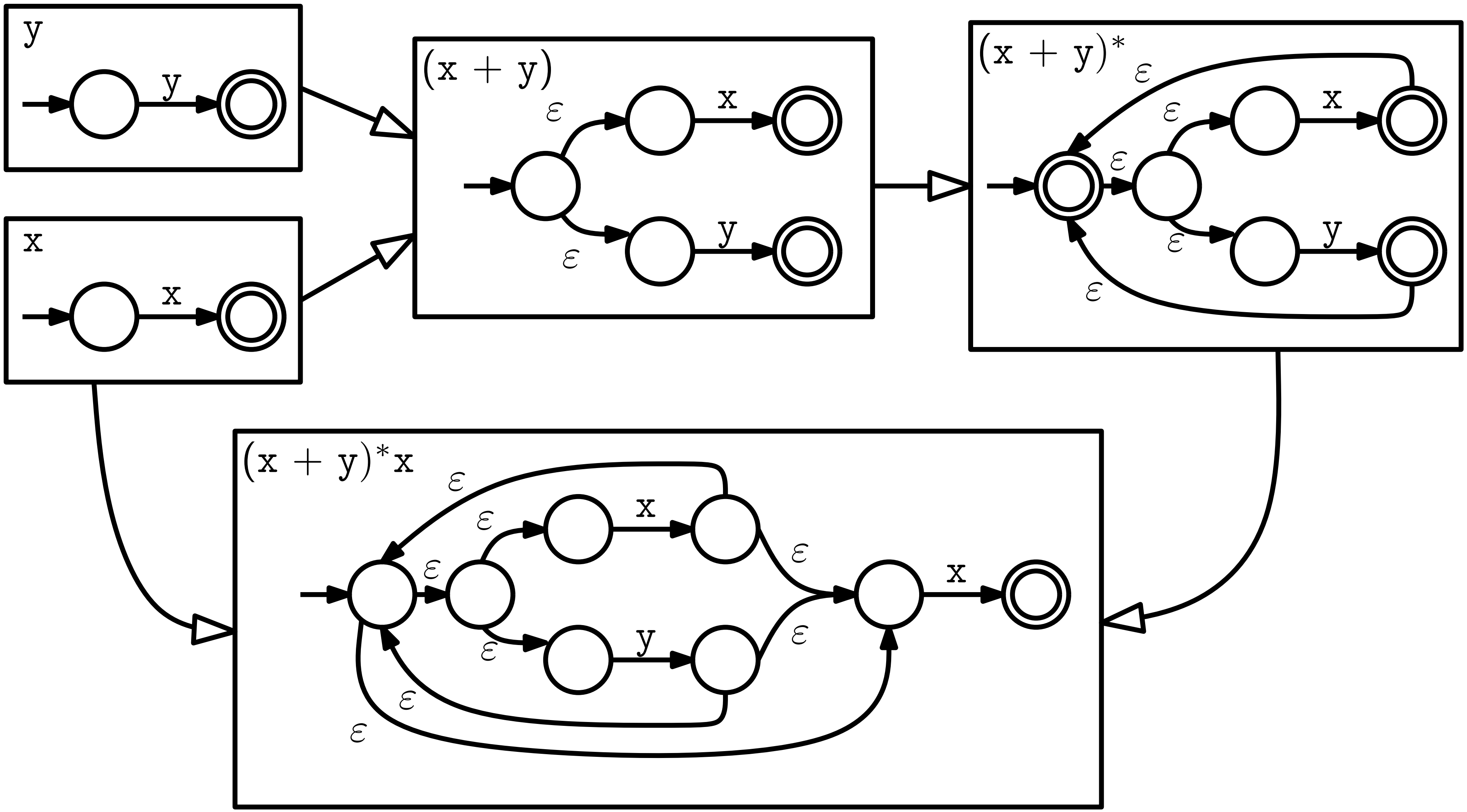
Beispiel

$R = (\mathrm{x} + \mathrm{y})^*\mathrm{x}$



Beispiel

$R = (\mathrm{x} + \mathrm{y})^*\mathrm{x}$



Vom DEA zum RA

Vom DEA zum RA

Ansatz

- Wenn $L \in$ REG, dann existiert ein DEA M der L erkennt.

Vom DEA zum RA

Ansatz

- Wenn $L \in$ REG, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$

Vom DEA zum RA

Ansatz

- Wenn $L \in$ REG, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

Vom DEA zum RA

Ansatz

- Wenn $L \in \text{REG}$, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

R_{ij}^{k} := RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

Vom DEA zum RA

Ansatz

- Wenn $L \in \text{REG}$, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

$R_{ij}^k :=$ RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

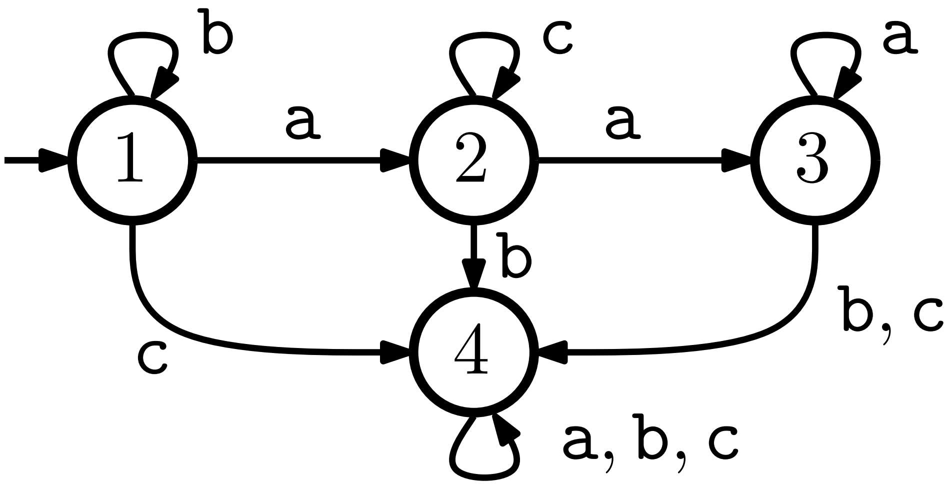
Anfangs- und Endzustand i, j dürfen $> k$ sein!

Vom DEA zum RA

Ansatz
- Wenn $L \in$ REG, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

R_{ij}^k := RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

Anfangs- und Endzustand i, j dürfen $> k$ sein!



Bsp

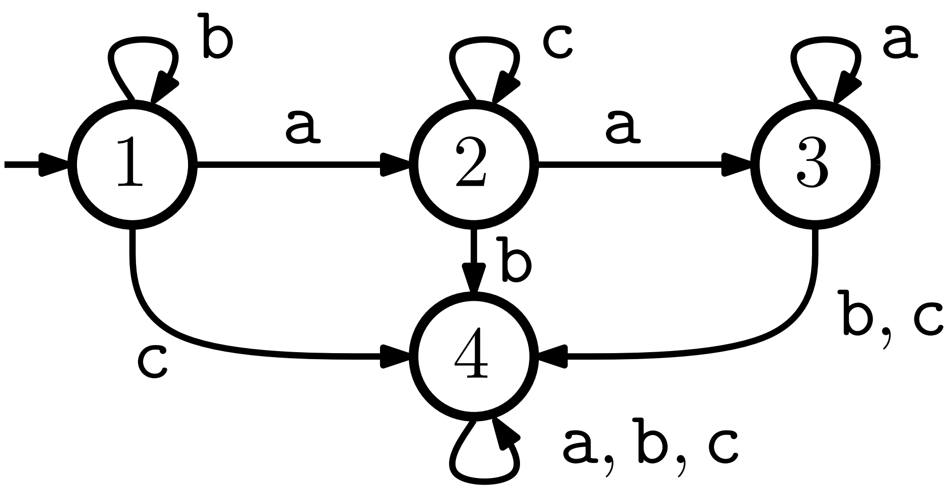
Vom DEA zum RA

Ansatz

- Wenn $L \in$ REG, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

R_{ij}^k := RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

Anfangs- und Endzustand i, j dürfen $> k$ sein!



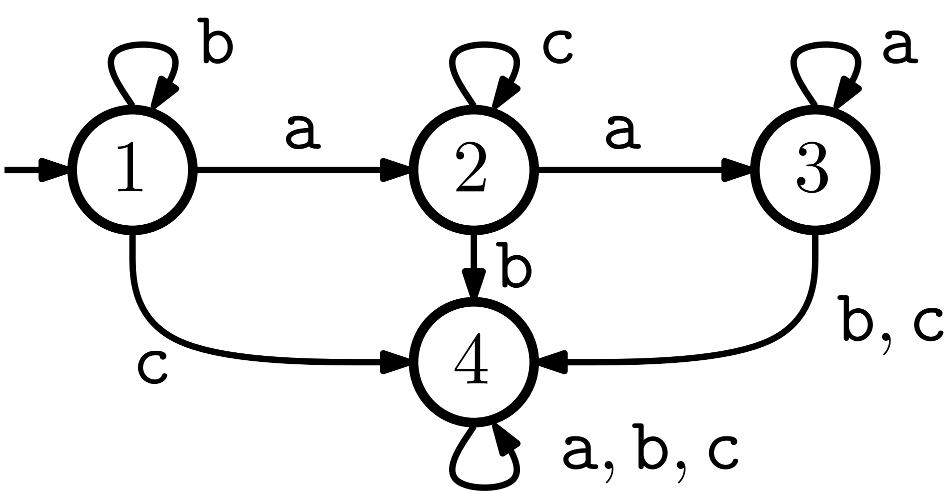
Bsp $R_{22}^0 =$

Vom DEA zum RA

Ansatz
- Wenn $L \in$ REG, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

$R_{ij}^k :=$ RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

Anfangs- und Endzustand i, j dürfen $> k$ sein!



Bsp $R_{22}^0 = \mathtt{c} + \varepsilon$

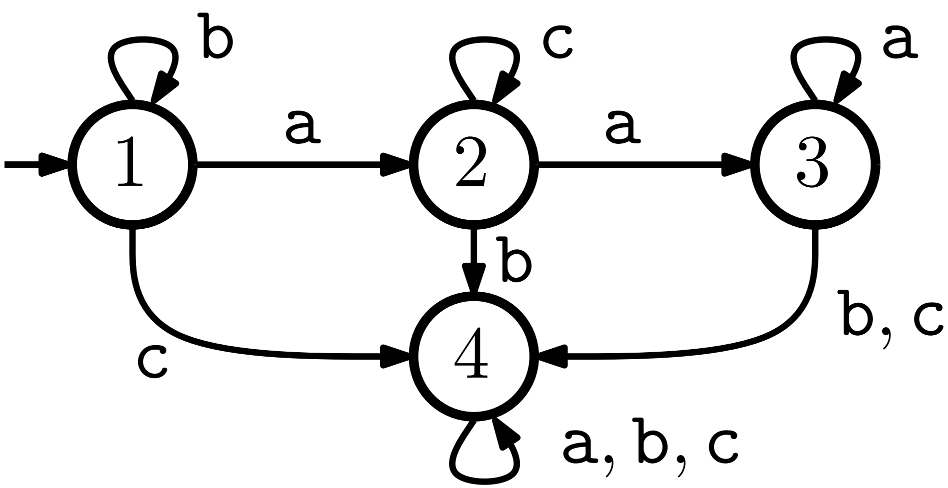
Vom DEA zum RA

Ansatz

- Wenn $L \in$ REG, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

R^k_{ij} := RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

Anfangs- und Endzustand i, j dürfen $> k$ sein!



Bsp $R^0_{22} = \mathtt{c} + \varepsilon$

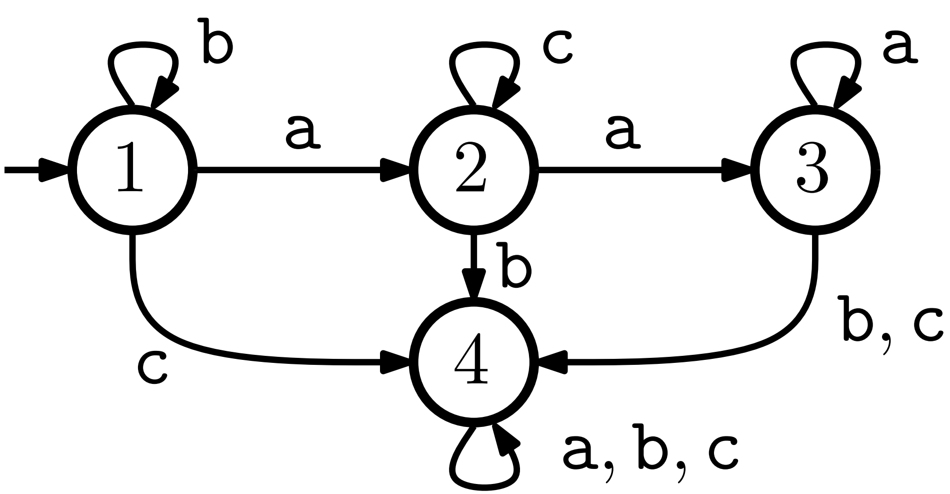
$R^2_{24} =$

Vom DEA zum RA

Ansatz
- Wenn $L \in$ REG, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

$R_{ij}^k :=$ RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

Anfangs- und Endzustand i, j dürfen $> k$ sein!



Bsp $R_{22}^0 = \mathtt{c} + \varepsilon$

$R_{24}^2 = \mathtt{c}^*\mathtt{b}$

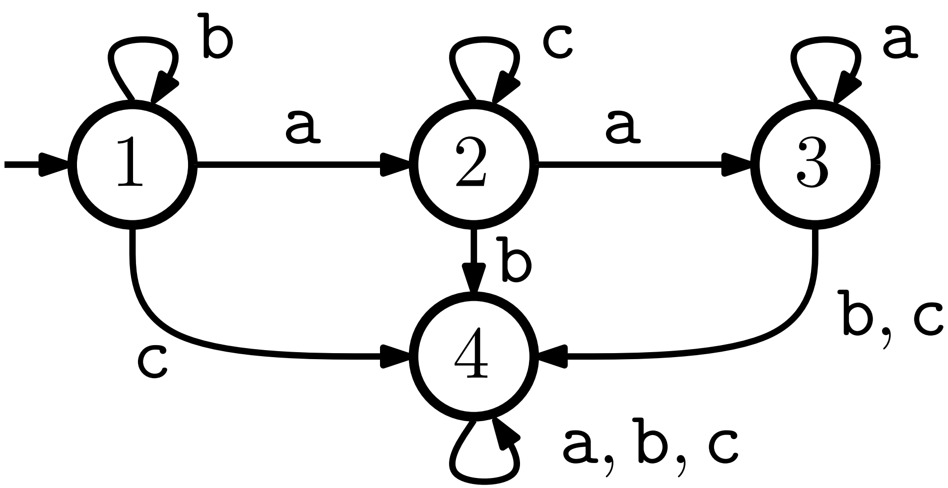
Vom DEA zum RA

Ansatz

- Wenn $L \in \text{REG}$, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

R_{ij}^k := RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

Anfangs- und Endzustand i, j dürfen $> k$ sein!



Bsp $R_{22}^0 = \mathtt{c} + \varepsilon$

$R_{24}^2 = \mathtt{c}^*\mathtt{b}$

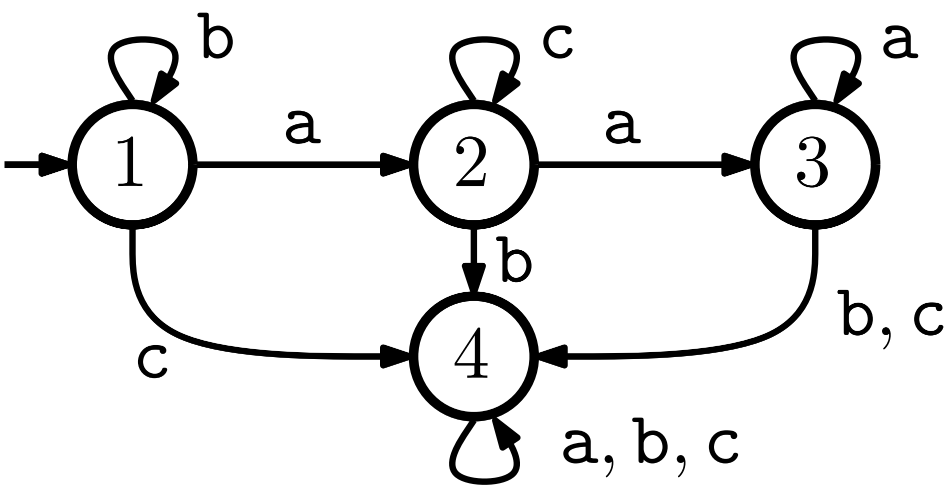
$R_{24}^3 =$

Vom DEA zum RA

Ansatz
- Wenn $L \in$ REG, dann existiert ein DEA M der L erkennt.
- $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \ldots, n\}$
- **Ziel:** Konstruiere RA R mit $L(R) = L$

$R_{ij}^k :=$ RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

Anfangs- und Endzustand i, j dürfen $> k$ sein!



Bsp $R_{22}^0 = \mathtt{c} + \varepsilon$

$R_{24}^2 = \mathtt{c}^*\mathtt{b}$

$R_{24}^3 = (\mathtt{c}^*\mathtt{b} + \mathtt{c}^*\mathtt{aa}^*(\mathtt{b} + \mathtt{c}))$

Von den Termen zum RA

Von den Termen zum RA

Annahme: Ich kenne alle Terme R^k_{ij}

Von den Termen zum RA

Annahme: Ich kenne alle Terme R_{ij}^k

$\rightarrow w \in L(M)$ genau dann, wenn $\delta^*(1, w) = j$ und $j \in F$

Von den Termen zum RA

Annahme: Ich kenne alle Terme R_{ij}^k

$\rightarrow w \in L(M)$ genau dann, wenn $\underbrace{\delta^*(1, w) = j}_{w \in R_{1j}^n}$ und $j \in F$

Von den Termen zum RA

Annahme: Ich kenne alle Terme R_{ij}^k

$\rightarrow w \in L(M)$ genau dann, wenn $\underbrace{\delta^*(1, w) = j}_{w \in R_{1j}^n}$ und $j \in F$

oberer Index n heißt keine Beschränkung

Von den Termen zum RA

Annahme: Ich kenne alle Terme R_{ij}^k

$\rightarrow w \in L(M)$ genau dann, wenn $\underbrace{\delta^*(1, w) = j}_{w \in R_{1j}^n}$ und $j \in F$

oberer Index n heißt keine Beschränkung

$\rightarrow$ verbinde für alle akzeptierende Zustände die R_{ij}^k Terme

Von den Termen zum RA

Annahme: Ich kenne alle Terme R_{ij}^k

$\rightarrow w \in L(M)$ genau dann, wenn $\underbrace{\delta^*(1,w) = j}_{w \in R_{1j}^n}$ und $j \in F$

oberer Index n heißt keine Beschränkung

$\rightarrow$ verbinde für alle akzeptierende Zustände die R_{ij}^k Terme

$$R := R_{1f_1}^n + R_{1f_2}^n + \cdots R_{1f_k}^n$$

wobei $F = \{f_1, f_2, \ldots, f_k\}$

Konstruktion der R^k_{ij} Terme

Konstruktion der R_{ij}^k Terme

Rekursive Konstruktion

Konstruktion der R_{ij}^k Terme

Rekursive Konstruktion

Basisfall (R_{ij}^0): nur direkte Übergänge

Konstruktion der R_{ij}^k Terme

Rekursive Konstruktion

Basisfall (R_{ij}^0): nur direkte Übergänge

$$R_{ij}^0 := \begin{cases} \emptyset + a_1 + a_2 + \dots & \text{falls } i \neq j \text{ und } a_i \in \{a \mid \delta(i,a) = j\} \\ \varepsilon + a_1 + a_2 + \dots & \text{falls } i = j \text{ und } a_i \in \{a \mid \delta(i,a) = i\} \end{cases}$$

Konstruktion der R_{ij}^k Terme

Rekursive Konstruktion

Basisfall (R_{ij}^0): nur direkte Übergänge

$$R_{ij}^0 := \begin{cases} \emptyset + a_1 + a_2 + \ldots & \text{falls } i \neq j \text{ und } a_i \in \{a \mid \delta(i,a) = j\} \\ \varepsilon + a_1 + a_2 + \ldots & \text{falls } i = j \text{ und } a_i \in \{a \mid \delta(i,a) = i\} \end{cases}$$

Rekursion (R_{ij}^k):

Konstruktion der R_{ij}^k Terme

Rekursive Konstruktion

Basisfall (R_{ij}^0): nur direkte Übergänge

$$R_{ij}^0 := \begin{cases} \emptyset + a_1 + a_2 + \dots & \text{falls } i \neq j \text{ und } a_i \in \{a \mid \delta(i,a) = j\} \\ \varepsilon + a_1 + a_2 + \dots & \text{falls } i = j \text{ und } a_i \in \{a \mid \delta(i,a) = i\} \end{cases}$$

Rekursion (R_{ij}^k): ich kenne bereits alle R_{ij}^{k-1}

Konstruktion der R_{ij}^k Terme

Rekursive Konstruktion

Basisfall (R_{ij}^0): nur direkte Übergänge

$$R_{ij}^0 := \begin{cases} \emptyset + a_1 + a_2 + \ldots & \text{falls } i \neq j \text{ und } a_i \in \{a \mid \delta(i,a) = j\} \\ \varepsilon + a_1 + a_2 + \ldots & \text{falls } i = j \text{ und } a_i \in \{a \mid \delta(i,a) = i\} \end{cases}$$

Rekursion (R_{ij}^k): ich kenne bereits alle R_{ij}^{k-1}

Bsp. von einem "Lauf" aus R_{ij}^6

$$i \xrightarrow{\mathtt{x}} 2 \xrightarrow{\mathtt{x}} 3 \xrightarrow{\mathtt{y}} 2 \xrightarrow{\mathtt{z}} 6 \xrightarrow{\mathtt{z}} 5 \xrightarrow{\mathtt{y}} 1 \xrightarrow{\mathtt{y}} 6 \xrightarrow{\mathtt{y}} 4 \xrightarrow{\mathtt{y}} 6 \xrightarrow{\mathtt{z}} 5 \xrightarrow{\mathtt{z}} j$$

Konstruktion der R_{ij}^k Terme

Rekursive Konstruktion

Basisfall (R_{ij}^0): nur direkte Übergänge

$$R_{ij}^0 := \begin{cases} \emptyset + a_1 + a_2 + \ldots & \text{falls } i \neq j \text{ und } a_i \in \{a \mid \delta(i,a) = j\} \\ \varepsilon + a_1 + a_2 + \ldots & \text{falls } i = j \text{ und } a_i \in \{a \mid \delta(i,a) = i\} \end{cases}$$

Rekursion (R_{ij}^k): ich kenne bereits alle R_{ij}^{k-1}

Bsp. von einem "Lauf" aus R_{ij}^6

$$\underbrace{i \xrightarrow{x} 2 \xrightarrow{x} 3 \xrightarrow{y} 2 \xrightarrow{z} 6}_{\in R_{i6}^5} \xrightarrow{z} 5 \xrightarrow{y} 1 \xrightarrow{y} 6 \xrightarrow{y} 4 \xrightarrow{y} 6 \xrightarrow{z} 5 \xrightarrow{z} j$$

Konstruktion der R^k_{ij} Terme

Rekursive Konstruktion

Basisfall (R^0_{ij}): nur direkte Übergänge

$$R^0_{ij} := \begin{cases} \emptyset + a_1 + a_2 + \dots & \text{falls } i \neq j \text{ und } a_i \in \{a \mid \delta(i,a) = j\} \\ \varepsilon + a_1 + a_2 + \dots & \text{falls } i = j \text{ und } a_i \in \{a \mid \delta(i,a) = i\} \end{cases}$$

Rekursion (R^k_{ij}): ich kenne bereits alle R^{k-1}_{ij}

Bsp. von einem "Lauf" aus R^6_{ij}

$$\underbrace{i \xrightarrow{x} 2 \xrightarrow{x} 3 \xrightarrow{y} 2 \xrightarrow{z} 6}_{\in R^5_{i6}} \xrightarrow{z} 5 \xrightarrow{y} 1 \xrightarrow{y} 6 \xrightarrow{y} 4 \xrightarrow{y} \underbrace{6 \xrightarrow{z} 5 \xrightarrow{z} j}_{\in R^5_{6j}}$$

Konstruktion der R_{ij}^k Terme

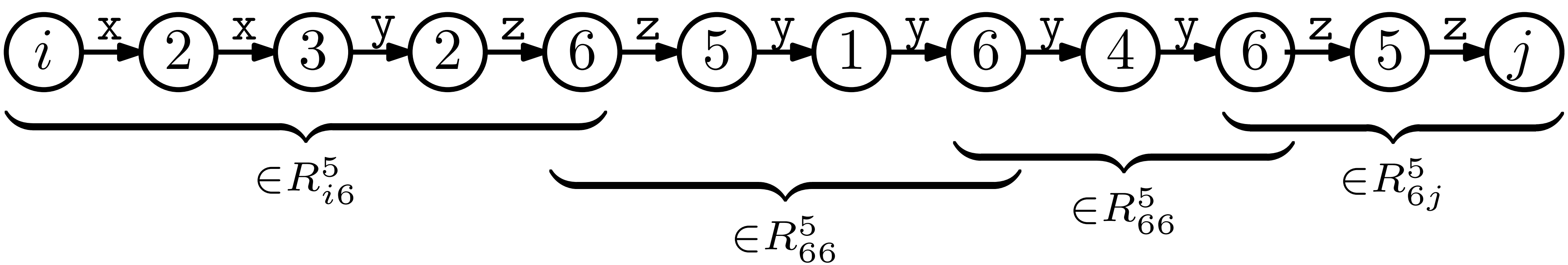
Rekursive Konstruktion

Basisfall (R_{ij}^0): nur direkte Übergänge

$$R_{ij}^0 := \begin{cases} \emptyset + a_1 + a_2 + \ldots & \text{falls } i \neq j \text{ und } a_i \in \{a \mid \delta(i,a) = j\} \\ \varepsilon + a_1 + a_2 + \ldots & \text{falls } i = j \text{ und } a_i \in \{a \mid \delta(i,a) = i\} \end{cases}$$

Rekursion (R_{ij}^k): ich kenne bereits alle R_{ij}^{k-1}

Bsp. von einem “Lauf” aus R_{ij}^6



Konstruktion der R_{ij}^k Terme

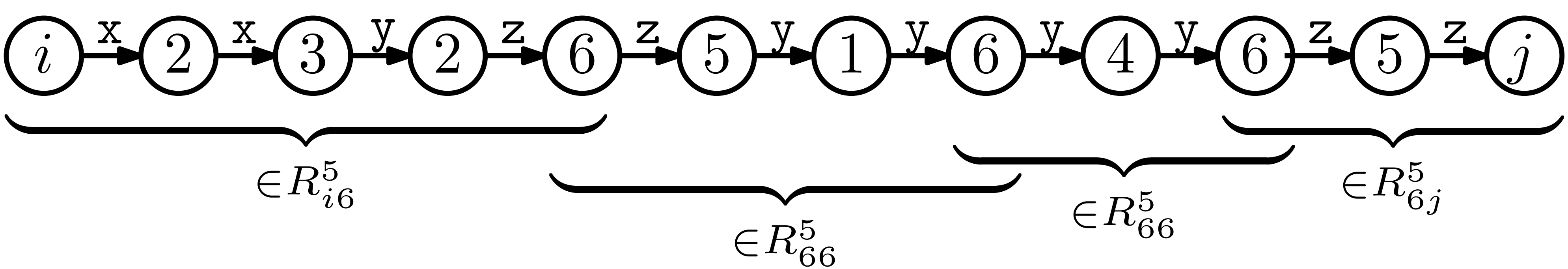
Rekursive Konstruktion

Basisfall (R_{ij}^0): nur direkte Übergänge

$$R_{ij}^0 := \begin{cases} \emptyset + a_1 + a_2 + \dots & \text{falls } i \neq j \text{ und } a_i \in \{a \mid \delta(i,a) = j\} \\ \varepsilon + a_1 + a_2 + \dots & \text{falls } i = j \text{ und } a_i \in \{a \mid \delta(i,a) = i\} \end{cases}$$

Rekursion (R_{ij}^k): ich kenne bereits alle R_{ij}^{k-1}

Bsp. von einem "Lauf" aus R_{ij}^6



$\rightarrow$ R_{**}^k **kann ich mit Termen** R_{**}^{k-1} **beschreiben**

Rekursion für R_{ij}^k

Rekursion für R_{ij}^k

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k

Rekursion für R_{ij}^k

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ij}^{k-1}$

Rekursion für R_{ij}^{k}

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k
  $\rightarrow R_{ij}^{k} \supseteq R_{ij}^{k-1}$
- 2. Möglichkeit: der *Lauf* eines Wortes besucht einmal oder mehrmals den Zustand k

Rekursion für R_{ij}^k

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ij}^{k-1}$

- 2. Möglichkeit: der *Lauf* eines Wortes besucht einmal oder mehrmals den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$

Rekursion für R^k_{ij}

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k

  $\rightarrow R^k_{ij} \supseteq R^{k-1}_{ij}$

- 2. Möglichkeit: der *Lauf* eines Wortes besucht einmal oder mehrmals den Zustand k

  $\rightarrow R^k_{ij} \supseteq R^{k-1}_{ik}(R^{k-1}_{kk})^* R^{k-1}_{kj}$

- es gibt keine andere Möglichkeit, deshalb

Rekursion für R_{ij}^k

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ij}^{k-1}$

- 2. Möglichkeit: der *Lauf* eines Wortes besucht einmal oder mehrmals den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$

- es gibt keine andere Möglichkeit, deshalb

$$\boxed{R_{ij}^k = R_{ij}^{k-1} + R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}}$$

Rekursion für R_{ij}^k

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ij}^{k-1}$

- 2. Möglichkeit: der *Lauf* eines Wortes besucht einmal oder mehrmals den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$

- es gibt keine andere Möglichkeit, deshalb

$$\boxed{R_{ij}^k = R_{ij}^{k-1} + R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}}$$

Berechnung mit ansteigendem k ($k = 0, \ldots, n$) aller R_{ij}^k Terme

Rekursion für R_{ij}^k

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ij}^{k-1}$

- 2. Möglichkeit: der *Lauf* eines Wortes besucht einmal oder mehrmals den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$

- es gibt keine andere Möglichkeit, deshalb

$$\boxed{R_{ij}^k = R_{ij}^{k-1} + R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}}$$

Berechnung mit ansteigendem k ($k = 0, \ldots, n$) aller R_{ij}^k Terme

□

Rekursion für R_{ij}^k

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ij}^{k-1}$

- 2. Möglichkeit: der *Lauf* eines Wortes besucht einmal oder mehrmals den Zustand k

  $\rightarrow R_{ij}^k \supseteq R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$

- es gibt keine andere Möglichkeit, deshalb

$$\boxed{R_{ij}^k = R_{ij}^{k-1} + R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}}$$

Berechnung mit ansteigendem k ($k = 0, \ldots, n$) aller R_{ij}^k Terme $\square$

Dynamisches Programmieren