Theoretische Grundlagen der Informatik
Übung

2. Übungstermin · 9. November 2017
Guido Brückner

INSTITUT FÜR THEORETISCHE INFORMATIK · LEHRSTUHL ALGORITHMIK

Gliederung

Inhalt

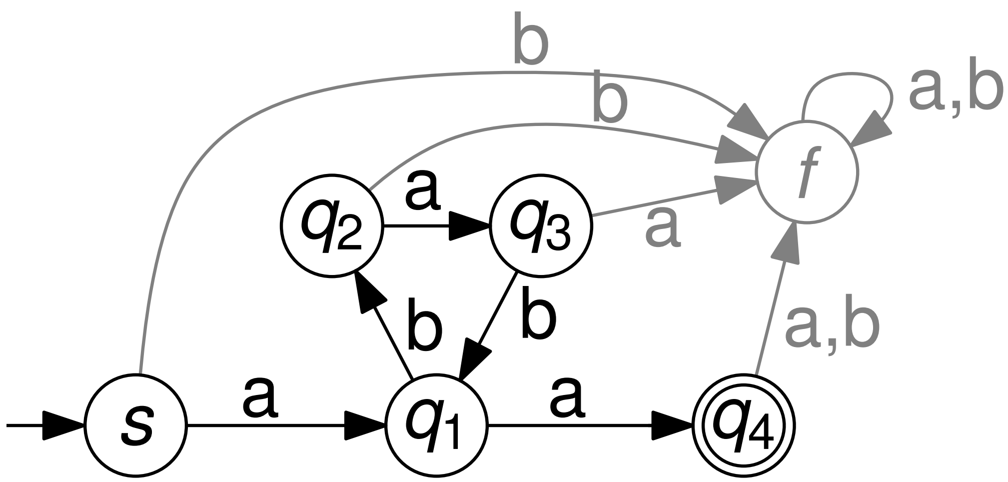
- Pumping Lemma
- Bestimmung eines regulären Ausdrucks
- Nichtdeterministische endliche Automaten
- ε-Abschluss
- Potenzmengenkonstruktion
- Minimierung von Automaten
- Cantors 2. Diagonalargument

Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \varepsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.



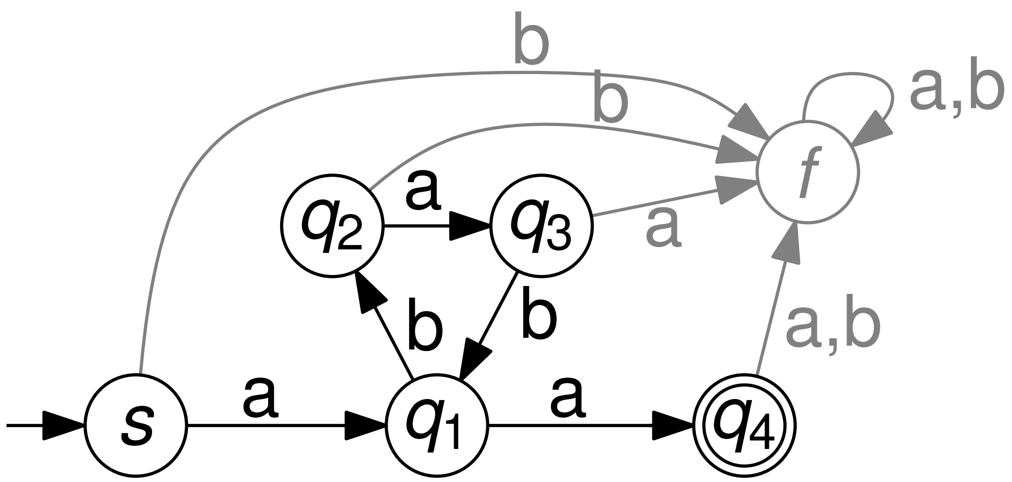
Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \varepsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Erklärung: Sei L reguläre Sprache und $\mathcal{A}$ entsprechender endlicher Automat.



Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

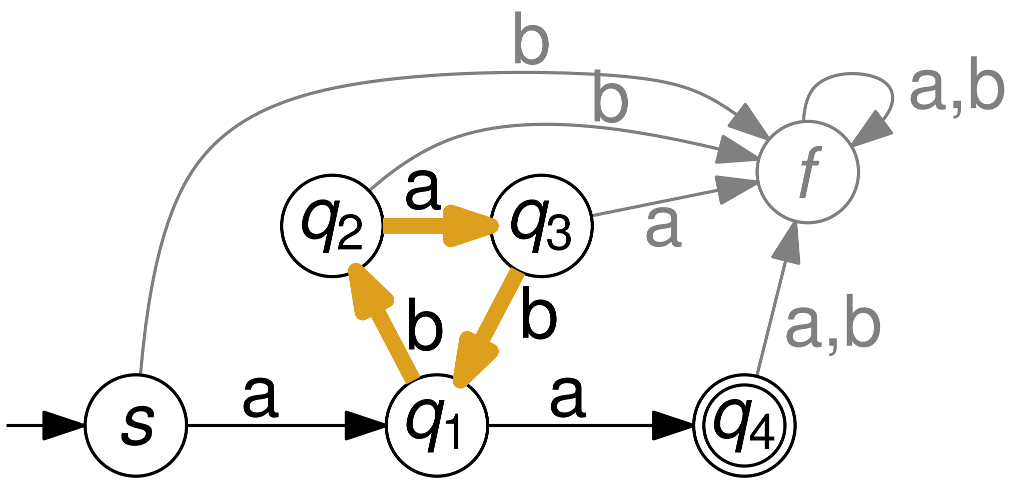
$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \varepsilon,$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$.

Erklärung: Sei L reguläre Sprache und $\mathcal{A}$ entsprechender endlicher Automat.

Egal wie viele Zustände für $\mathcal{A}$ verwendet werden, für jedes Wort $w \in L$, das mehr Zeichen hat als $\mathcal{A}$ Zustände, gilt

während der Abarbeitung von w durchläuft man einen Zyklus $\mathcal{Z}$ in $\mathcal{A}$.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \varepsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Erklärung: Sei L reguläre Sprache und $\mathcal{A}$ entsprechender endlicher Automat.

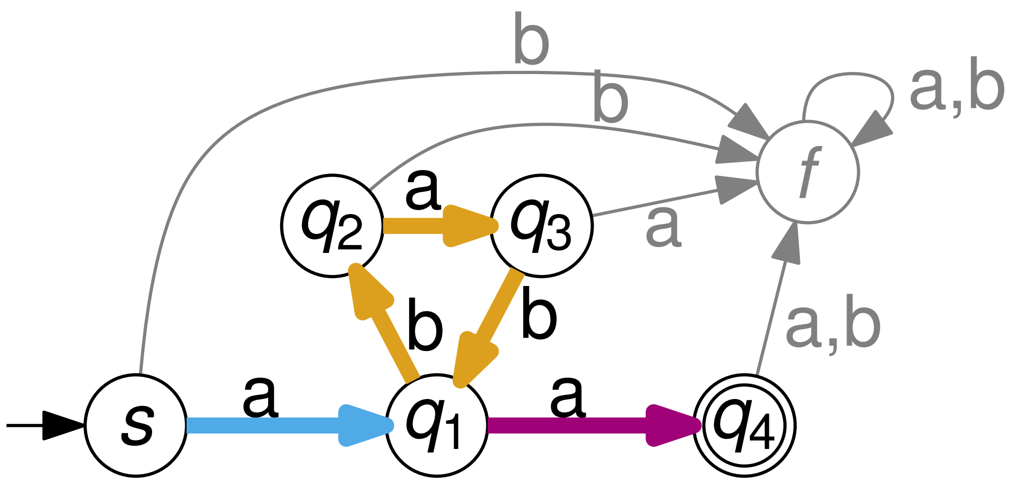
Egal wie viele Zustände für $\mathcal{A}$ verwendet werden, für jedes Wort $w \in L$, das mehr Zeichen hat als $\mathcal{A}$ Zustände, gilt

während der Abarbeitung von w durchläuft man einen Zyklus $\mathcal{Z}$ in $\mathcal{A}$.

Sei

- u das Teilwort von w, das vor $\mathcal{Z}$,
- v das Teilwort von w, das in $\mathcal{Z}$, und
- x das Teilwort von w, das nach $\mathcal{Z}$

abgearbeitet wird.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \varepsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Erklärung: Sei L reguläre Sprache und $\mathcal{A}$ entsprechender endlicher Automat.

Egal wie viele Zustände für $\mathcal{A}$ verwendet werden, für jedes Wort $w \in L$, das mehr Zeichen hat als $\mathcal{A}$ Zustände, gilt

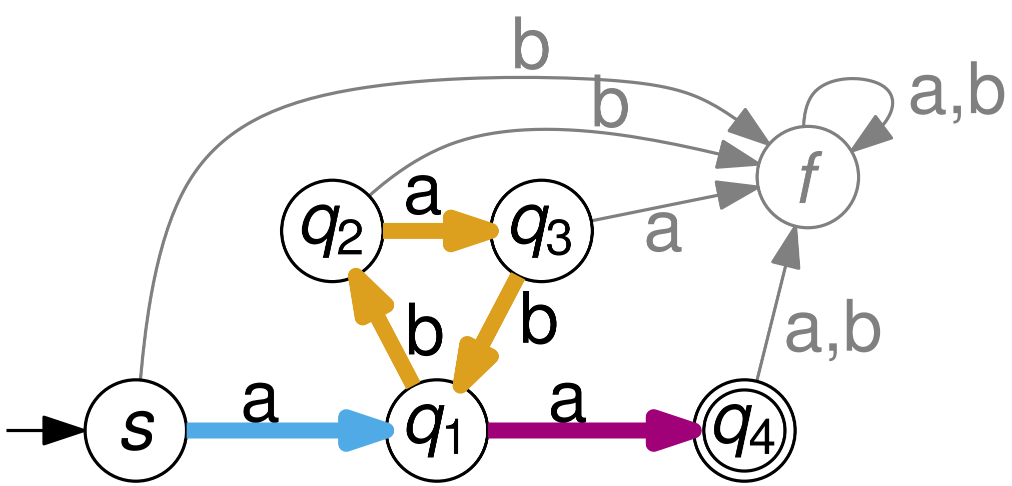
während der Abarbeitung von w durchläuft man einen Zyklus $\mathcal{Z}$ in $\mathcal{A}$.

Sei

- u das Teilwort von w, das vor $\mathcal{Z}$,
- v das Teilwort von w, das in $\mathcal{Z}$, und
- x das Teilwort von w, das nach $\mathcal{Z}$

abgearbeitet wird.

→ uv^ix (mit $i \in \mathbb{N}$) ist auch in L enthalten.
Durchlaufe $\mathcal{Z}$ entsprechend häufig.



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Zeigen oder widerlegen Sie, dass folgende Sprachen regulär sind.

(a) $L_1 = \{a^{2i} \in \{a, b\}^\star \mid i \in N_0\}$

(b) $L_2 = \{a^{i^2} \in \{a, b\}^\star \mid i \in N_0\}$

(c) $L_3 = \{w \in \{0, 1\}^\star \mid w$ enthält das Teilwort 000 genauso häufig wie das Teilwort 111$\}$

(d) $L_4 = \{w \in \{0, 1\}^\star \mid$ auf jedes Symbol 0 in w folgt das Symbol 1 und nach maximal dreimal 1 in w folgt das Symbol 0. $\}$

(e) $L_5 = \{a\}$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Zeigen oder widerlegen Sie, dass folgende Sprachen regulär sind.

(a) $L_1 = \{a^{2i} \in \{a, b\}^\star \mid i \in N_0\}$

- L_1 besteht aus dem leeren Wort und allen Worten, die aus einer geraden Anzahl an *a*'s bestehen.
- Sprache kann durch den regulären Ausdruck $(aa)^\star$ beschrieben werden

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Zeigen oder widerlegen Sie, dass folgende Sprachen regulär sind.

(a) $L_1 = \{a^{2i} \in \{a, b\}^\star \mid i \in N_0\}$ ✓

- L_1 besteht aus dem leeren Wort und allen Worten, die aus einer geraden Anzahl an a's bestehen.
- Sprache kann durch den regulären Ausdruck $(aa)^\star$ beschrieben werden

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Zeigen oder widerlegen Sie, dass folgende Sprachen regulär sind.

(b) $L_2 = \{a^{i^2} \in \{a, b\}^\star \mid i \in \mathbb{N}_0\}$

Annahme L_2 ist regulär $\rightarrow$ Pumping-Lemma gilt. Sei n die *Pumping-Zahl*.

Betrachte das Wort $w = a^{n^2} \in L_2$. Es gilt $|w| > n$

Es gibt Zerlegung $w = u\,v\,x \in L_2$ mit $|uv| \leq n$ und $v \neq \varepsilon$, sodass für alle $uv^i x \in L_2$ $(i \in \mathbb{N}_0)$

Beobachtung:

Beschreibe alle so möglichen Zerlegungen als: $\underbrace{a^p}_{u}\;\underbrace{a^q}_{v}\;\underbrace{a^r}_{x}$

mit $p + q + r = n^2$, $p + q \leq n$ und $1 \leq q \leq n$.

Nach Pumping-Lemma: $u\,v^2\,x \in L$

Also:

$$|a^{n^2}| < |uv^2x| = |a^p a^{2q} a^r| = |a^{n^2} a^q| \leq |a^{n^2} a^n| < |a^{n^2} a^{2n} a| = |a^{(n+1)^2}|$$

Es gibt kein $i \in \mathbb{N}$: $|uv^2x| = i^2$

↯

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Zeigen oder widerlegen Sie, dass folgende Sprachen regulär sind.

(c) $L_3 = \{w \in \{0,1\}^\star \mid w$ enthält das Teilwort 000 genauso häufig wie das Teilwort 111$\}$

Annahme L_3 ist regulär $\rightarrow$ Pumping-Lemma gilt. Sei n die *Pumping-Zahl*.

Betrachte das Wort $w = (000)^n(111)^n \in L_3$. Es gilt $|w| > n$

Es gibt Zerlegung $w = u\,v\,x \in L_3$ mit $|uv| \leq n$ und $v \neq \varepsilon$, sodass für alle $uv^ix \in L_3$ $(i \in \mathbb{N}_0)$

Beobachtung:

Beschreibe alle so möglichen Zerlegungen als: $\underbrace{0^p}_{u}\ \underbrace{0^q}_{v}\ \underbrace{0^{3n-p-q}\ (111)^n}_{x \text{ mit } |x| = r}$

mit $p + q + r = 6n$, $p + q \leq n$ und $1 \leq q \leq n$.

Nach Pumping-Lemma: $u\,v^2\,x \in L_3$

Also:

$$0^p\ 0^{2q}\ 0^{3n-p-q}\ (111)^n = 0^{3n+q}\ (111)^n \notin L_3$$

↯

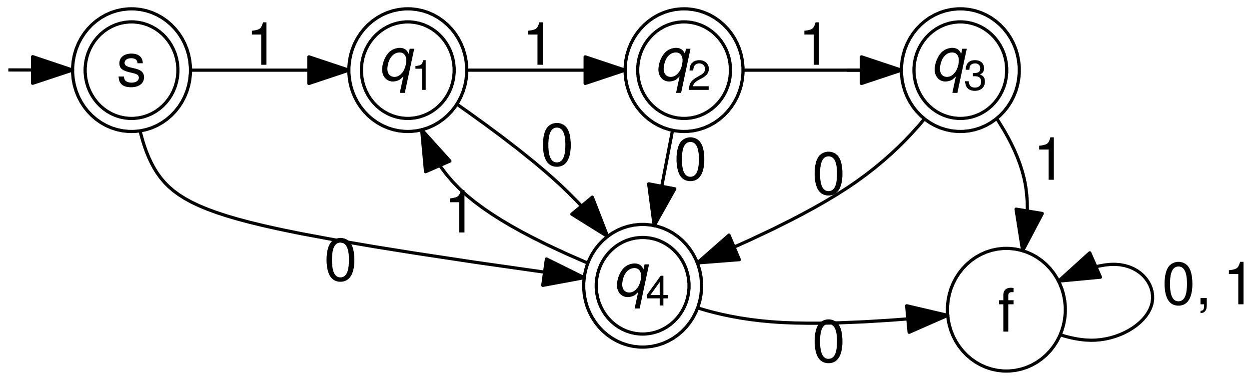
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Zeigen oder widerlegen Sie, dass folgende Sprachen regulär sind.

(d) $L_4 = \{ w \in \{0,1\}^\star \mid$ auf jedes Symbol 0 in w folgt das Symbol 1 und nach maximal dreimal 1 in w folgt das Symbol 0. $\}$

✓



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Zeigen oder widerlegen Sie, dass folgende Sprachen regulär sind.

(e) $L_5 = \{a\}$

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \varepsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

- L_5 ist regulär
- $L_5 \cap \left(\bigcup_{j>1} \{a\}^j \right) = \emptyset$
- Pumping-Lemma ist auch erfüllt, z.B. mit $n = 1$

Pumping-Lemma

Zeigen oder widerlegen Sie, dass folgende Sprachen regulär sind.

(a) $L_1 = \{a^{2i} \in \{a, b\}^\star \mid i \in N_0\}$ ✓

(b) $L_2 = \{a^{i^2} \in \{a, b\}^\star \mid i \in N_0\}$ ↯

(c) $L_3 = \{w \in \{0, 1\}^\star \mid$ w enthält das Teilwort 000 genauso häufig wie das Teilwort 111$\}$ ↯

(d) $L_4 = \{w \in \{0, 1\}^\star \mid$ auf jedes Symbol 0 in w folgt das Symbol 1 und nach maximal dreimal 1 in w folgt das Symbol 0. $\}$ ✓

(e) $L_5 = \{a\}$ ✓

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Pumping-Lemma

Zeigen oder widerlegen Sie, dass folgende Sprachen regulär sind.

(a) $L_1 = \{a^{2i} \in \{a, b\}^\star \mid i \in N_0\}$

Mit dem Pumpinglemma kann man nicht zeigen, dass eine Sprache regulär ist, man kann es höchstens widerlegen!

wie

und

nach maximal dreimal 1 in w folgt das Symbol 0. $\}$

(e) $L_5 = \{a\}$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Gegeben: deterministischer endlicher Automat $\mathcal{A}$

Gesucht: regulärer Ausdruck, der $L(\mathcal{A})$ erzeugt

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^* L_{q_{i+1},i,q_t}\right)$$

Bestimmung eines regulären Ausdrucks

Gegeben: deterministischer endlicher Automat $\mathcal{A}$

Gesucht: regulärer Ausdruck, der $L(\mathcal{A})$ erzeugt

$$
\begin{aligned}
L_{q_r,i+1,q_t} &= L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^* L_{q_{i+1},i,q_t}\right) \\
&= \text{Sprache, die jedes Wort } w \text{ enthält, so dass wenn man in } \mathcal{A} \text{ im Zustand } q_r \text{ startet und } w \\
&\quad \text{abarbeitet und nur die Zustände } q_1, \ldots, q_{i+1} \text{ benutzt, man in Zustand } q_t \text{ enden kann.}
\end{aligned}
$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

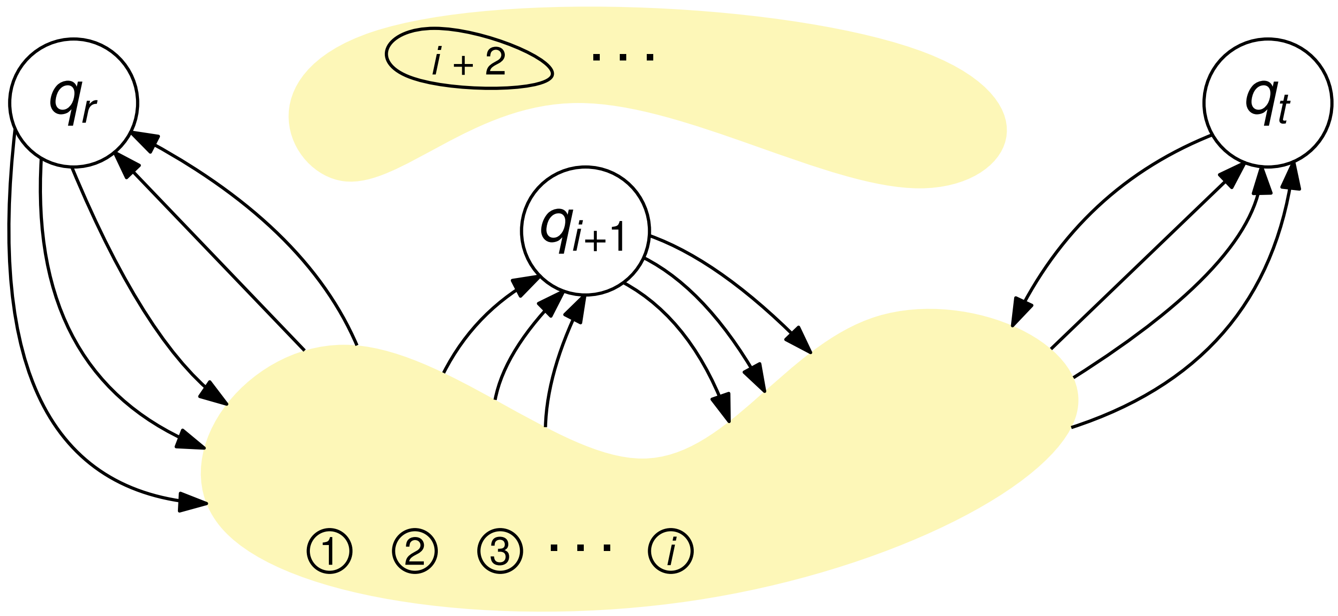
Bestimmung eines regulären Ausdrucks

Gegeben: deterministischer endlicher Automat $\mathcal{A}$

Gesucht: regulärer Ausdruck, der $L(\mathcal{A})$ erzeugt

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^* L_{q_{i+1},i,q_t} \right)$$

Erklärung:



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

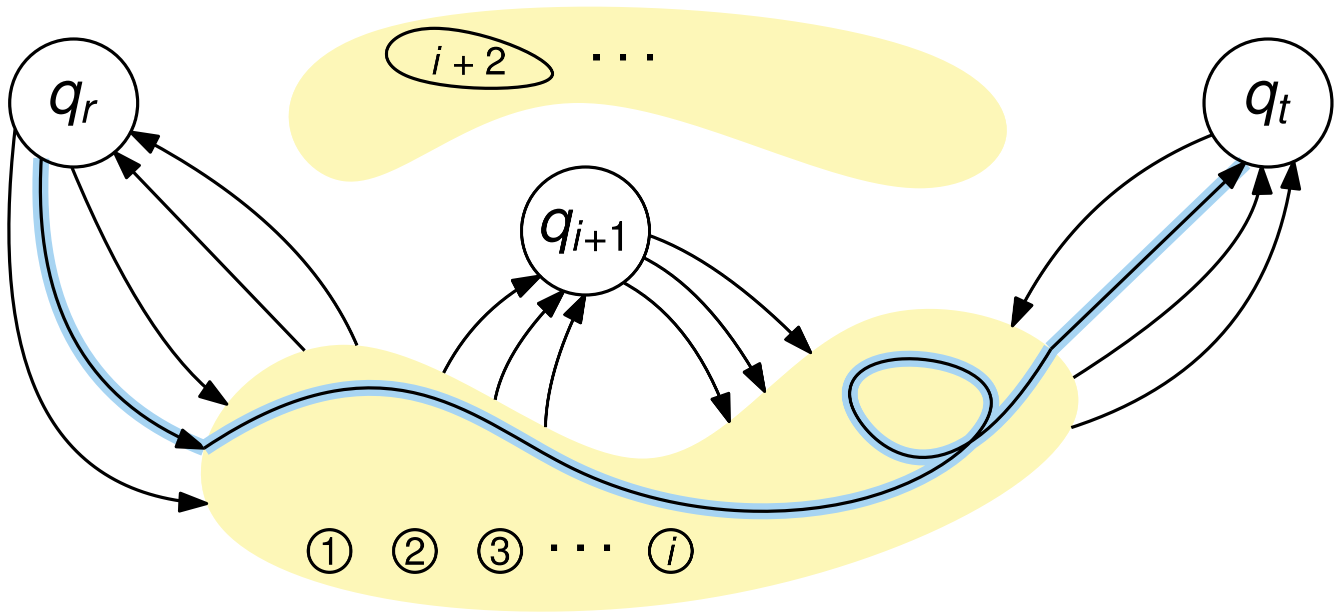
Bestimmung eines regulären Ausdrucks

Gegeben: deterministischer endlicher Automat $\mathcal{A}$

Gesucht: regulärer Ausdruck, der $L(\mathcal{A})$ erzeugt

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^* L_{q_{i+1},i,q_t} \right)$$

Erklärung:



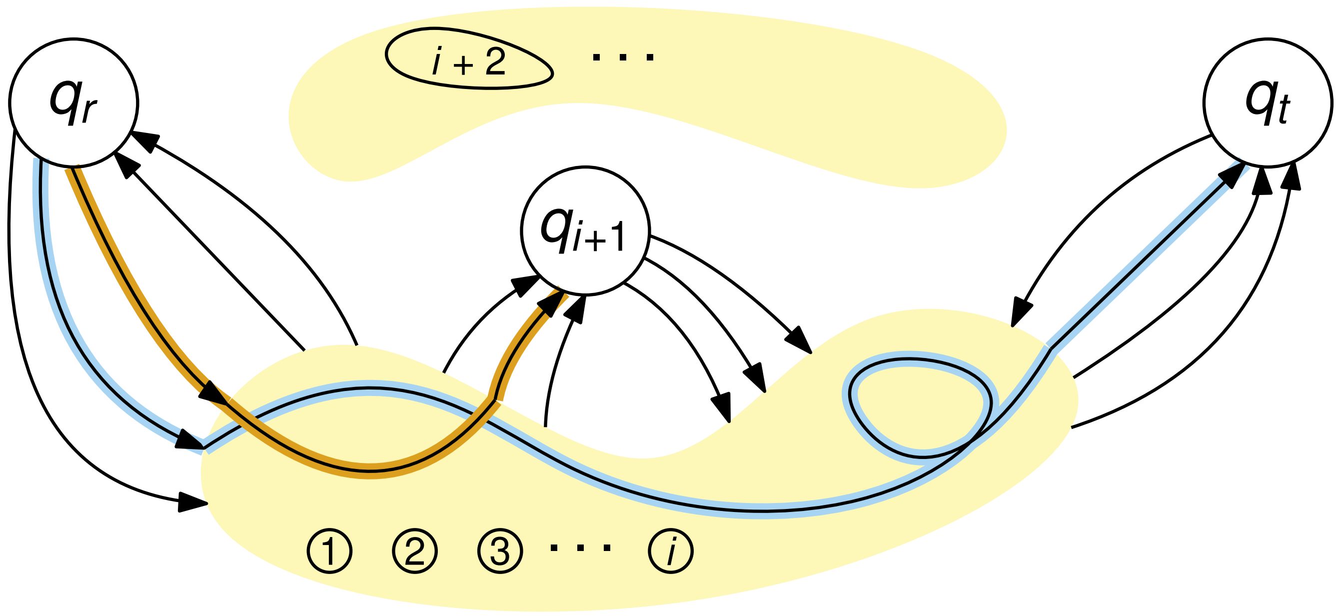
Bestimmung eines regulären Ausdrucks

Gegeben: deterministischer endlicher Automat $\mathcal{A}$

Gesucht: regulärer Ausdruck, der $L(\mathcal{A})$ erzeugt

$$L_{q_r,i+1,q_t} = \boxed{L_{q_r,i,q_t}} \cup \left(\boxed{L_{q_r,i,q_{i+1}}}(L_{q_{i+1},i,q_{i+1}})^* L_{q_{i+1},i,q_t}\right)$$

Erklärung:



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

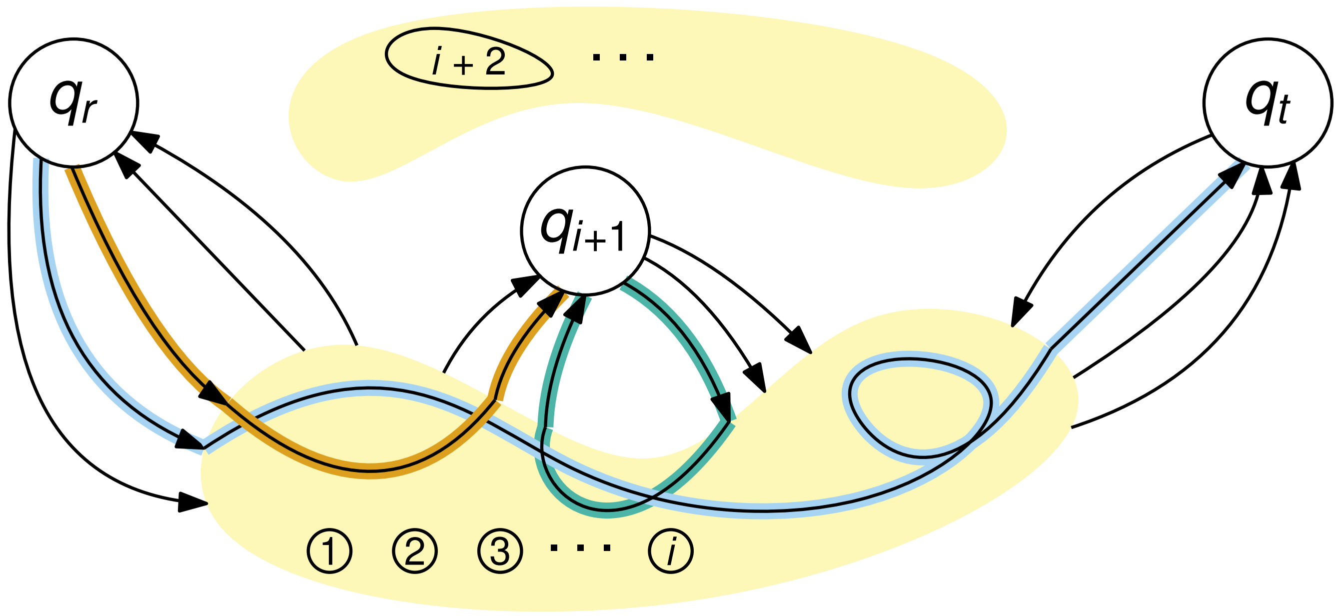
Bestimmung eines regulären Ausdrucks

Gegeben: deterministischer endlicher Automat $\mathcal{A}$

Gesucht: regulärer Ausdruck, der $L(\mathcal{A})$ erzeugt

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^* L_{q_{i+1},i,q_t} \right)$$

Erklärung:



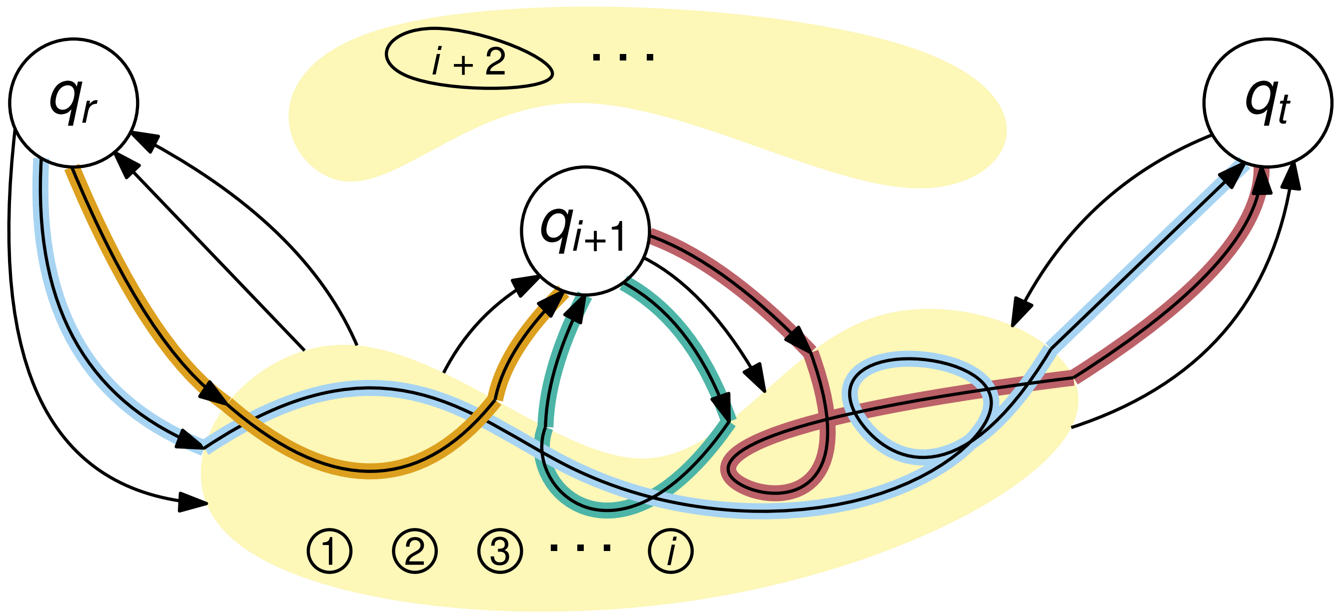
Bestimmung eines regulären Ausdrucks

Gegeben: deterministischer endlicher Automat $\mathcal{A}$

Gesucht: regulärer Ausdruck, der $L(\mathcal{A})$ erzeugt

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} \left(L_{q_{i+1},i,q_{i+1}} \right)^* L_{q_{i+1},i,q_t} \right)$$

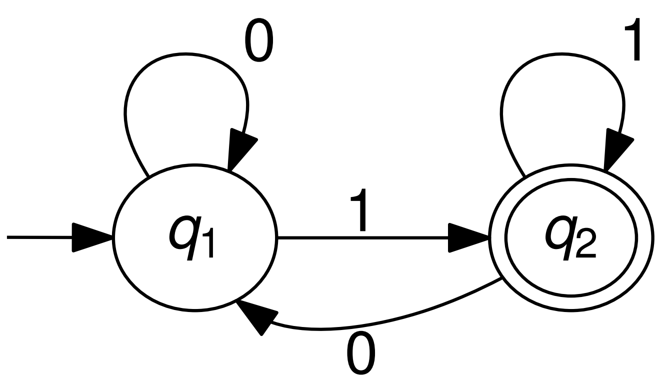
Erklärung:



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

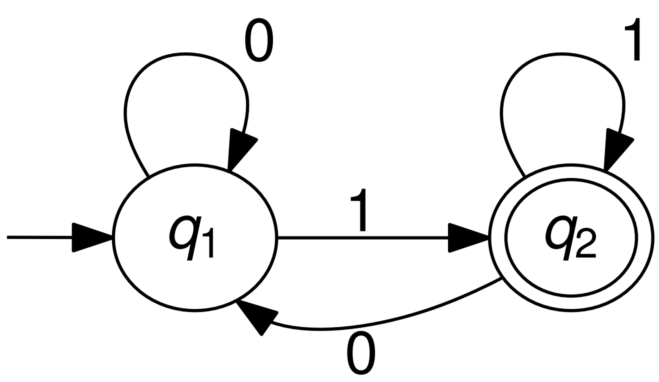


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

$L_{1,2,2}$



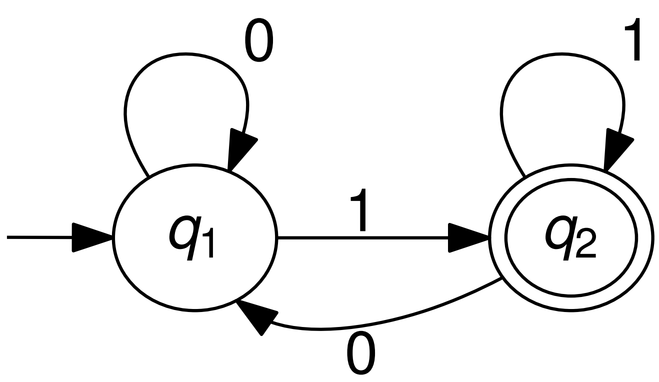
$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t} \right)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

$$L_{1,2,2} = L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2})$$



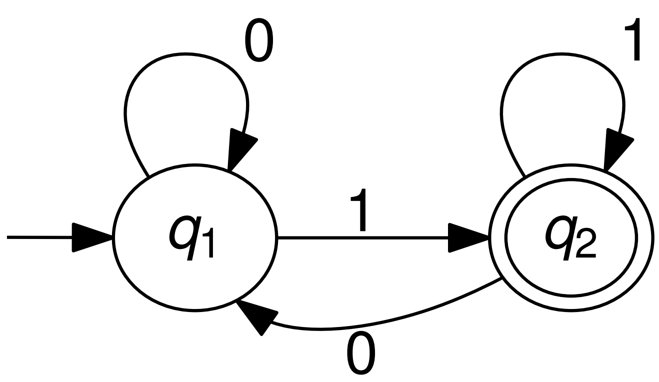
$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t} \right)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

$$L_{1,2,2} = L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2})$$
$$L_{1,1,2} = L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2})$$

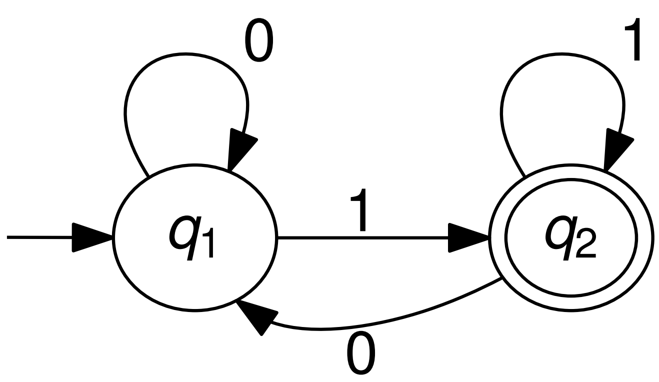


$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t}\right)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



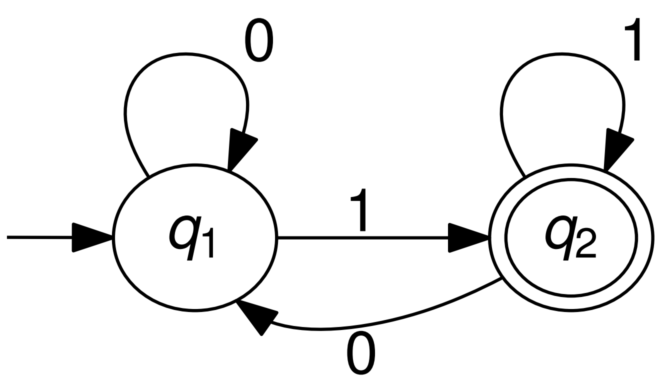
$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2})
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t}\right)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

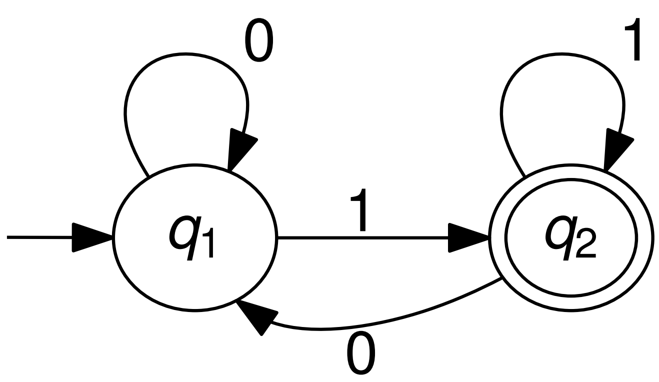


$$\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0
\end{aligned}$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t} \right)$$

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

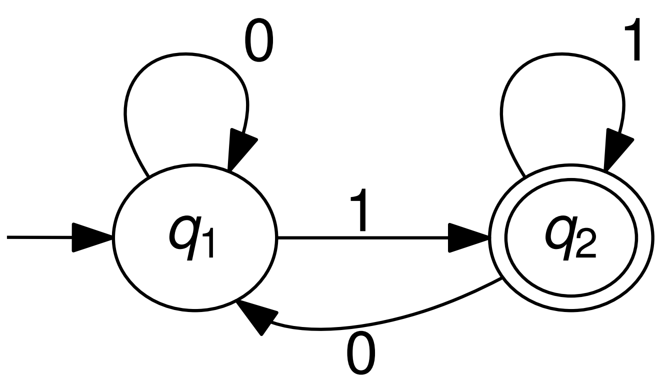


$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t} \right)$$

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

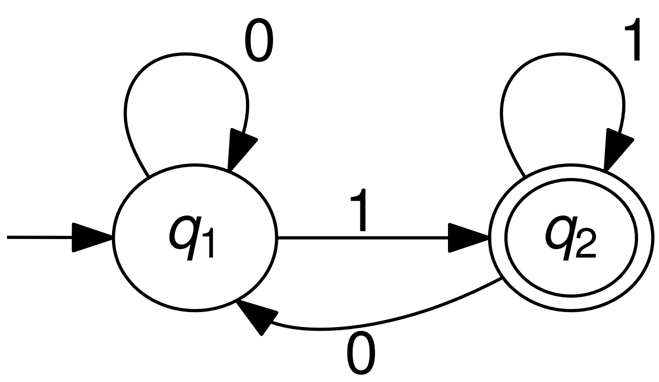


$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t}\right)$$

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

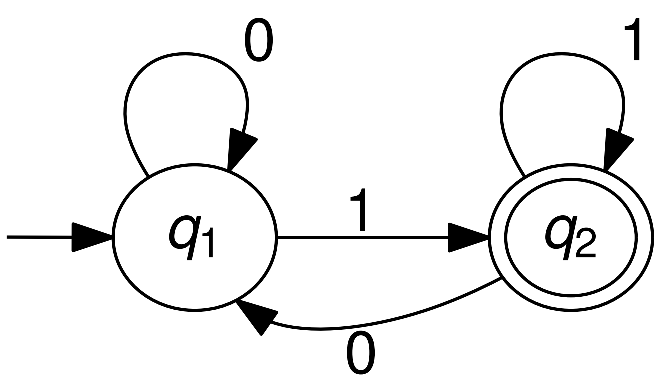


$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t} \right)$$

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



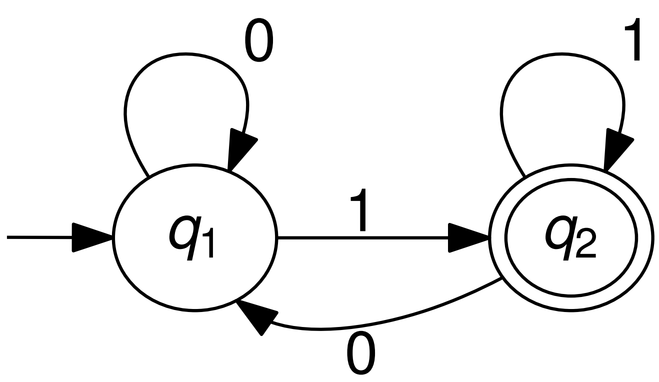
$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^\star_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^\star_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^\star_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^\star L_{q_{i+1},i,q_t}\right)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

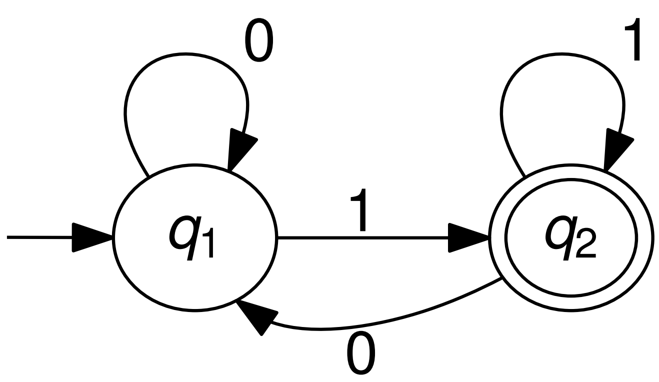


$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t} \right)$$

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

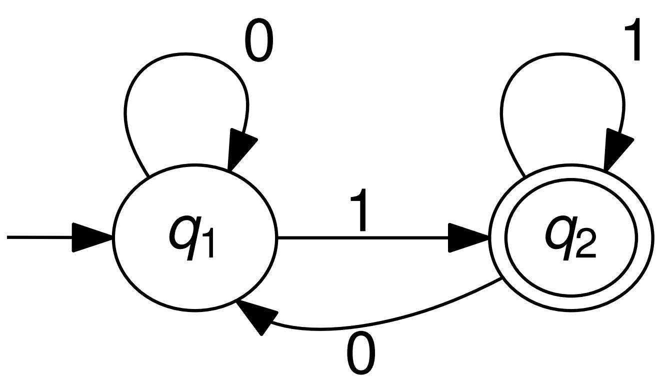


$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^{\star} 1)
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t} \right)$$

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



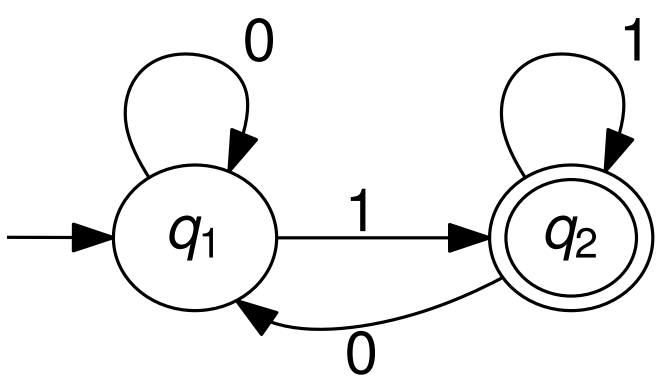
$$\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^{\star} 1) \\
&= 1 \cup (\quad 0^{\star} \quad 1)
\end{aligned}$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t} \right)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



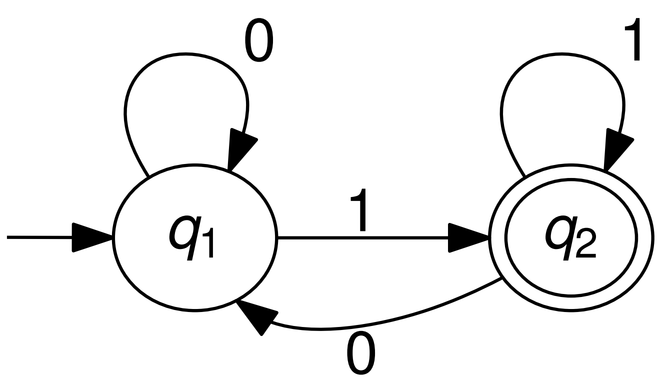
$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L_{2,1,2}^{\star} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L_{1,0,1}^{\star} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L_{1,0,1}^{\star} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^{\star} 1) \\
&= 1 \cup (0^{\star} 1) = 0^{\star} 1
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t} \right)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



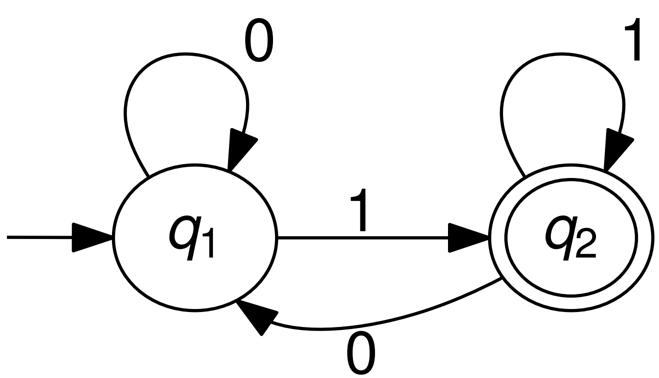
$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^{\star} 1) = 0^{\star} 1 \\
L_{2,1,2} &= (\varepsilon \cup 1) \cup (0(\varepsilon \cup 0)^{\star} 1)
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^{\star} L_{q_{i+1},i,q_t}\right)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

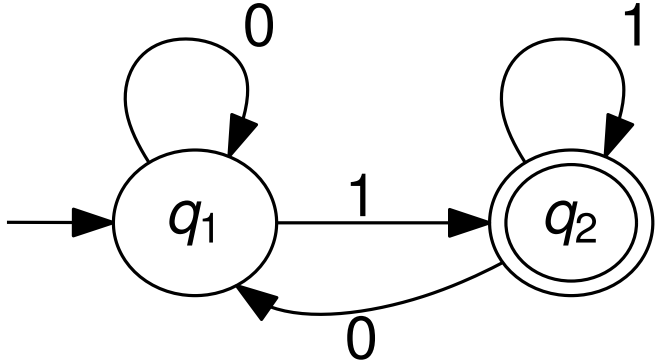


$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^\star_{2,1,2} L_{2,1,2})\\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^\star_{1,0,1} L_{1,0,2})\\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^\star_{1,0,1} L_{1,0,2})\\
L_{1,0,1} &= \varepsilon \cup 0\\
L_{1,0,2} &= 1\\
L_{2,0,2} &= \varepsilon \cup 1\\
L_{2,0,1} &= 0\\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^\star 1) = 0^\star 1\\
L_{2,1,2} &= (\varepsilon \cup 1) \cup (0(\varepsilon \cup 0)^\star 1)
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^\star L_{q_{i+1},i,q_t}\right)$$

Bestimmung eines regulären Ausdrucks

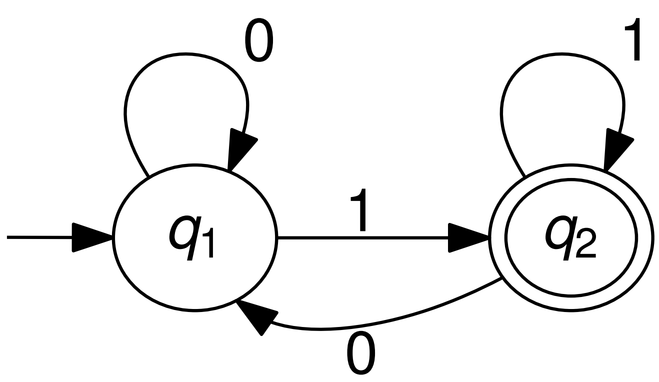
Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^{\star} 1) = 0^{\star} 1 \\
L_{2,1,2} &= (\varepsilon \cup 1) \cup (0 (\varepsilon \cup 0)^{\star} 1) \\
&= (\varepsilon \cup 1) \cup (0 \; 0^{\star} \; 1)
\end{aligned}
$$

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



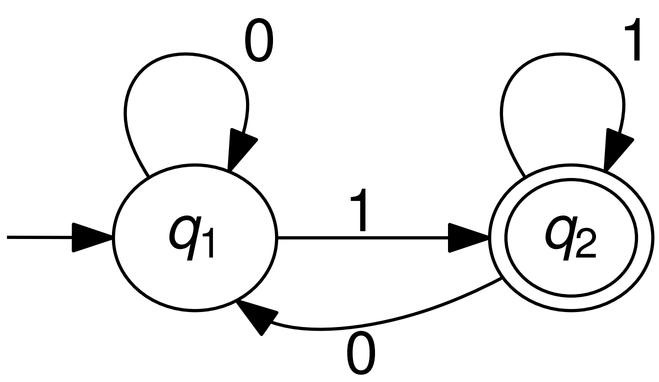
$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L_{2,1,2}^{\star} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L_{1,0,1}^{\star} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L_{1,0,1}^{\star} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^{\star} 1) \; = 0^{\star} 1 \\
L_{2,1,2} &= (\varepsilon \cup 1) \cup (00^{\star} 1)
\end{aligned}
$$

$$L_{1,2,2} = (0^{\star}1) \cup ((0^{\star}1)((\varepsilon \cup 1) \cup (00^{\star}1))^{\star}((\varepsilon \cup 1) \cup (00^{\star}1)))$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



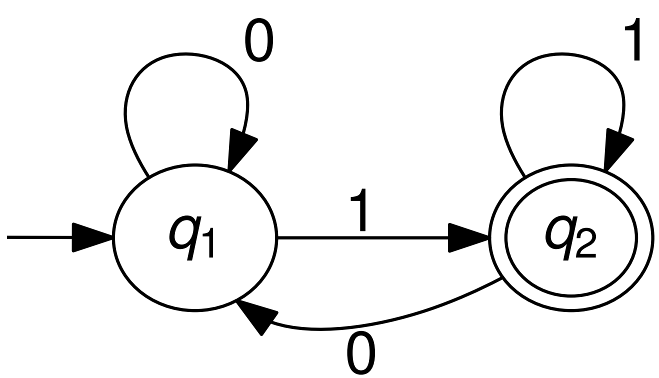
$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^{\star} 1) = 0^{\star} 1 \\
L_{2,1,2} &= (\varepsilon \cup 1) \cup (00^{\star} 1)
\end{aligned}
$$

$$
\begin{aligned}
L_{1,2,2} &= (0^{\star}1) \cup ((0^{\star}1)((\varepsilon \cup 1) \cup (00^{\star}1))^{\star}((\varepsilon \cup 1) \cup (00^{\star}1))) \\
&= (0^{\star}1) \cup (0^{\star}1)((\varepsilon \cup 1) \cup (00^{\star}1))^{+}
\end{aligned}
$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



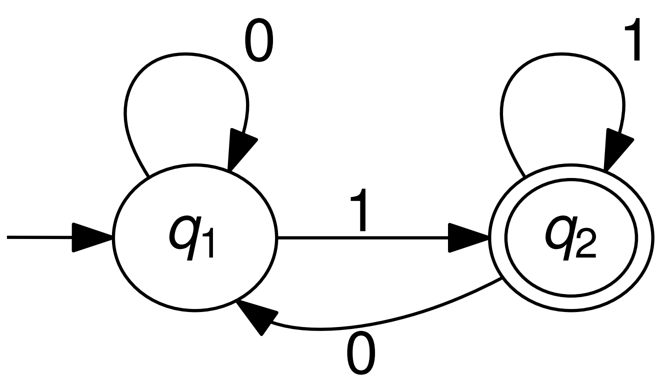
$$\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^{\star} 1) = 0^{\star} 1 \\
L_{2,1,2} &= (\varepsilon \cup 1) \cup (00^{\star} 1)
\end{aligned}$$

$$\begin{aligned}
L_{1,2,2} &= (0^{\star}1) \cup (0^{\star}1)((\varepsilon \cup 1) \cup (00^{\star}1))^{+} \\
&= (0^{\star}1)((\varepsilon \cup 1) \cup (00^{\star}1))^{\star}
\end{aligned}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache



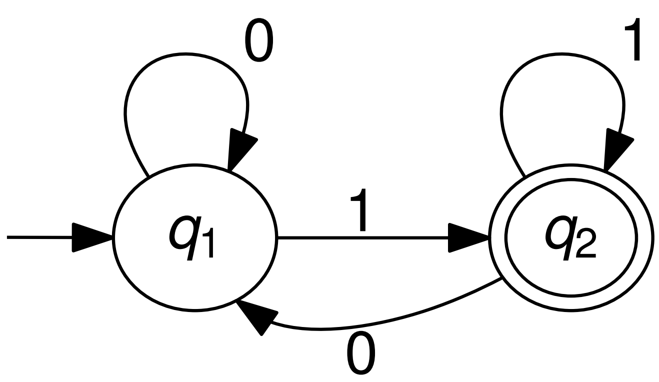
$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^{\star}_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^{\star}_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^{\star} 1) \ = 0^{\star} 1 \\
L_{2,1,2} &= (\varepsilon \cup 1) \cup (00^{\star} 1)
\end{aligned}
$$

$$
\begin{aligned}
L_{1,2,2} &= (0^{\star} 1)((\varepsilon \cup 1) \cup (00^{\star} 1))^{\star} \\
&= (0^{\star} 1) \quad (1 \cup (00^{\star} 1))^{\star}
\end{aligned}
$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Bestimmung eines regulären Ausdrucks

Bestimmen Sie nach der Methode aus der Vorlesung (Satz 2.14) einen regulären Ausdruck für die von folgendem Automaten $\mathcal{A}$ erkannte Sprache

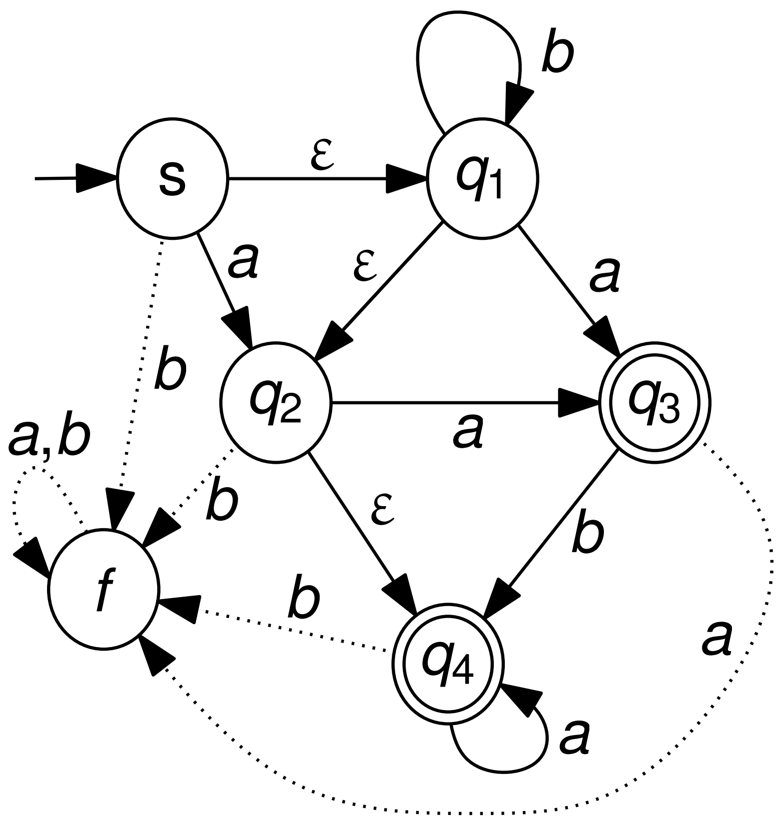


$$
\begin{aligned}
L_{1,2,2} &= L_{1,1,2} \cup (L_{1,1,2} L^\star_{2,1,2} L_{2,1,2}) \\
L_{1,1,2} &= L_{1,0,2} \cup (L_{1,0,1} L^\star_{1,0,1} L_{1,0,2}) \\
L_{2,1,2} &= L_{2,0,2} \cup (L_{2,0,1} L^\star_{1,0,1} L_{1,0,2}) \\
L_{1,0,1} &= \varepsilon \cup 0 \\
L_{1,0,2} &= 1 \\
L_{2,0,2} &= \varepsilon \cup 1 \\
L_{2,0,1} &= 0 \\
L_{1,1,2} &= 1 \cup ((\varepsilon \cup 0)(\varepsilon \cup 0)^\star 1) \ = 0^\star 1 \\
L_{2,1,2} &= (\varepsilon \cup 1) \cup (00^\star 1) \\
L_{1,2,2} &= (0^\star 1)(1 \cup (00^\star 1))^\star
\end{aligned}
$$

$$L_{q_r,i+1,q_t} = L_{q_r,i,q_t} \cup \left(L_{q_r,i,q_{i+1}} (L_{q_{i+1},i,q_{i+1}})^\star L_{q_{i+1},i,q_t}\right)$$

NEA

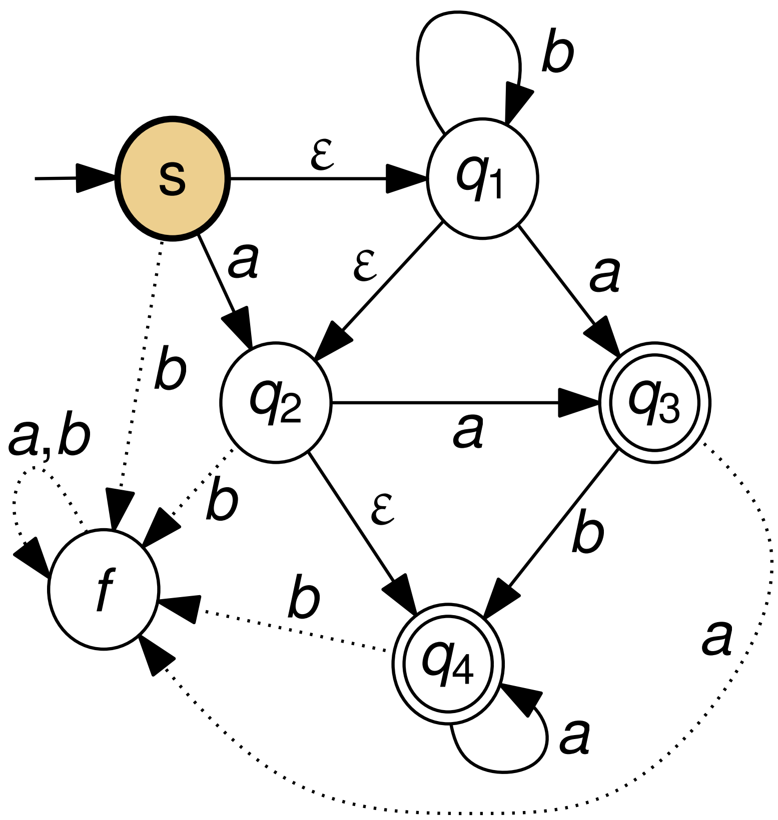
Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:



NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.

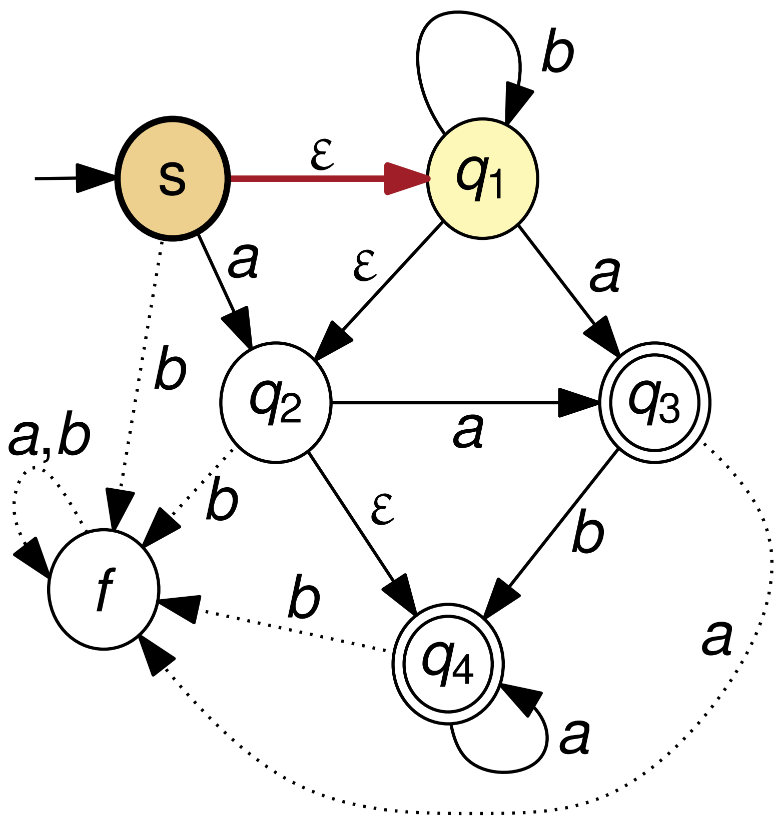


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

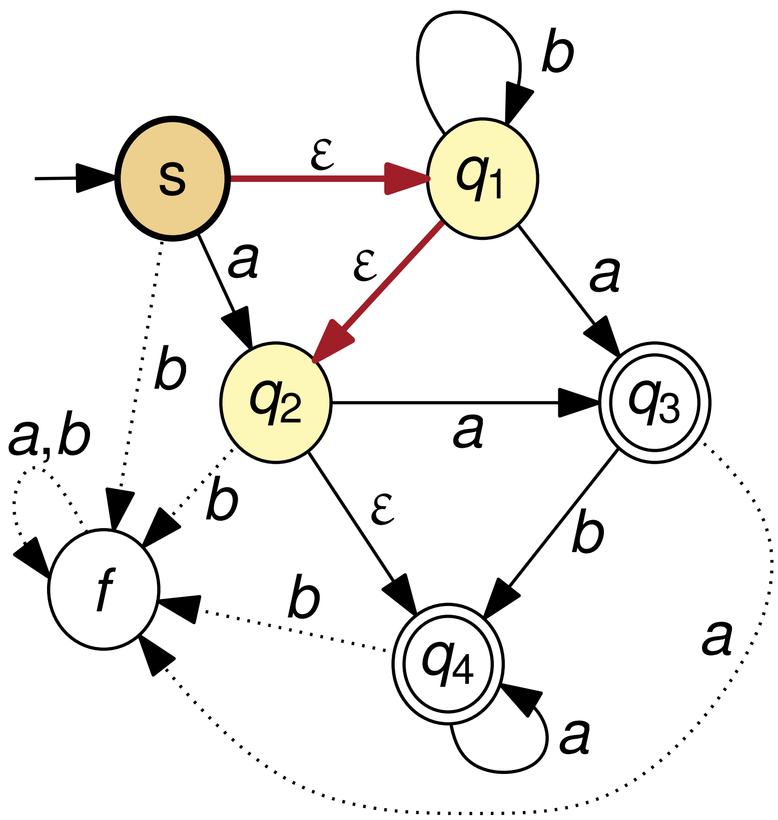
(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.

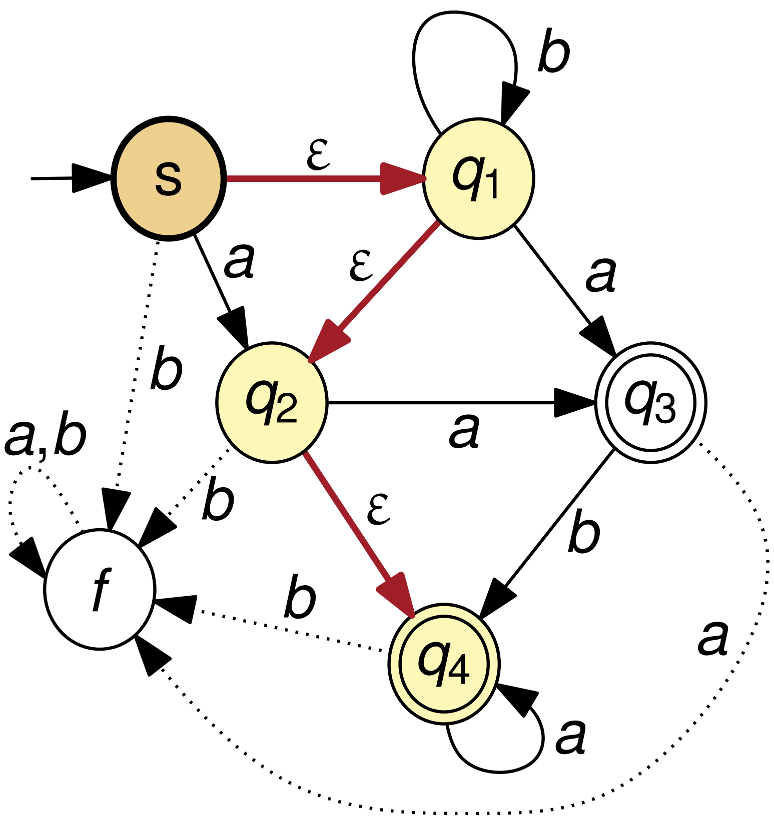


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.

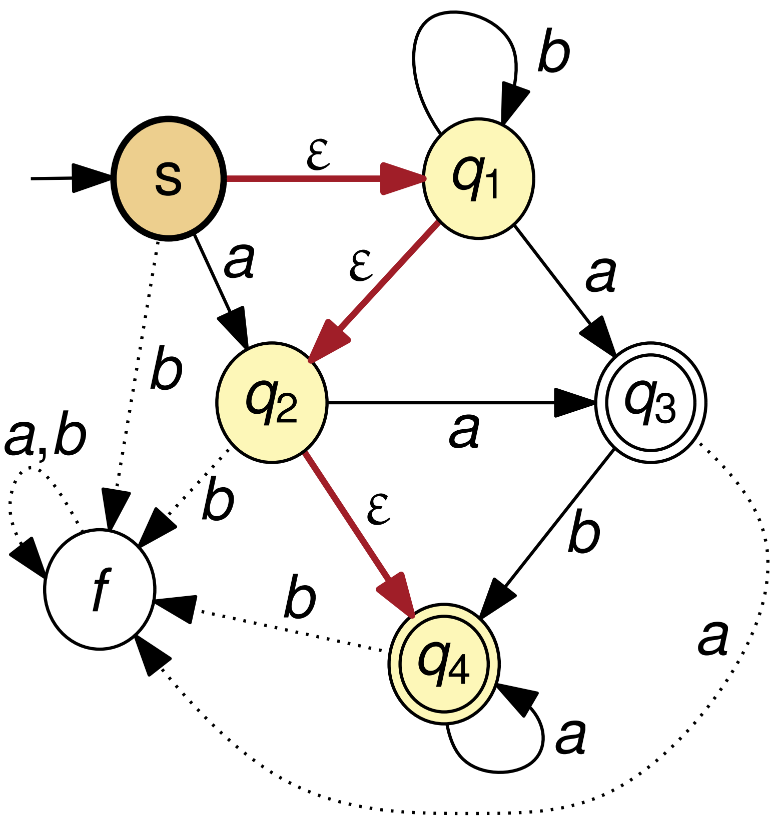


Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



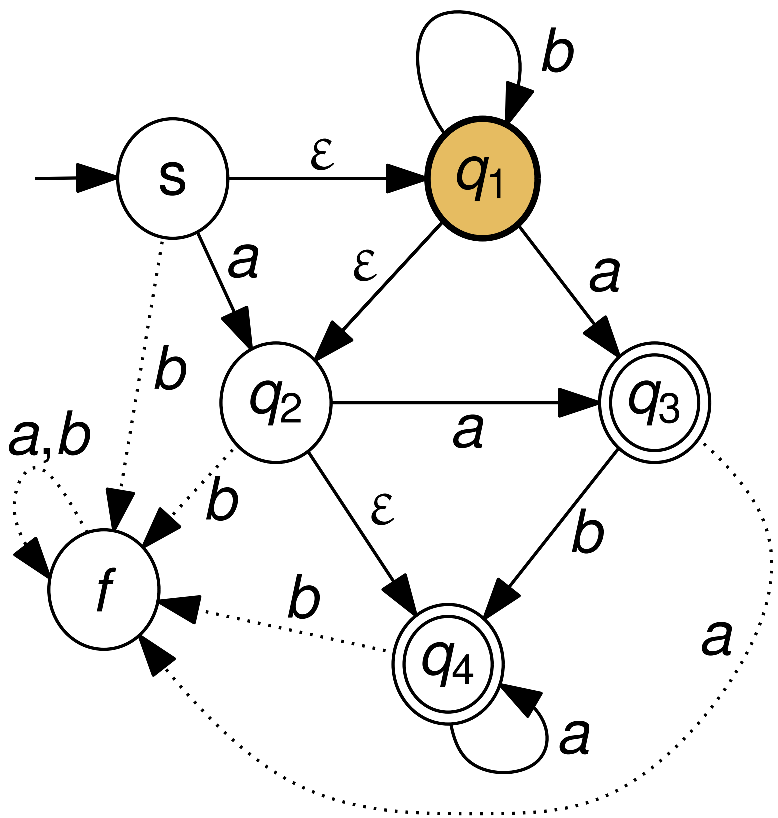
$$E(s) = \{s, q_1, q_2, q_4\},$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



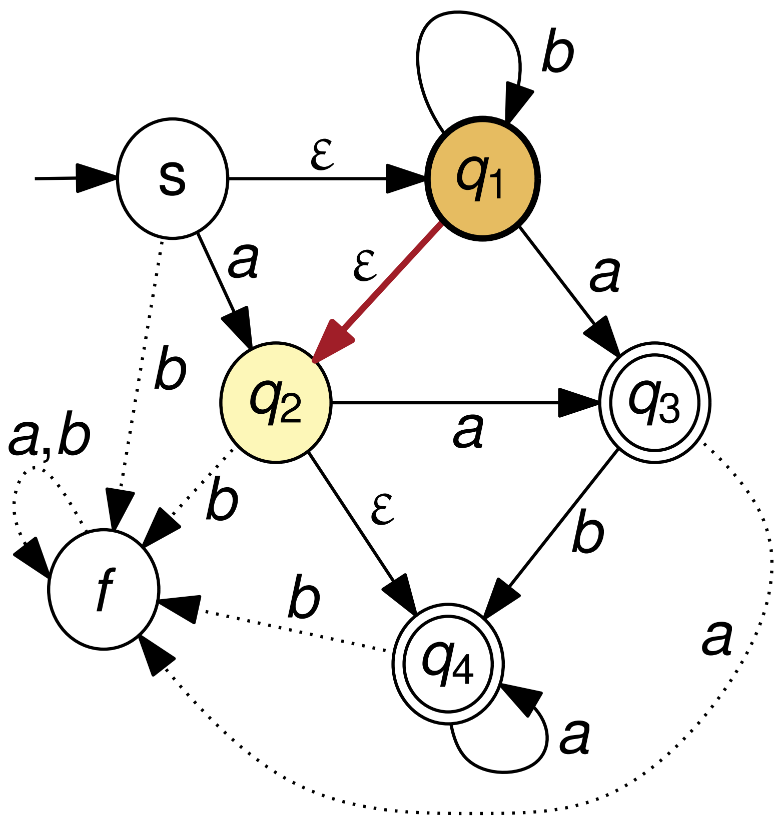
$$E(s) = \{s, q_1, q_2, q_4\},$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.

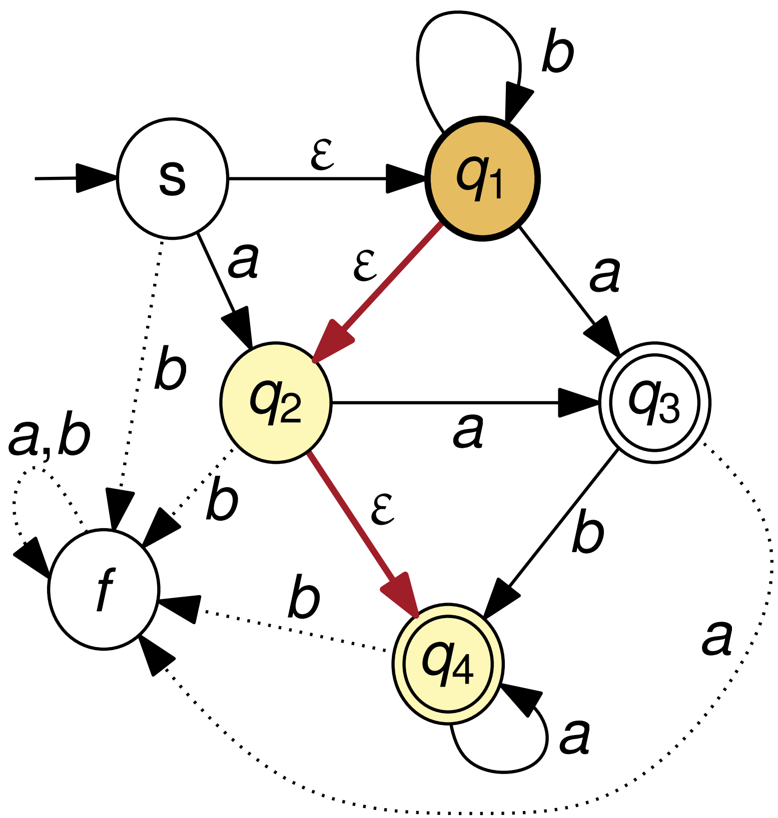


$$E(s) = \{s, q_1, q_2, q_4\},$$

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



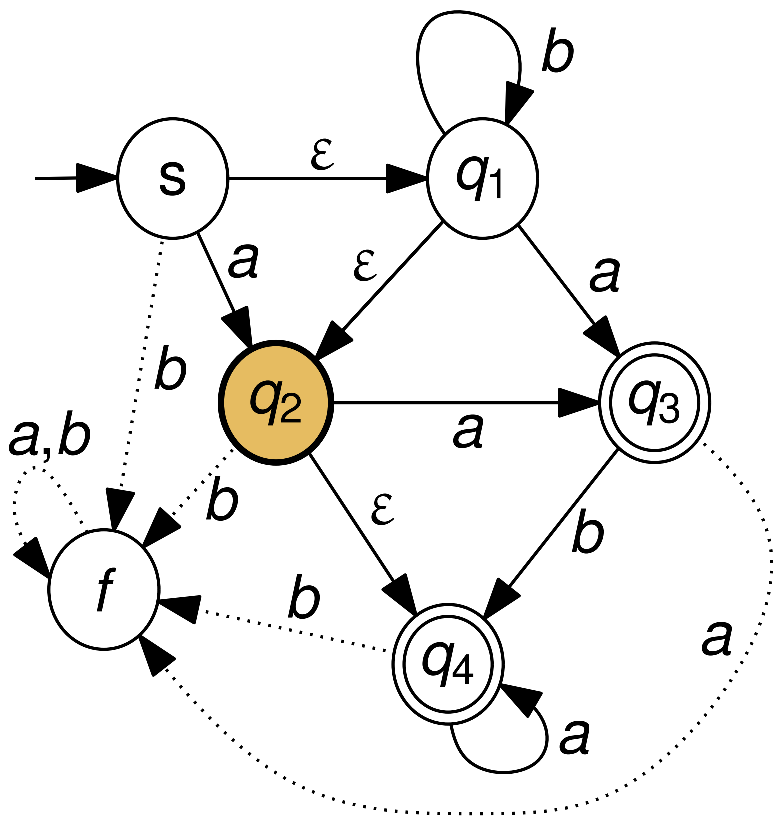
$$E(s) = \{s, q_1, q_2, q_4\},$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



$$E(s) = \{s, q_1, q_2, q_4\},$$

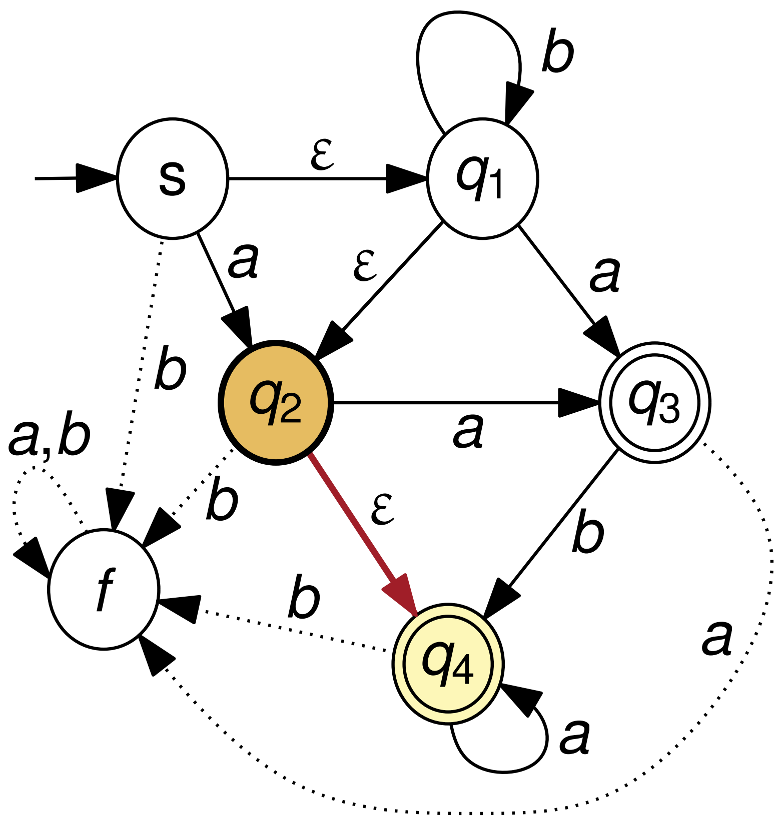
$$E(q_1) = \{q_1, q_2, q_4\},$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



$$E(s) = \{s, q_1, q_2, q_4\},$$

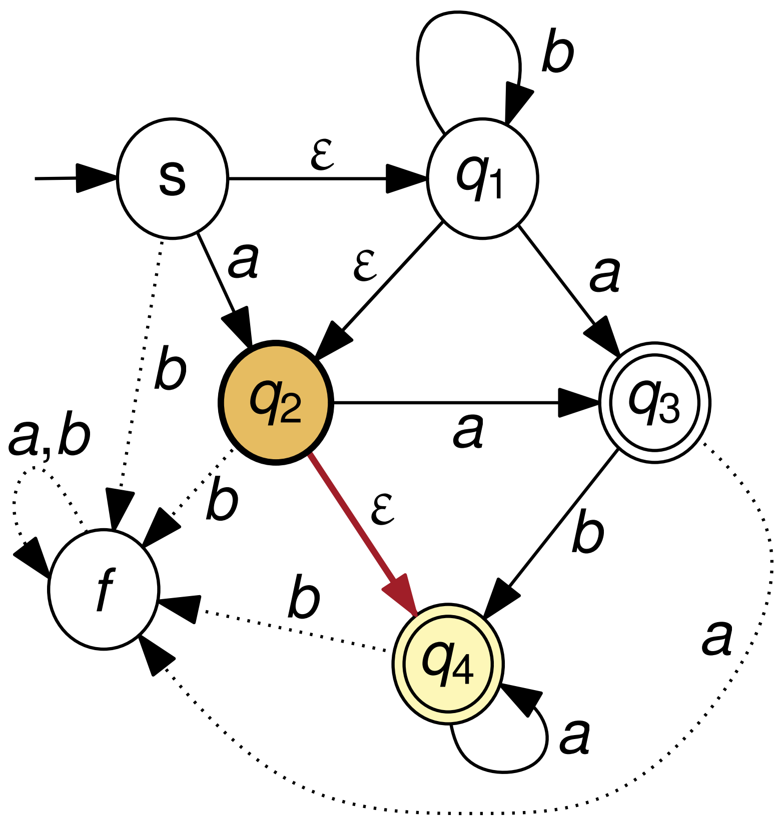
$$E(q_1) = \{q_1, q_2, q_4\},$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



$$E(s) = \{s, q_1, q_2, q_4\},$$

$$E(q_1) = \{q_1, q_2, q_4\},$$

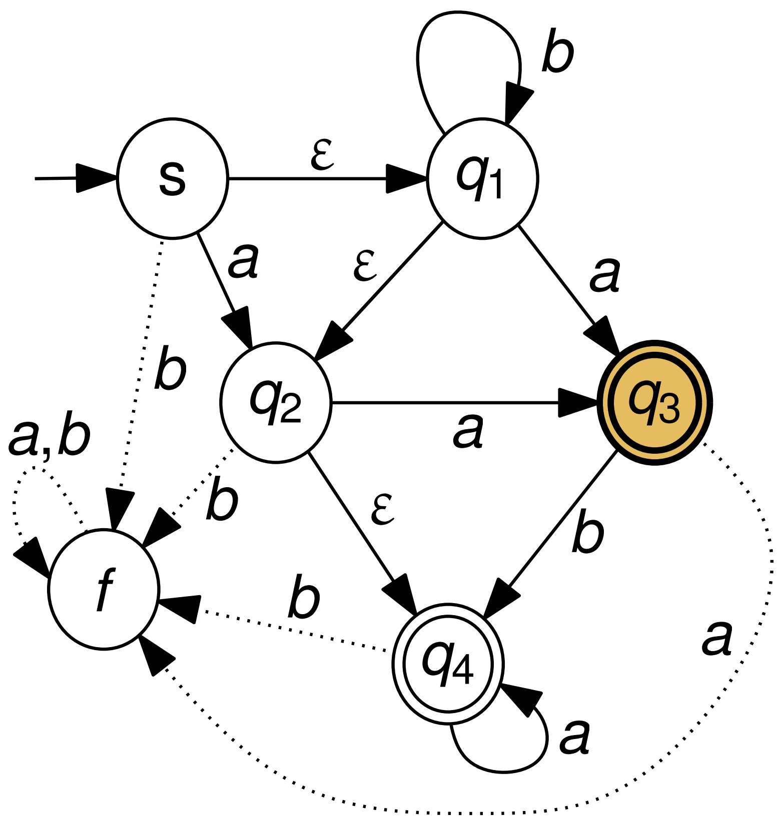
$$E(q_2) = \{q_2, q_4\},$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



$E(s) = \{s, q_1, q_2, q_4\}$,

$E(q_1) = \{q_1, q_2, q_4\}$,

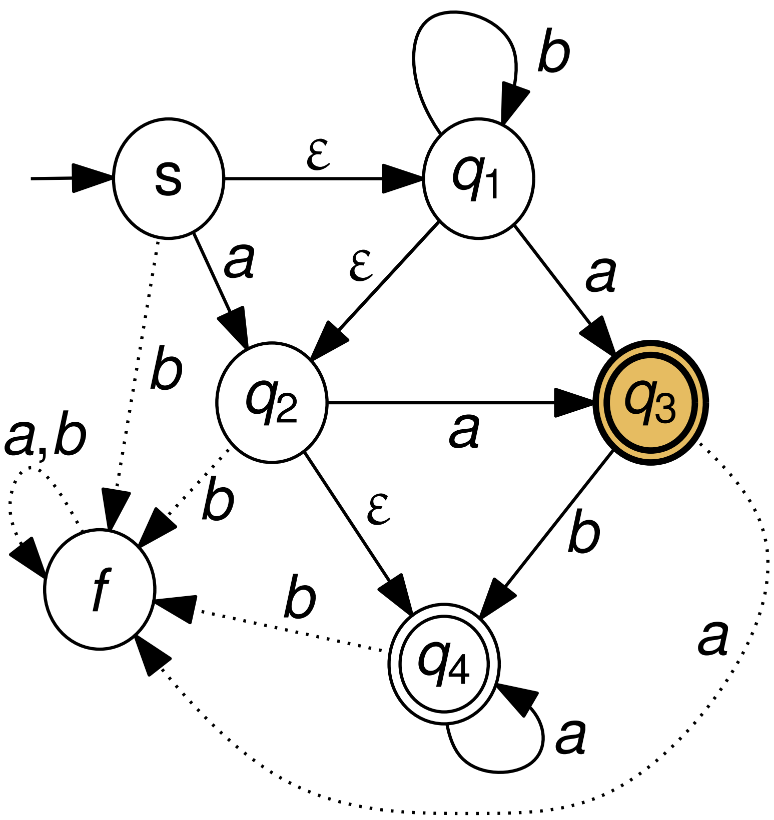
$E(q_2) = \{q_2, q_4\}$,

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



$$E(s) = \{s, q_1, q_2, q_4\},$$

$$E(q_1) = \{q_1, q_2, q_4\},$$

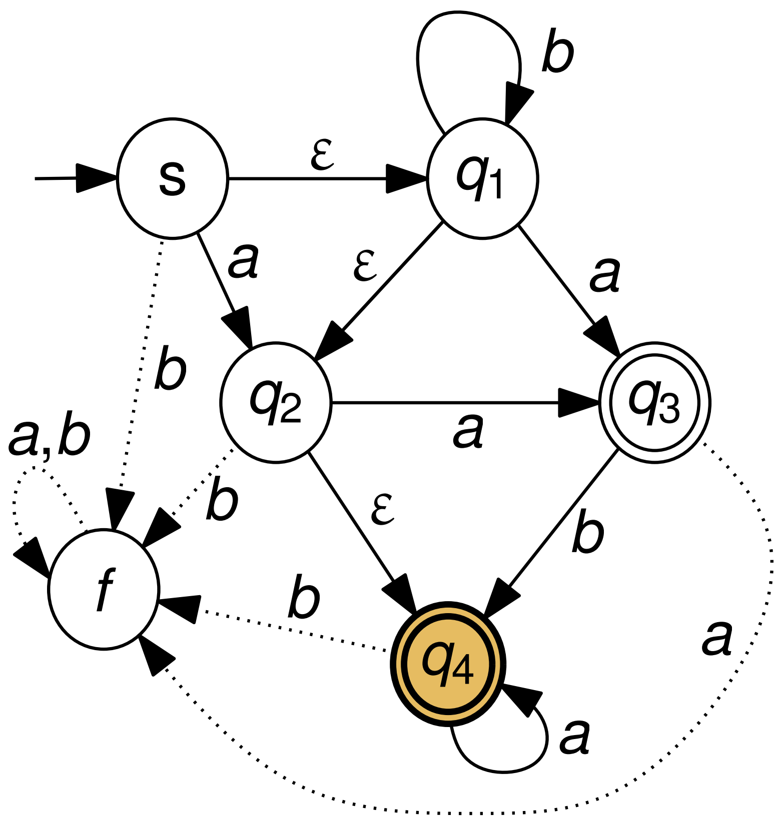
$$E(q_2) = \{q_2, q_4\},$$

$$E(q_3) = \{q_3\},$$

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



$$E(s) = \{s, q_1, q_2, q_4\},$$

$$E(q_1) = \{q_1, q_2, q_4\},$$

$$E(q_2) = \{q_2, q_4\},$$

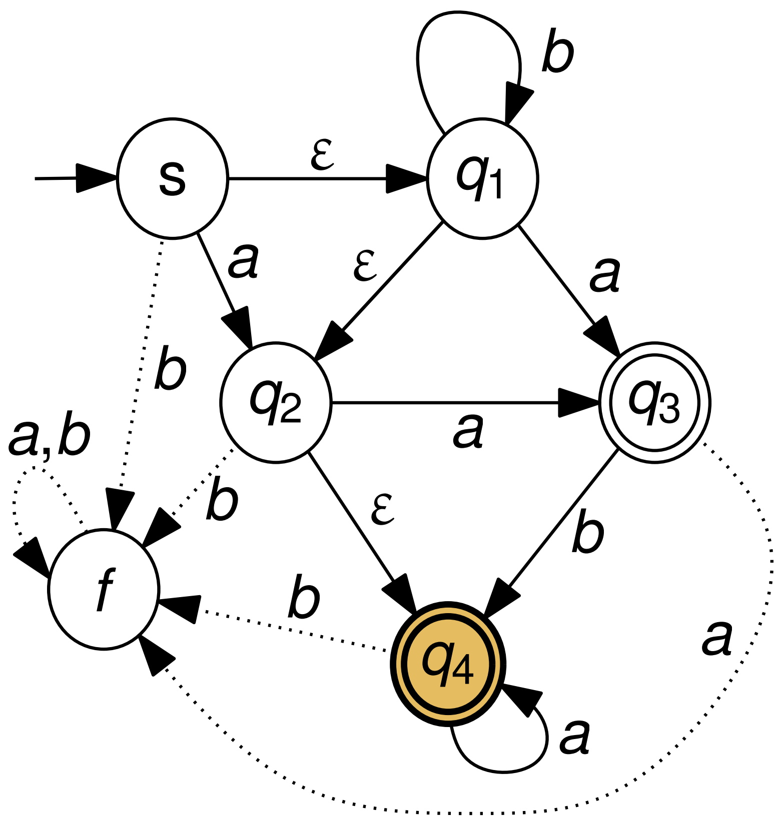
$$E(q_3) = \{q_3\},$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



$$E(s) = \{s, q_1, q_2, q_4\},$$

$$E(q_1) = \{q_1, q_2, q_4\},$$

$$E(q_2) = \{q_2, q_4\},$$

$$E(q_3) = \{q_3\},$$

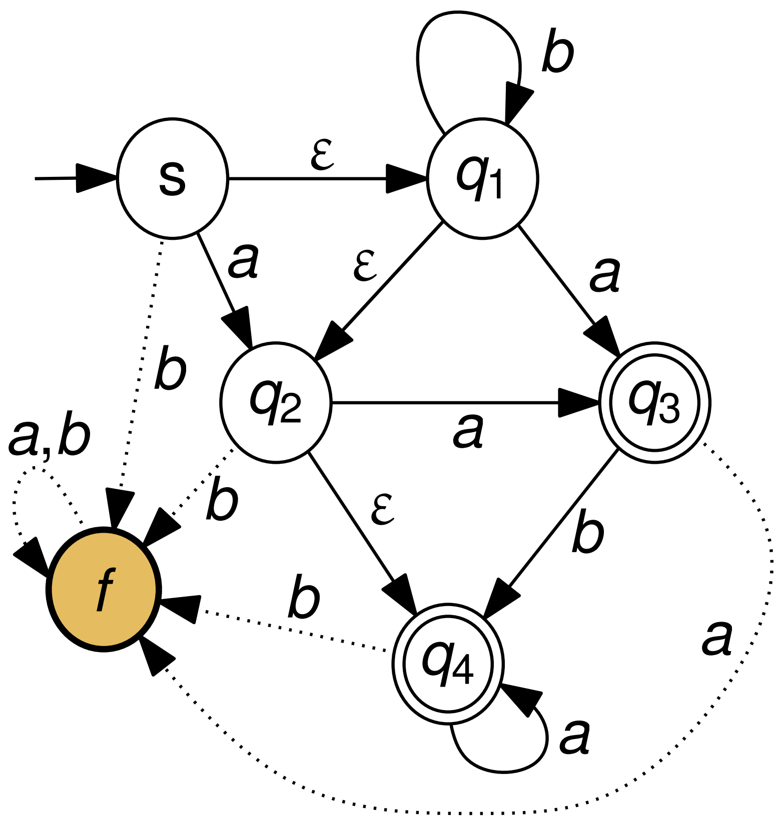
$$E(q_4) = \{q_4\},$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



$$E(s) = \{s, q_1, q_2, q_4\},$$

$$E(q_1) = \{q_1, q_2, q_4\},$$

$$E(q_2) = \{q_2, q_4\},$$

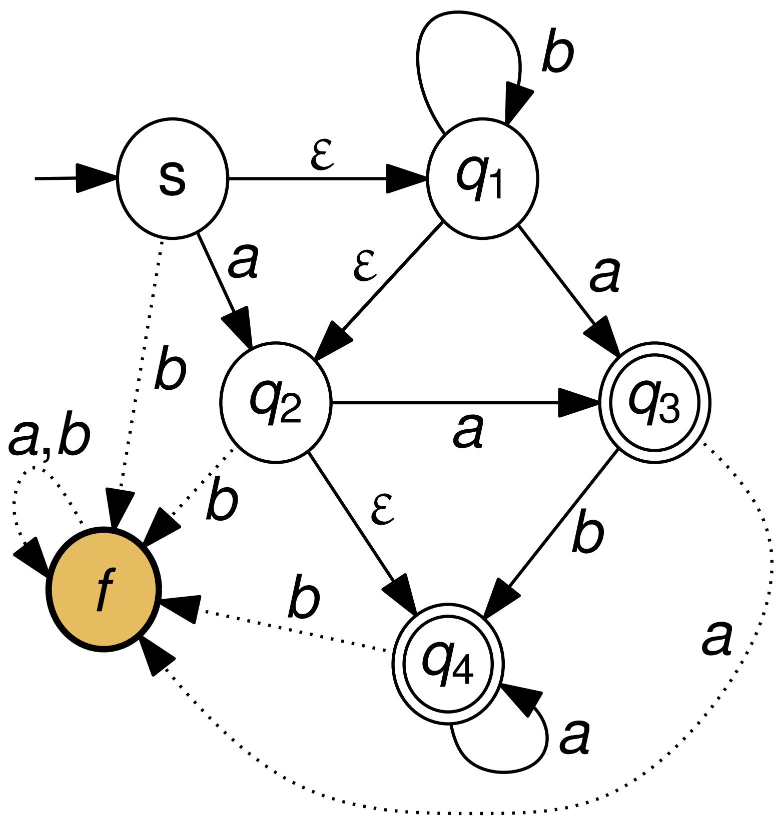
$$E(q_3) = \{q_3\},$$

$$E(q_4) = \{q_4\},$$

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(a) Geben Sie für jeden Zustand dessen ε-Abschluss an.



$$E(s) = \{s, q_1, q_2, q_4\},$$

$$E(q_1) = \{q_1, q_2, q_4\},$$

$$E(q_2) = \{q_2, q_4\},$$

$$E(q_3) = \{q_3\},$$

$$E(q_4) = \{q_4\},$$

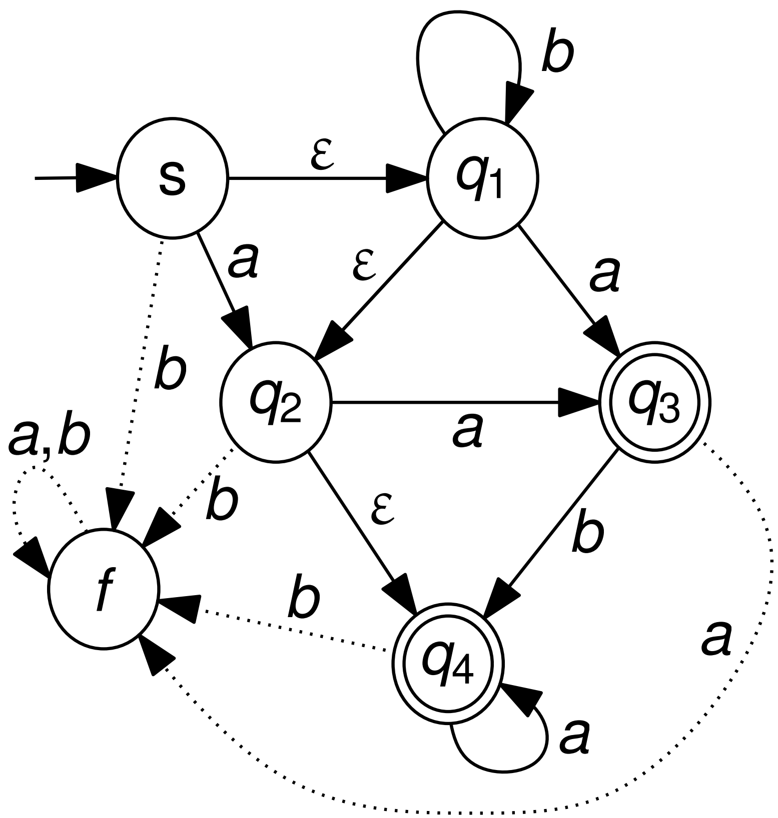
$$E(f) = \{f\}.$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(b) Geben Sie für $\mathcal{A}$ einen äquivalenten NEA an, der keine ε-Übergänge enthält.



$$E(s) = \{s, q_1, q_2, q_4\},$$

$$E(q_1) = \{q_1, q_2, q_4\},$$

$$E(q_2) = \{q_2, q_4\},$$

$$E(q_3) = \{q_3\},$$

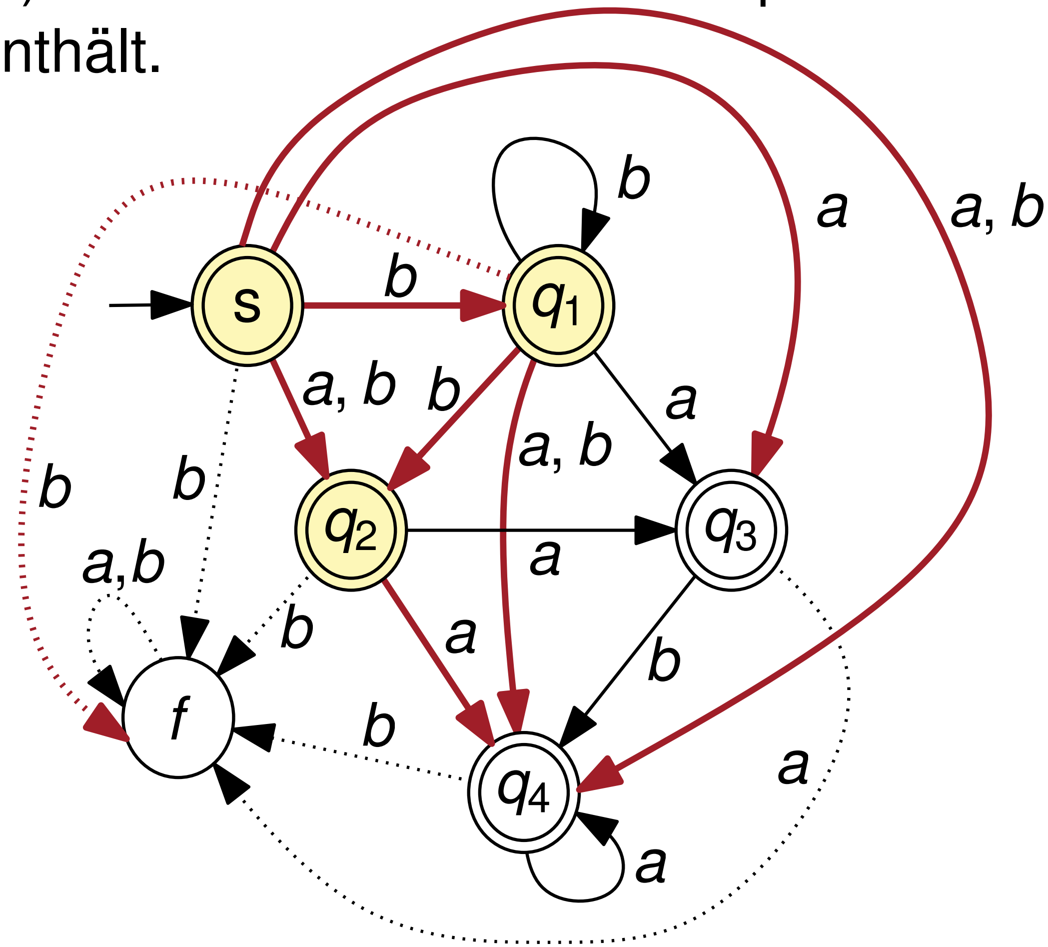
$$E(q_4) = \{q_4\},$$

$$E(f) = \{f\}.$$

NEA

Sei $\mathcal{A}$ der nichtdeterministische endliche Automat, der durch folgenden Zustandsgraphen gegeben ist:

(b) Geben Sie für $\mathcal{A}$ einen äquivalenten NEA an, der keine ε-Übergänge enthält.



$E(s) = \{s, q_1, q_2, q_4\}$,

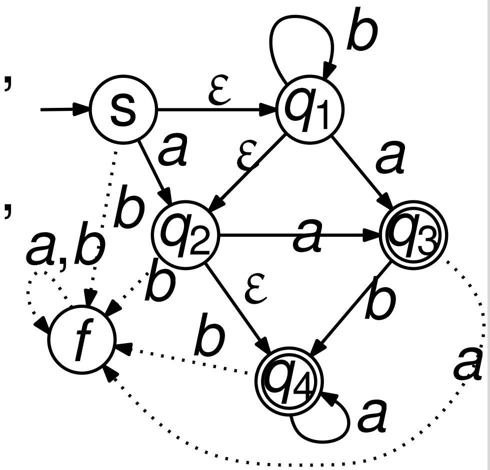
$E(q_1) = \{q_1, q_2, q_4\}$,

$E(q_2) = \{q_2, q_4\}$,

$E(q_3) = \{q_3\}$,

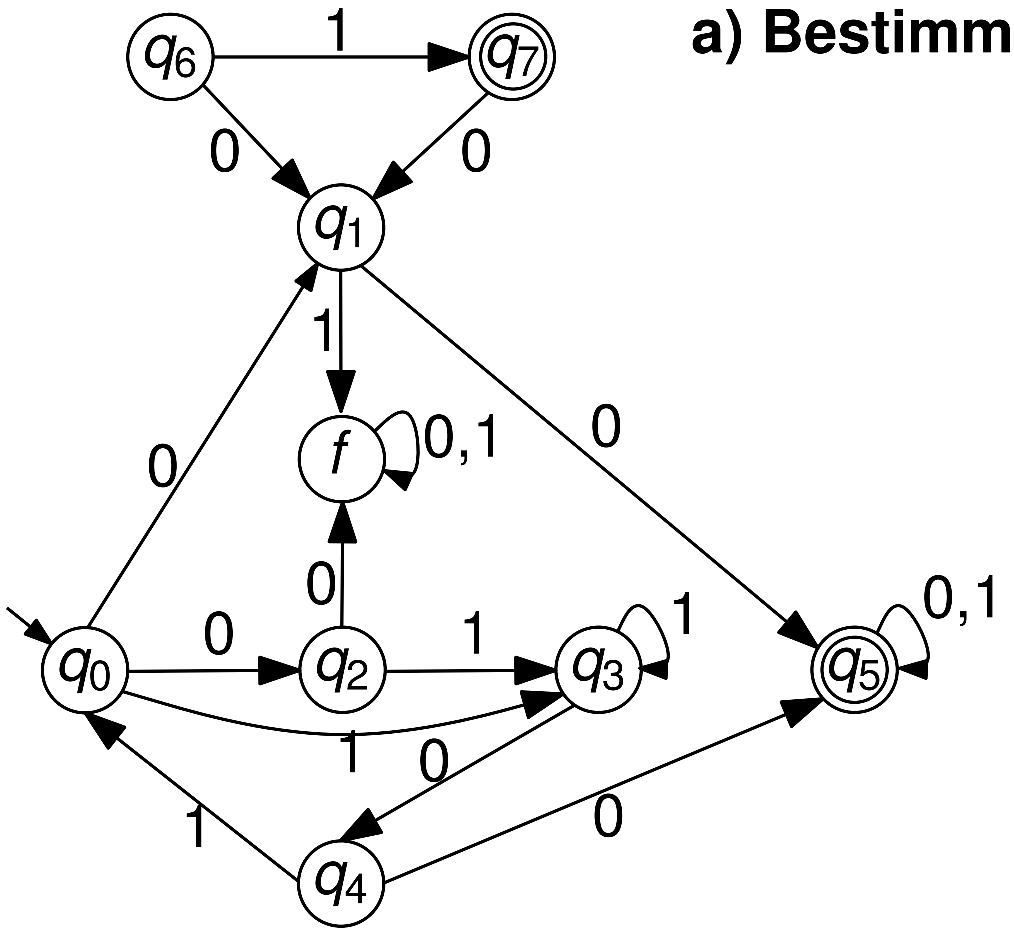
$E(q_4) = \{q_4\}$,

$E(f) = \{f\}$.



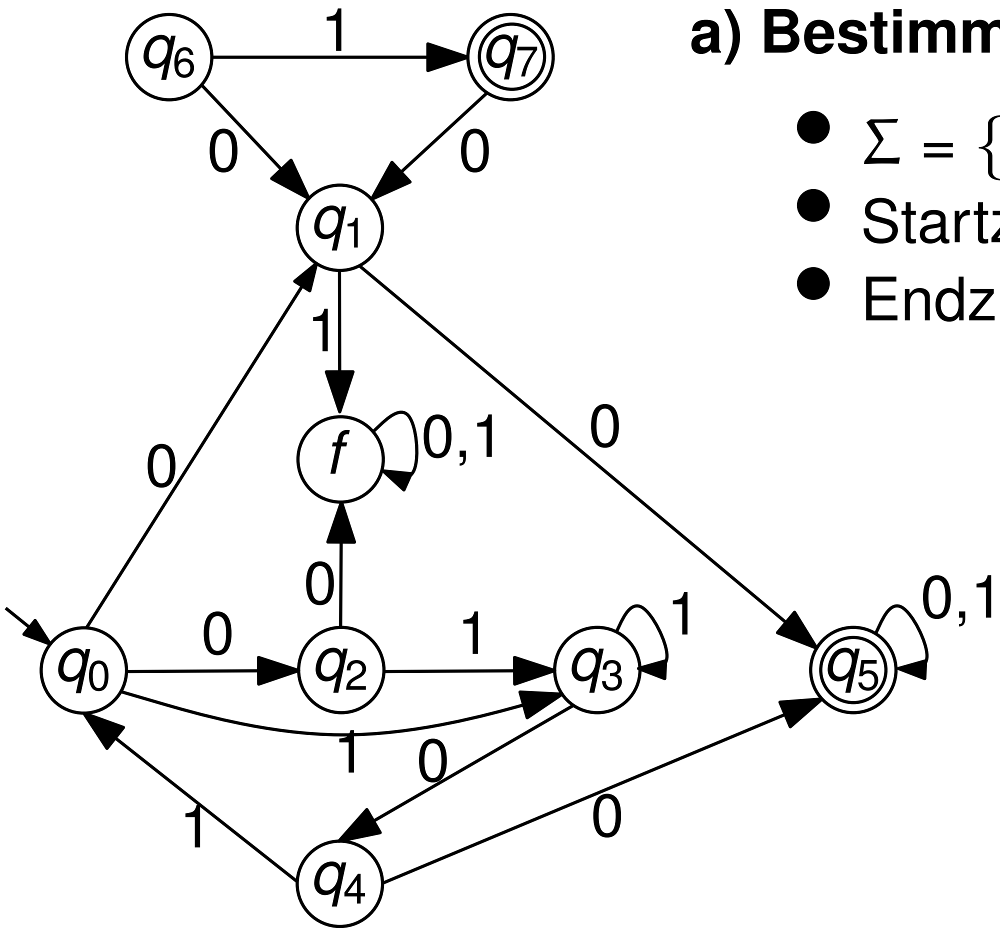
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Entfernen überflüssiger Zustände



a) Bestimme Σ, Start- und Endzustände.

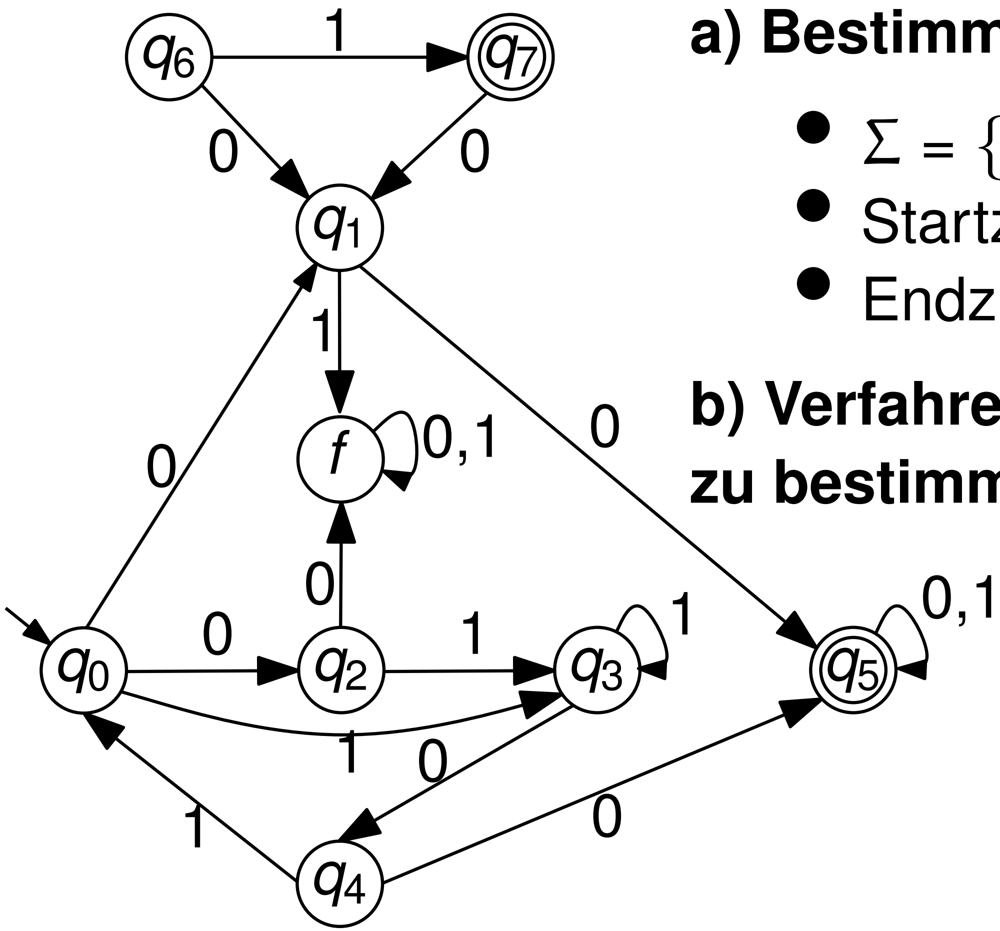
Entfernen überflüssiger Zustände



a) Bestimme Σ, Start- und Endzustände.

- $\Sigma = \{0, 1\}$
- Startzustand q_0
- Endzustände $F = \{q_5, q_7\}$

Entfernen überflüssiger Zustände

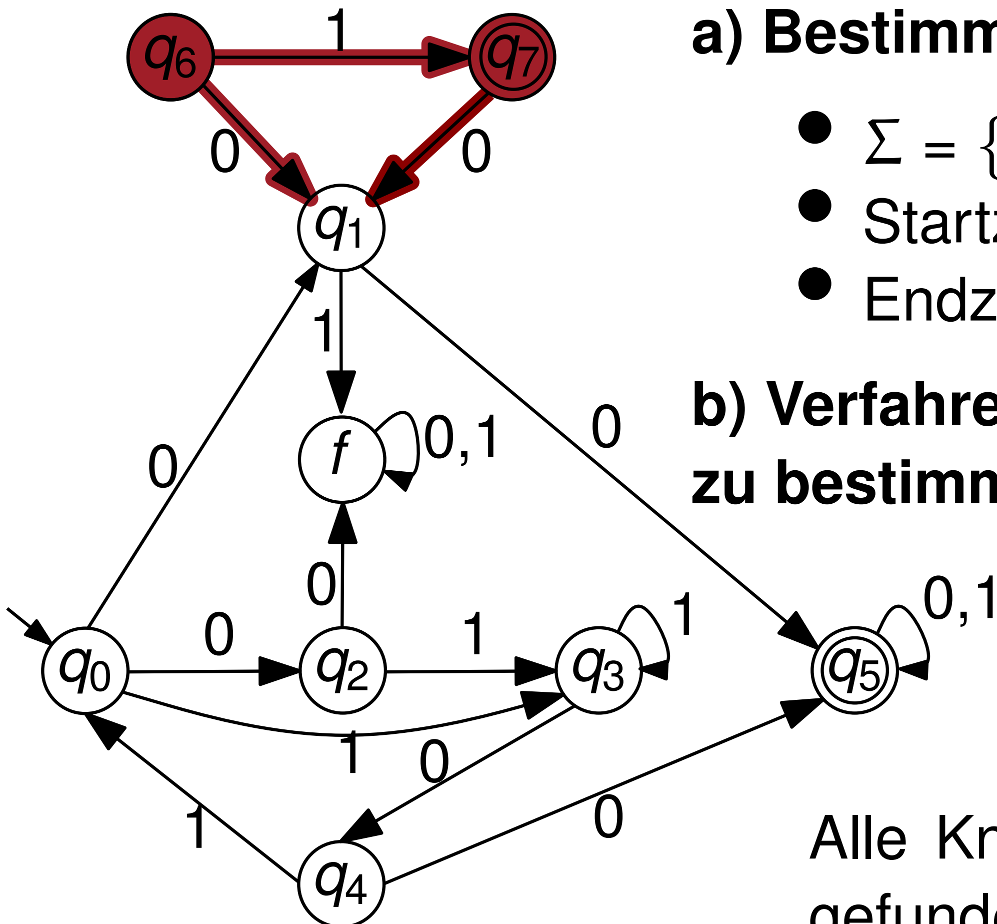


a) Bestimme Σ, Start- und Endzustände.

- $\Sigma = \{0, 1\}$
- Startzustand q_0
- Endzustände $F = \{q_5, q_7\}$

b) Verfahren um nicht erreichbare Zustände zu bestimmen?

Entfernen überflüssiger Zustände



a) Bestimme Σ, Start- und Endzustände.

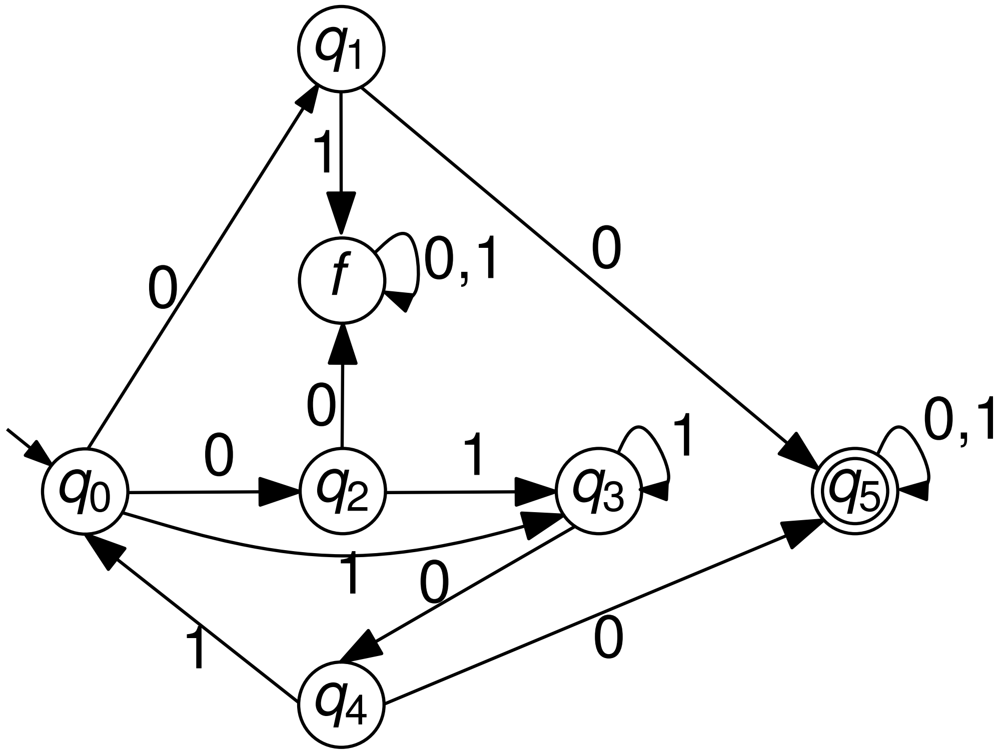
- $\Sigma = \{0, 1\}$
- Startzustand q_0
- Endzustände $F = \{q_5, q_7\}$

b) Verfahren um nicht erreichbare Zustände zu bestimmen?

Betrachte Zustandsgraph $G = (V, E)$
Wende Tiefen- oder Breitensuche von q_0 aus an.

Alle Knoten, die während der Suche nicht gefunden werden, sind unerreichbar.

Potenzmengenkonstruktion

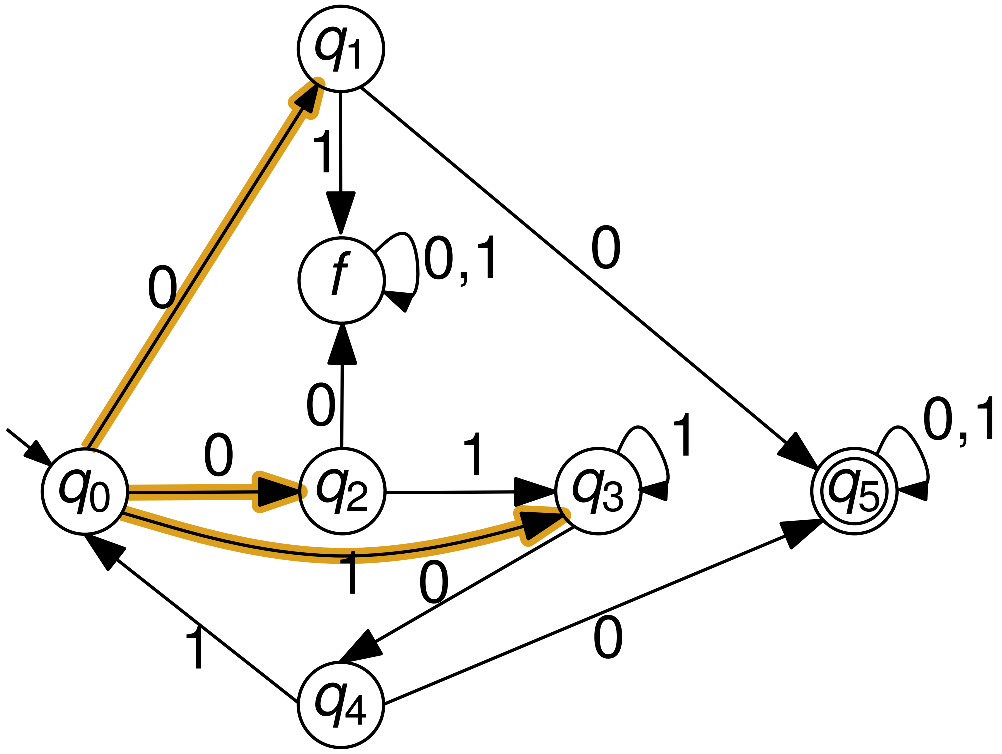


1. Schritt: NEA $\rightarrow$ DEA

Zustand	Übergänge 0	1
$\{q_0\}$		

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

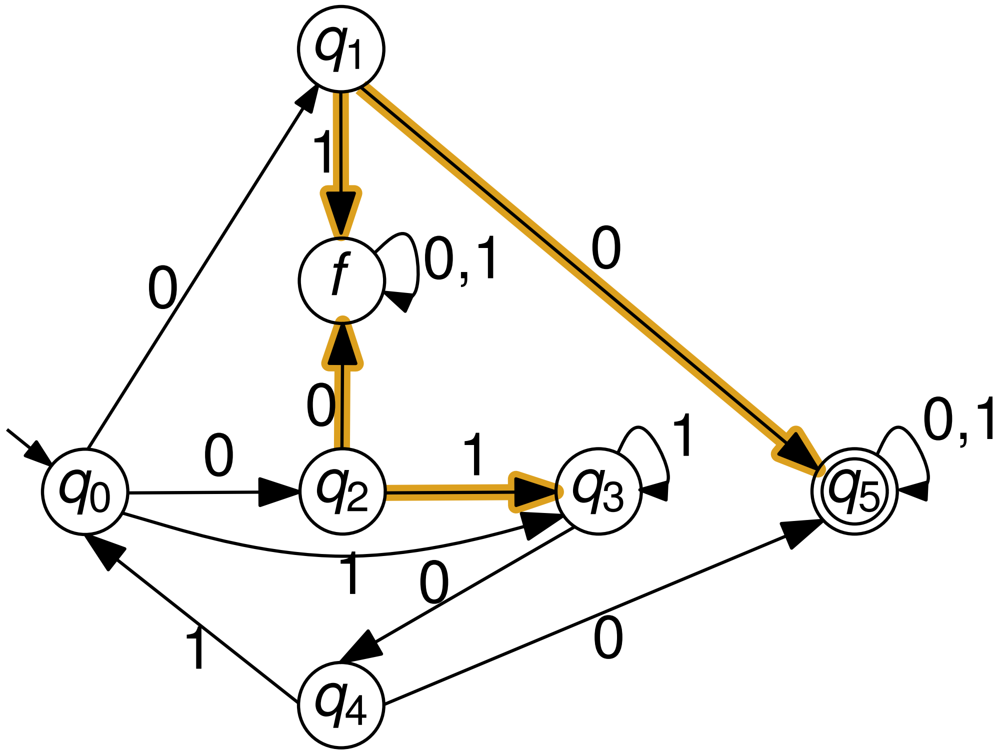
Potenzmengenkonstruktion



1. Schritt: NEA $\rightarrow$ DEA

Zustand	Übergänge 0	1
$\{q_0\}$	$\{q_1, q_2\}$	$\{q_3\}$
$\{q_1, q_2\}$		
$\{q_3\}$		

Potenzmengenkonstruktion

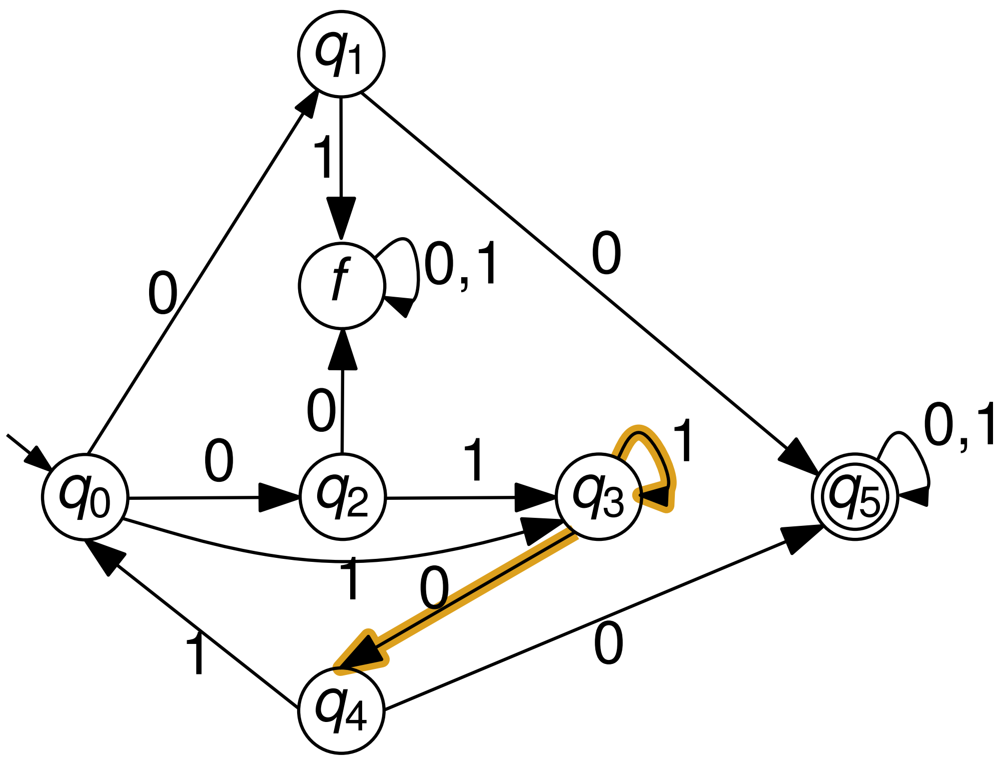


1. Schritt: NEA $\rightarrow$ DEA

Zustand	Übergänge 0	Übergänge 1
$\{q_0\}$	$\{q_1, q_2\}$	$\{q_3\}$
$\{q_1, q_2\}$	$\{q_5, f\}$	$\{q_3, f\}$
$\{q_3\}$		
$\{q_5, f\}$		
$\{q_3, f\}$		

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Potenzmengenkonstruktion

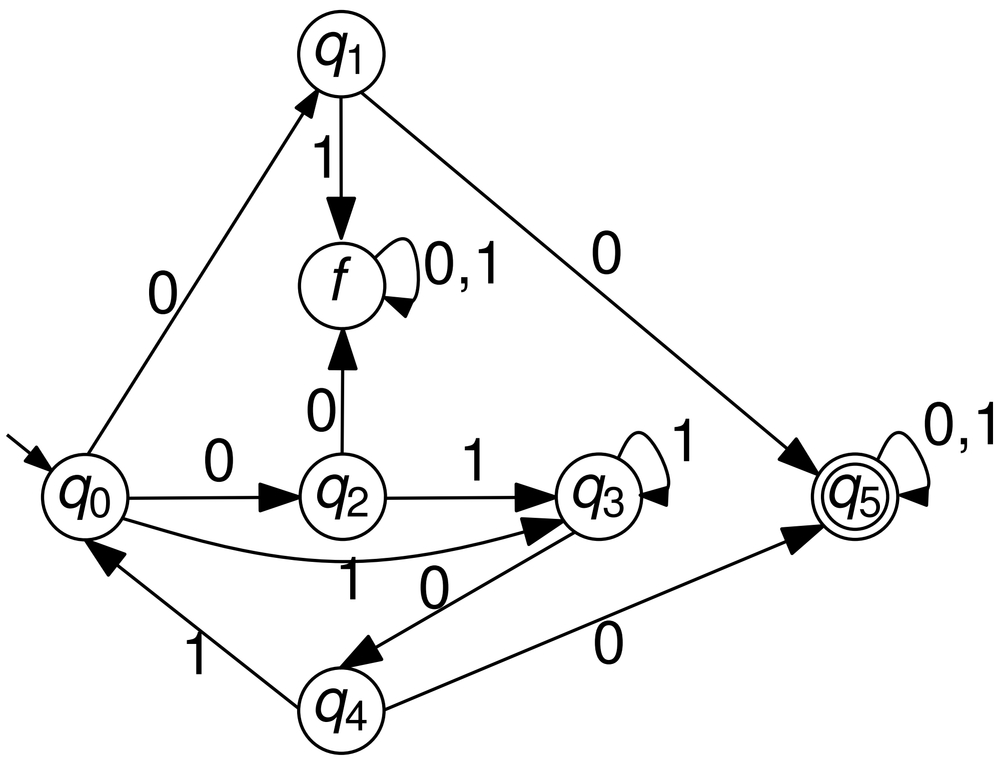


1. Schritt: NEA $\rightarrow$ DEA

Zustand	Übergänge 0	Übergänge 1
$\{q_0\}$	$\{q_1, q_2\}$	$\{q_3\}$
$\{q_1, q_2\}$	$\{q_5, f\}$	$\{q_3, f\}$
$\{q_3\}$	$\{q_4\}$	$\{q_3\}$
$\{q_5, f\}$		
$\{q_3, f\}$		
$\{q_4\}$		

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

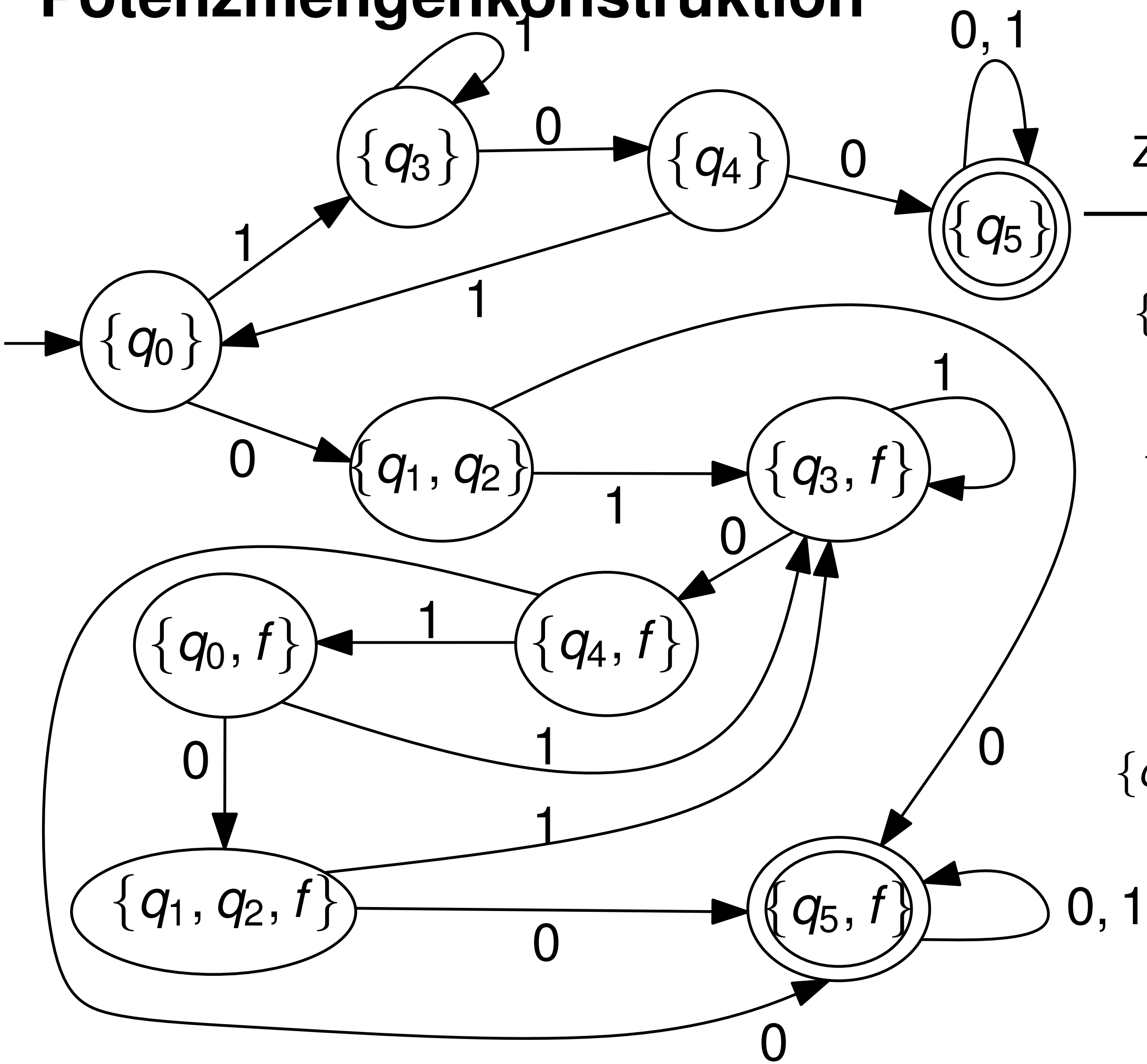
Potenzmengenkonstruktion



1. Schritt: NEA $\rightarrow$ DEA

Zustand	Übergänge 0	Übergänge 1
$\{q_0\}$	$\{q_1, q_2\}$	$\{q_3\}$
$\{q_1, q_2\}$	$\{q_5, f\}$	$\{q_3, f\}$
$\{q_3\}$	$\{q_4\}$	$\{q_3\}$
$\{q_5, f\}$	$\{q_5, f\}$	$\{q_5, f\}$
$\{q_3, f\}$	$\{q_4, f\}$	$\{q_3, f\}$
$\{q_4\}$	$\{q_5\}$	$\{q_0\}$
$\{q_4, f\}$	$\{q_5, f\}$	$\{q_0, f\}$
$\{q_5\}$	$\{q_5\}$	$\{q_5\}$
$\{q_0, f\}$	$\{q_1, q_2, f\}$	$\{q_3, f\}$
$\{q_1, q_2, f\}$	$\{q_5, f\}$	$\{q_3, f\}$

Potenzmengenkonstruktion



Zustand	Übergänge 0	Übergänge 1
$\{q_0\}$	$\{q_1, q_2\}$	$\{q_3\}$
$\{q_1, q_2\}$	$\{q_5, f\}$	$\{q_3, f\}$
$\{q_3\}$	$\{q_4\}$	$\{q_3\}$
$\{q_5, f\}$	$\{q_5, f\}$	$\{q_5, f\}$
$\{q_3, f\}$	$\{q_4, f\}$	$\{q_3, f\}$
$\{q_4\}$	$\{q_5\}$	$\{q_0\}$
$\{q_4, f\}$	$\{q_5, f\}$	$\{q_0, f\}$
$\{q_5\}$	$\{q_5\}$	$\{q_5\}$
$\{q_0, f\}$	$\{q_1, q_2, f\}$	$\{q_3, f\}$
$\{q_1, q_2, f\}$	$\{q_5, f\}$	$\{q_3, f\}$

Potenzmengenkonstruktion

Mit der Potenzmengenkonstruktion lässt sich aus jedem NEA ein äquivalenter DEA konstruieren.

NEAs und DEAs erkennen also dieselben Sprachen und sind deshalb **gleich mächtig**.

Es gibt NEAs, deren äquivalenter DEA wesentlich mehr Zustände hat. Das spielt für unsere Auffassung von Mächtigkeit aber keine Rolle!

Minimierung von Automaten

Zwei Zustände p und q eines deterministischen endlichen Automaten heißen *äquivalent* ($p \equiv q$), wenn für alle Wörter $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \quad \Longleftrightarrow \quad \delta(q, w) \in F.$$

Mit $[p]$ bezeichnen wir die Äquivalenzklasse der zu p äquivalenten Zustände.

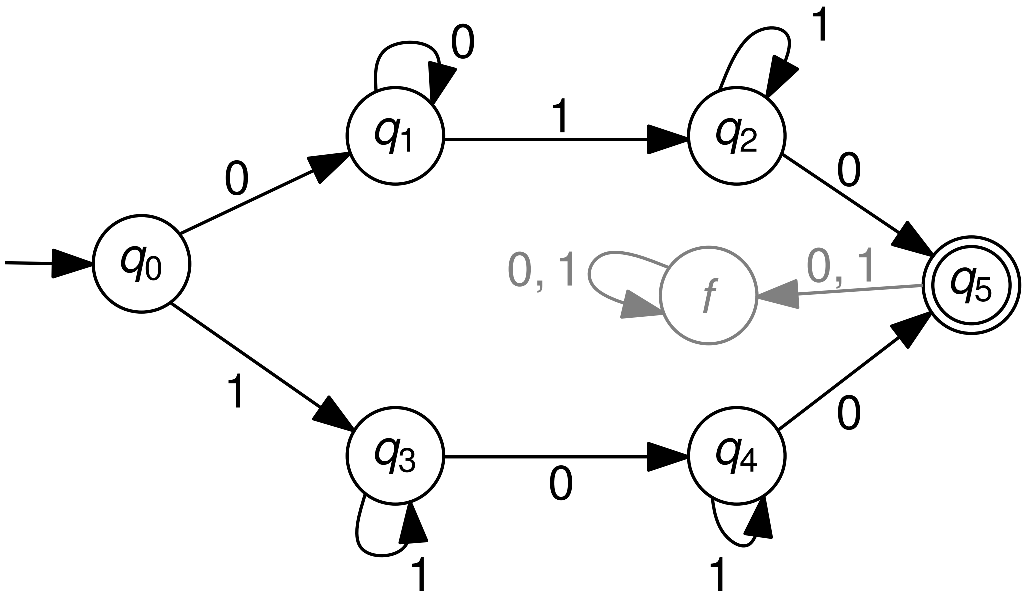
- Zwei Zustände haben dasselbe Akzeptanzverhalten, wenn es für das Erreichen eines Endzustandes durch Abarbeiten eines Wortes w unerheblich ist, aus welchem der beiden Zustände wir starten.
- Reduktion der Anzahl der Zustände durch Zusammlegen der Zustände mit gleichem Akzeptanzverhalten.

Minimierung von Automaten

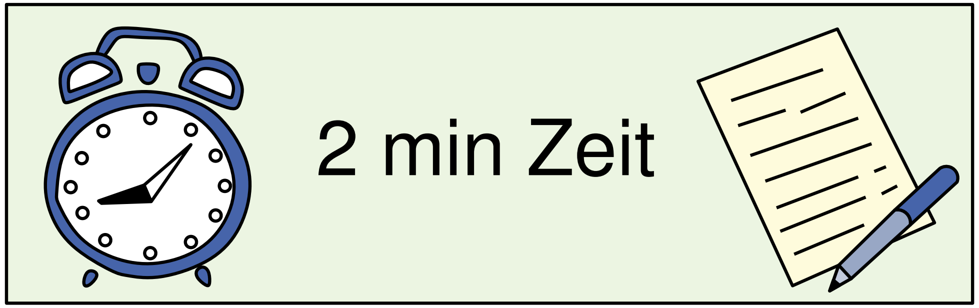
Zwei Zustände p und q eines deterministischen endlichen Automaten heißen *äquivalent* ($p \equiv q$), wenn für alle Wörter $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \quad \Longleftrightarrow \quad \delta(q, w) \in F.$$

Mit $[p]$ bezeichnen wir die Äquivalenzklasse der zu p äquivalenten Zustände.



Welche Zustände sind äquivalent?



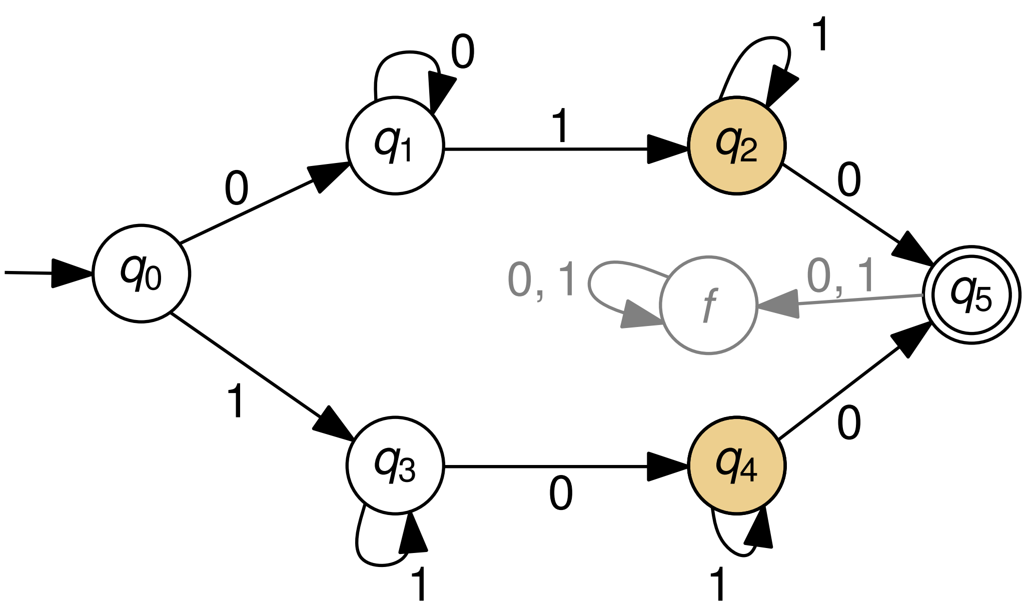
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Minimierung von Automaten

Zwei Zustände p und q eines deterministischen endlichen Automaten heißen *äquivalent* ($p \equiv q$), wenn für alle Wörter $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \quad \Longleftrightarrow \quad \delta(q, w) \in F.$$

Mit $[p]$ bezeichnen wir die Äquivalenzklasse der zu p äquivalenten Zustände.



Minimierung von Automaten

Zwei Zustände p und q eines deterministischen endlichen Automaten heißen *äquivalent* ($p \equiv q$), wenn für alle Wörter $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \quad \Longleftrightarrow \quad \delta(q, w) \in F.$$

Mit $[p]$ bezeichnen wir die Äquivalenzklasse der zu p äquivalenten Zustände.

Zu einem DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $\mathcal{A}^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch:

- $Q^{\equiv} := \{[q]\} \quad q \in Q$
- $\Sigma^{\equiv} := \Sigma$
- $\delta^{\equiv}([q], a) := [\delta(q, a)]$
- $s^{\equiv} := [s]$
- $F^{\equiv} := \{[f]\} \quad f \in F$

Minimierung von Automaten

Zwei Zustände p und q eines deterministischen endlichen Automaten heißen *äquivalent* ($p \equiv q$), wenn für alle Wörter $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \quad \Longleftrightarrow \quad \delta(q, w) \in F.$$

Mit $[p]$ bezeichnen wir die Äquivalenzklasse der zu p äquivalenten Zustände.

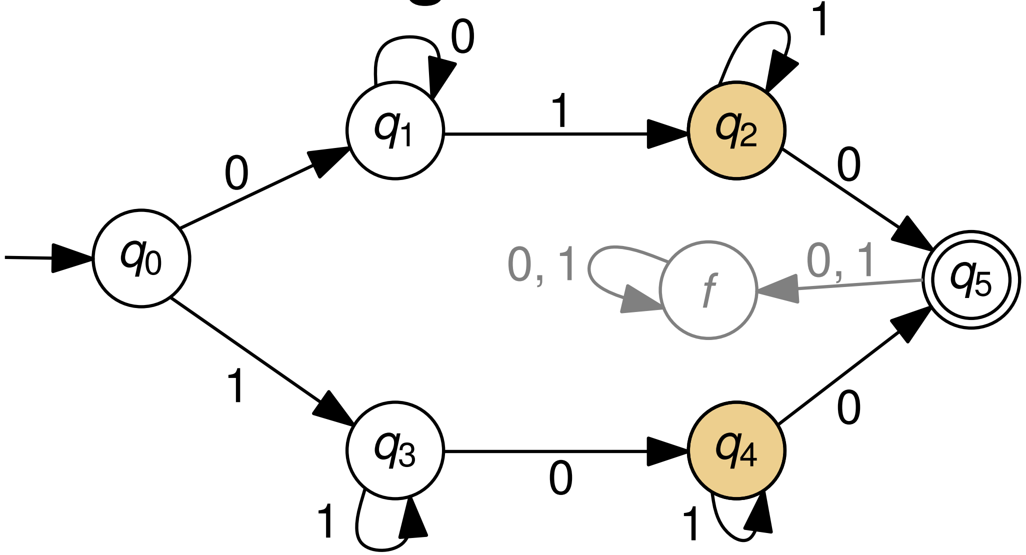
Zu einem DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $\mathcal{A}^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch

- $Q^{\equiv} := \{[q]\} \quad q \in Q$
- $\Sigma^{\equiv} := \Sigma$
- $\delta^{\equiv}([q], a) := [\delta(q, a)]$
- $s^{\equiv} := [s]$
- $F^{\equiv} := \{[f]\} \quad f \in F$

In der Vorlesung:

- $\mathcal{A}^{\equiv}$ akzeptiert dieselbe Sprache wie $\mathcal{A}$.
- $\mathcal{A}^{\equiv}$ ist minimal (nächste VL).

Minimierung von Automaten



Wie sieht der Äquivalenzklassenautomat aus?

2 min Zeit

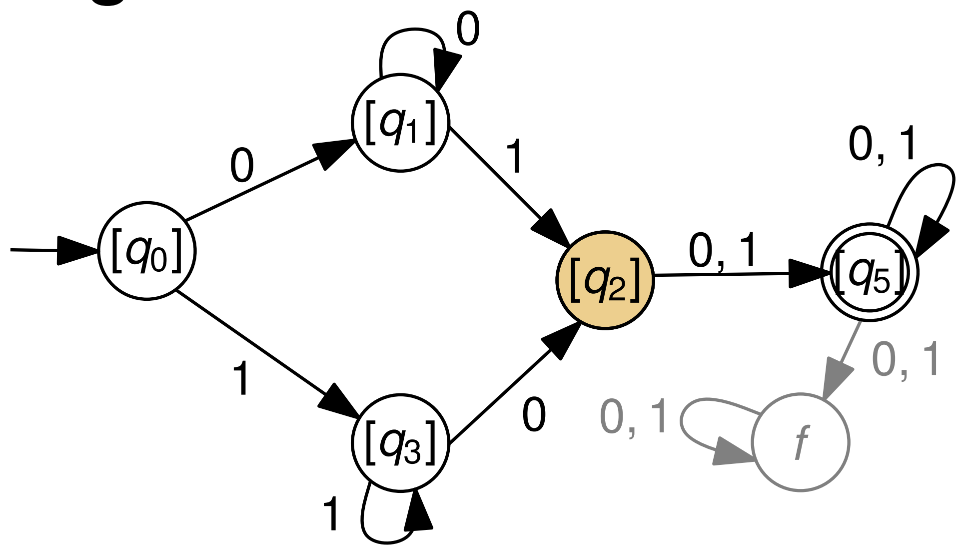
Zu einem DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $\mathcal{A}^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durc

- $Q^{\equiv} := \{[q]\} \quad q \in Q$
- $\Sigma^{\equiv} := \Sigma$
- $\delta^{\equiv}([q], a) := [\delta(q, a)]$
- $s^{\equiv} := [s]$
- $F^{\equiv} := \{[f]\} \quad f \in F$

In der Vorlesung:

- $\mathcal{A}^{\equiv}$ akzeptiert dieselbe Sprache wie $\mathcal{A}$.
- $\mathcal{A}^{\equiv}$ ist minimal (nächste VL).

Minimierung von Automaten



Zu einem DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $\mathcal{A}^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durch

- $Q^{\equiv} := \{[q]\} \quad q \in Q$
- $\Sigma^{\equiv} := \Sigma$
- $\delta^{\equiv}([q], a) := [\delta(q, a)]$
- $s^{\equiv} := [s]$
- $F^{\equiv} := \{[f]\} \quad f \in F$

In der Vorlesung:

- $\mathcal{A}^{\equiv}$ akzeptiert dieselbe Sprache wie $\mathcal{A}$.
- $\mathcal{A}^{\equiv}$ ist minimal (nächste VL).

Minimierung von Automaten

Zwei Zustände p und q eines deterministischen endlichen Automaten heißen *äquivalent* ($p \equiv q$), wenn für alle Wörter $w \in \Sigma^*$ gilt:

$$\delta(p, w) \in F \quad \Longleftrightarrow \quad \delta(q, w) \in F.$$

Mit $[p]$ bezeichnen wir die Äquivalenzklasse der zu p äquivalenten Zustände.

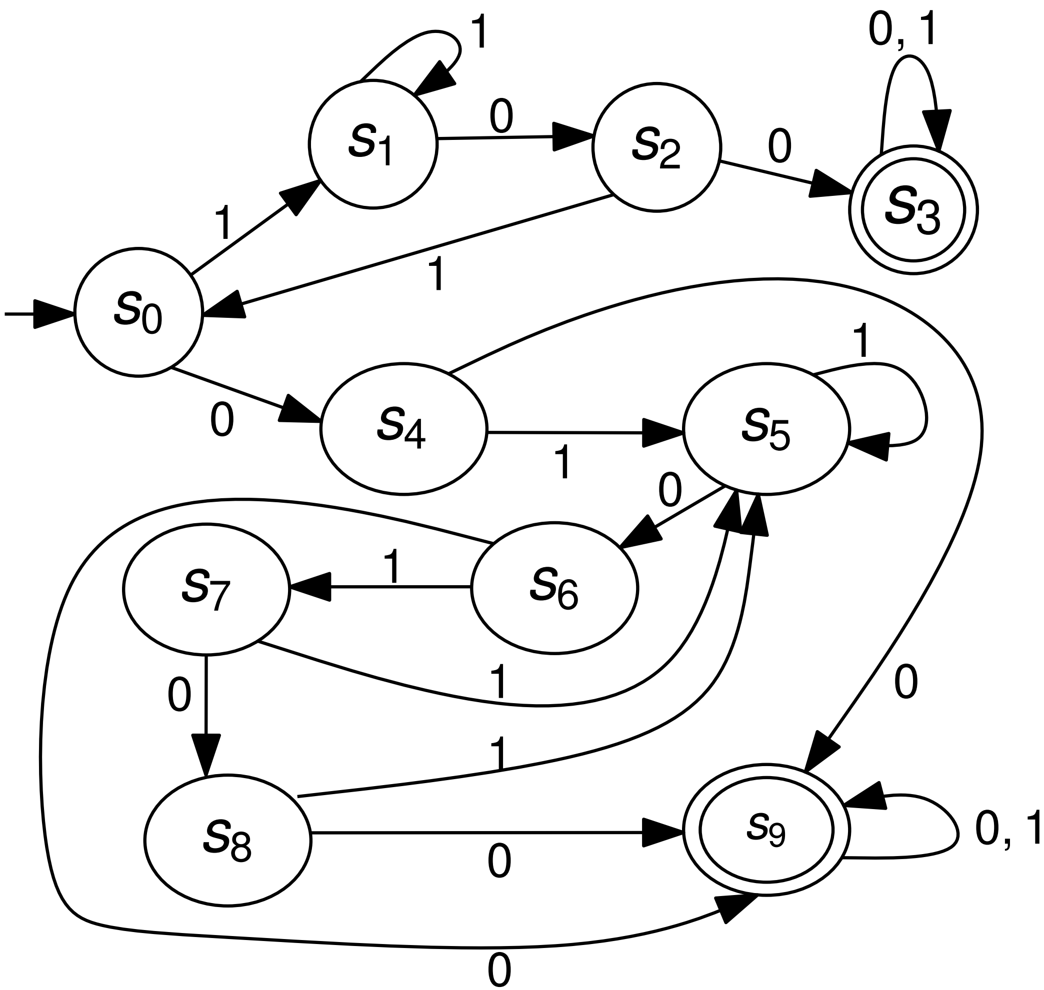
Zu einem DEA $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ definieren wir den Äquivalenzklassenautomaten $\mathcal{A}^{\equiv} = (Q^{\equiv}, \Sigma^{\equiv}, \delta^{\equiv}, s^{\equiv}, F^{\equiv})$ durc

- $Q^{\equiv} := \{[q]\} \quad q \in Q$
- $\Sigma^{\equiv} := \Sigma$
- $\delta^{\equiv}([q], a) := [\delta(q, a)]$
- $s^{\equiv} := [s]$
- $F^{\equiv} := \{[f]\} \quad f \in F$

Verfahren:
Finde systematisch **Zeugen** w für Zustände p und q, die nicht äquivalent sind:

$$\left|\{\delta(p, w), \delta(q, w)\} \cap F\right| = 1$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

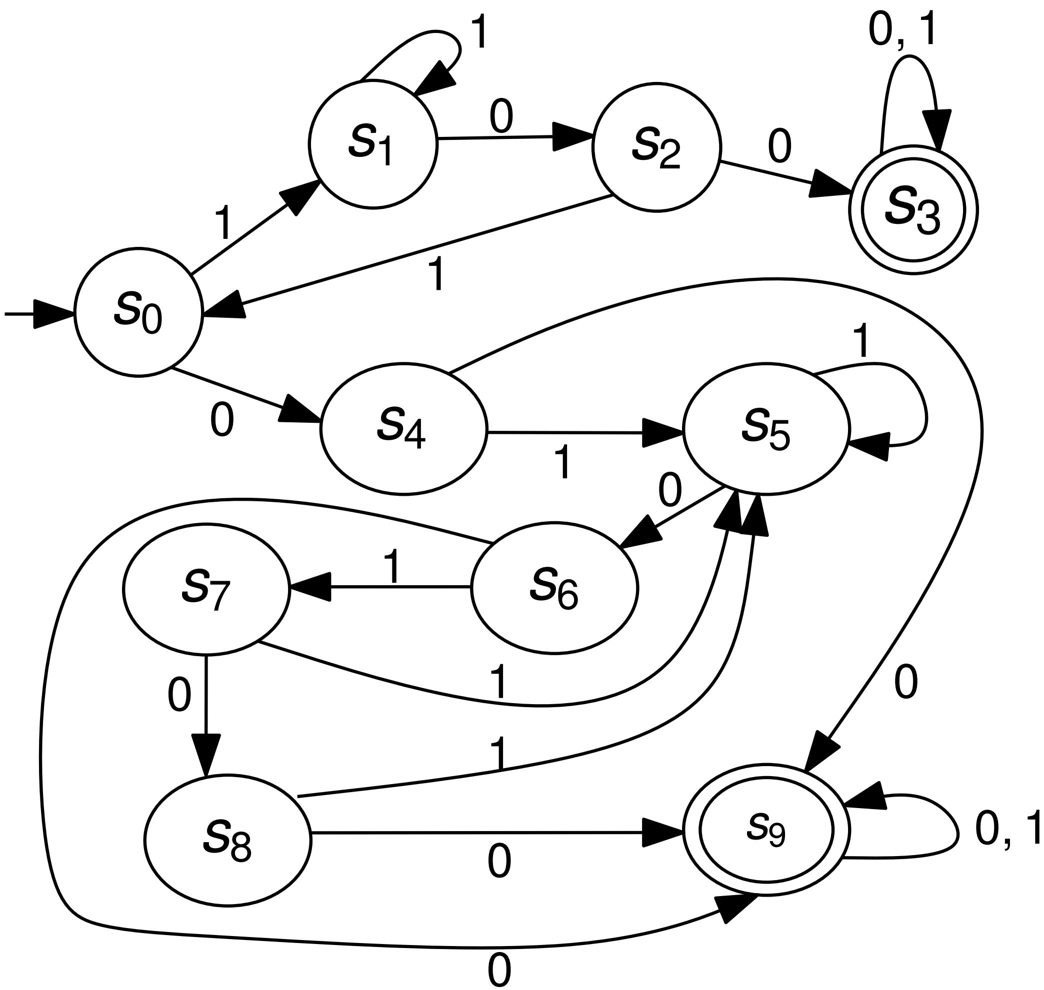


$s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9$

Betrachte Wort w =

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.



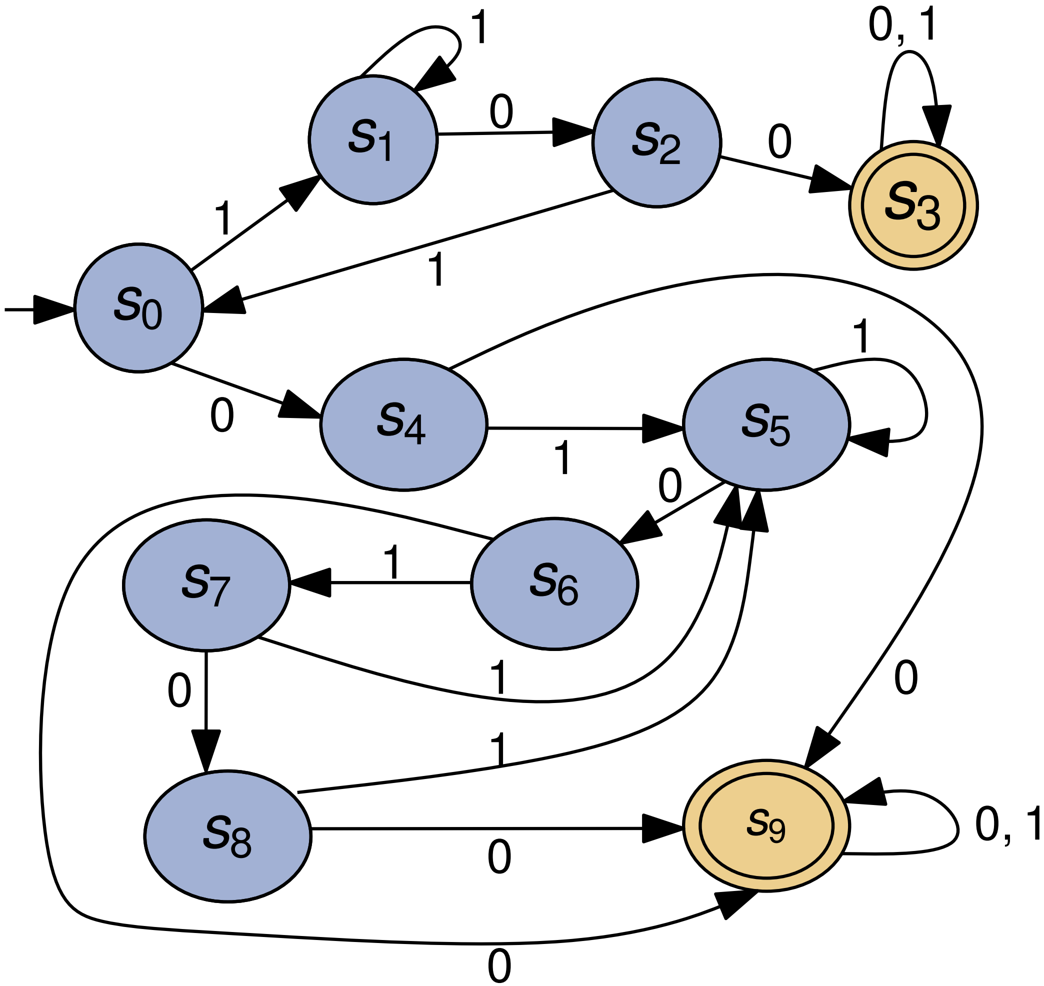
$s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9$

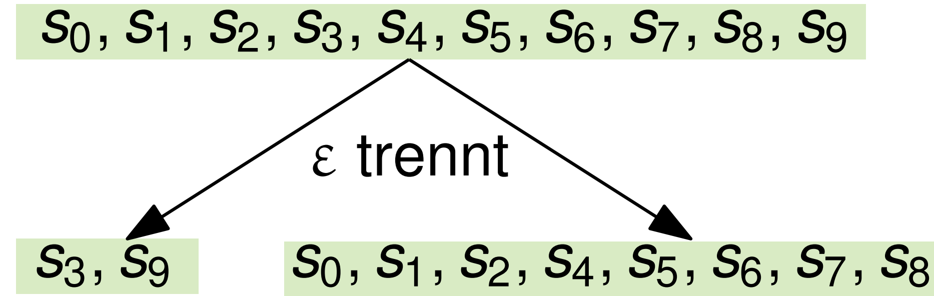
ε trennt

Betrachte Wort $w = \varepsilon$

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.



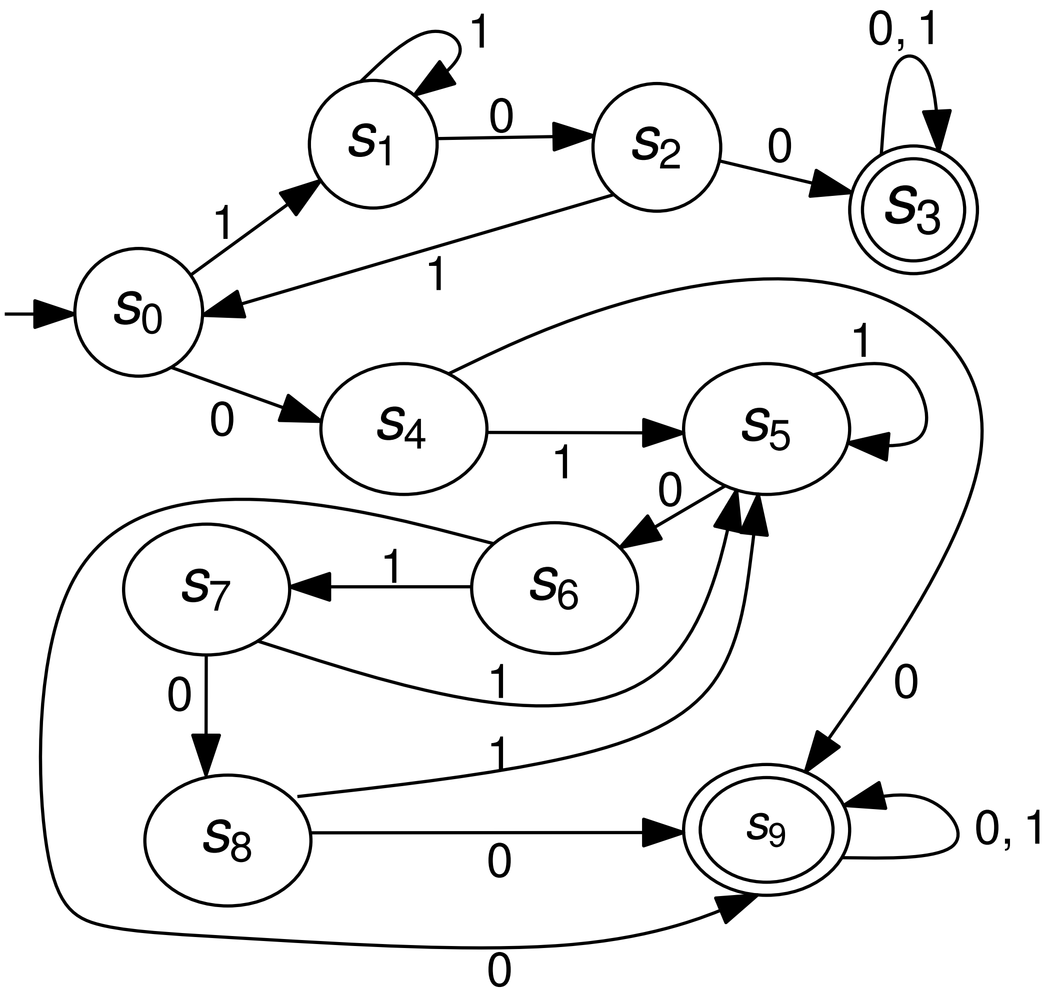


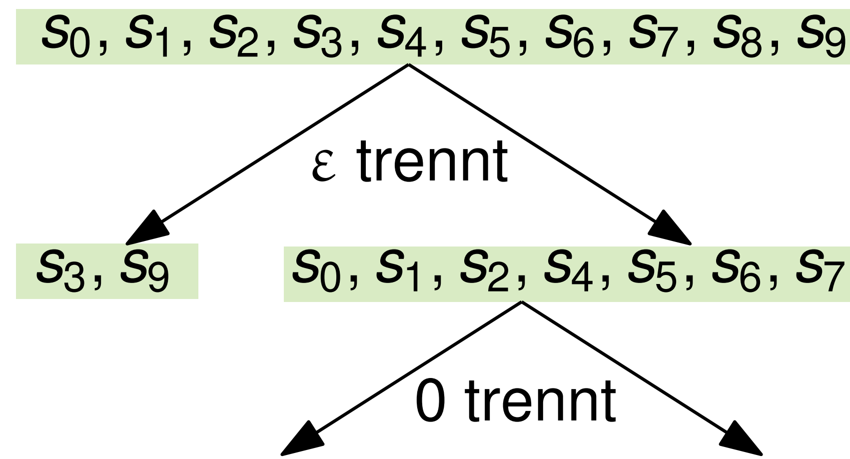
Betrachte Wort $w = \varepsilon$

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik



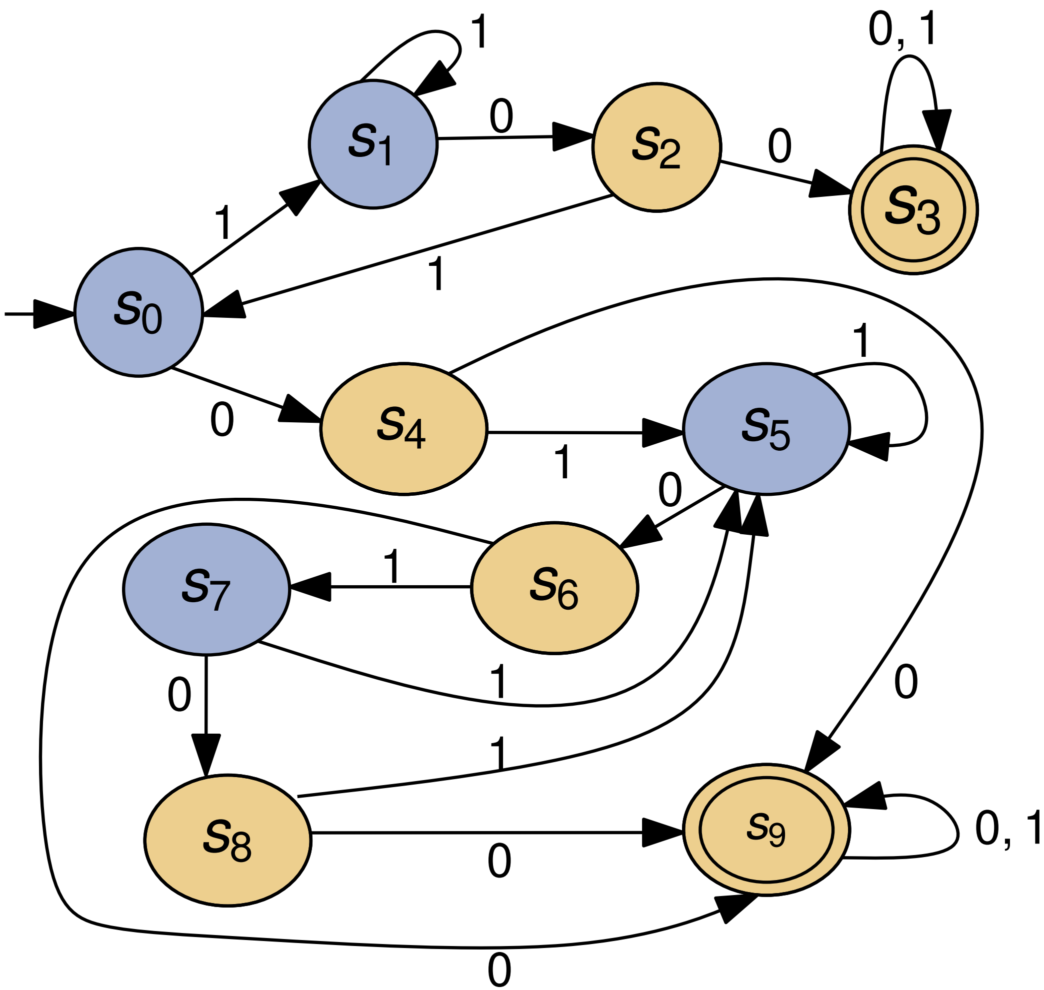


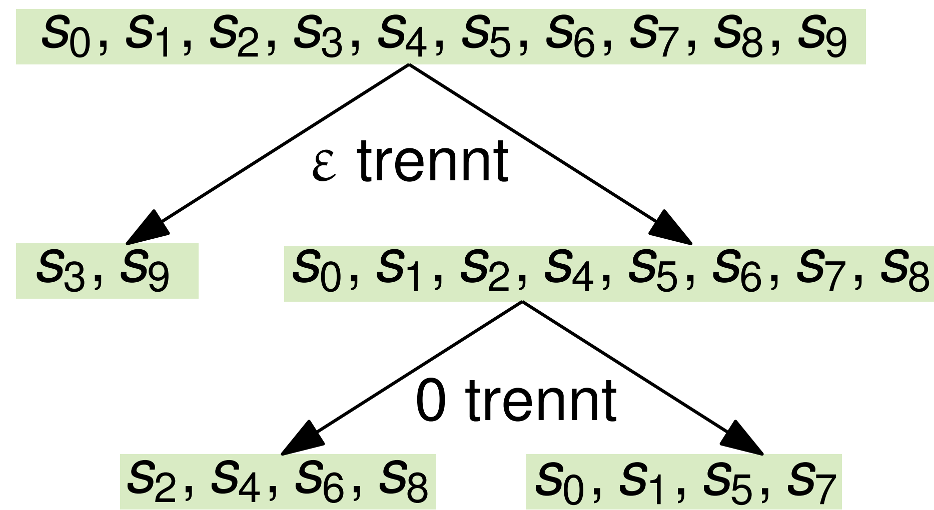
Betrachte Wort $w =$ 0

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

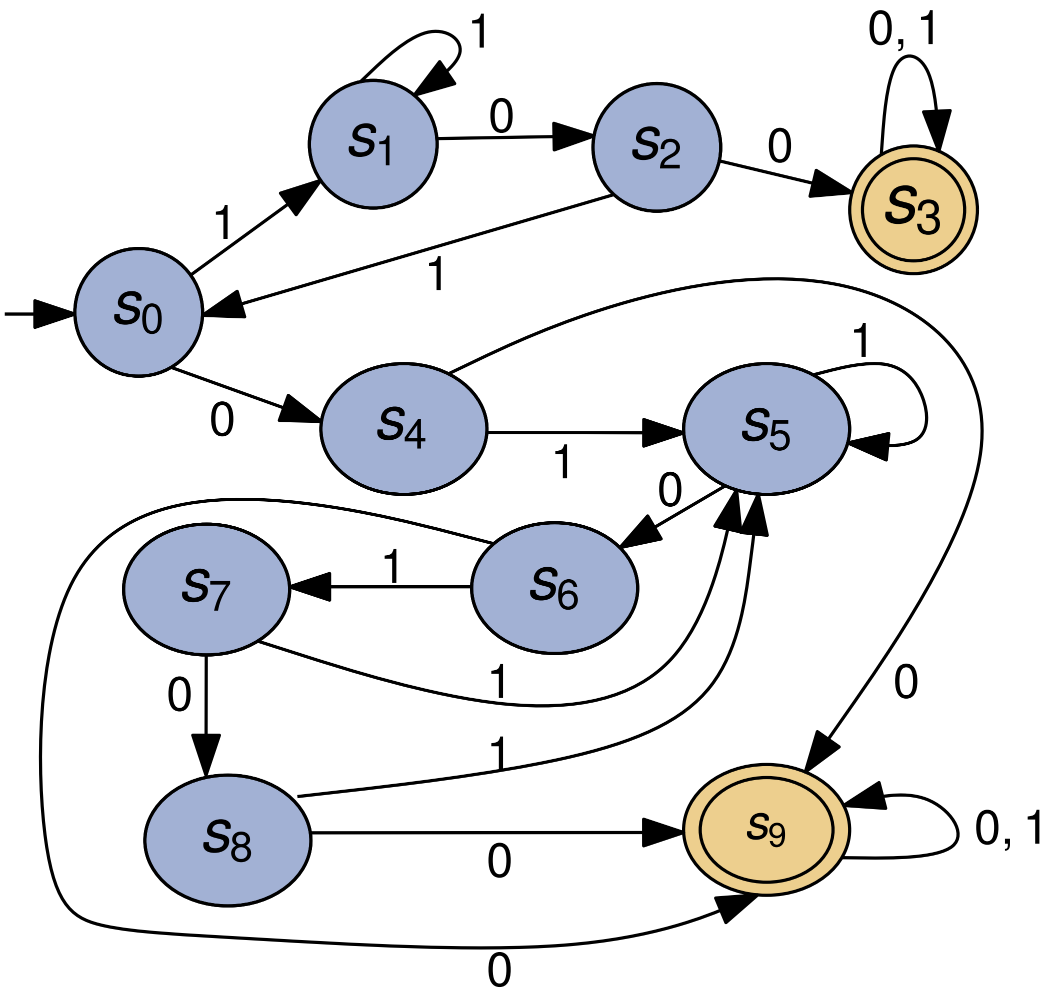


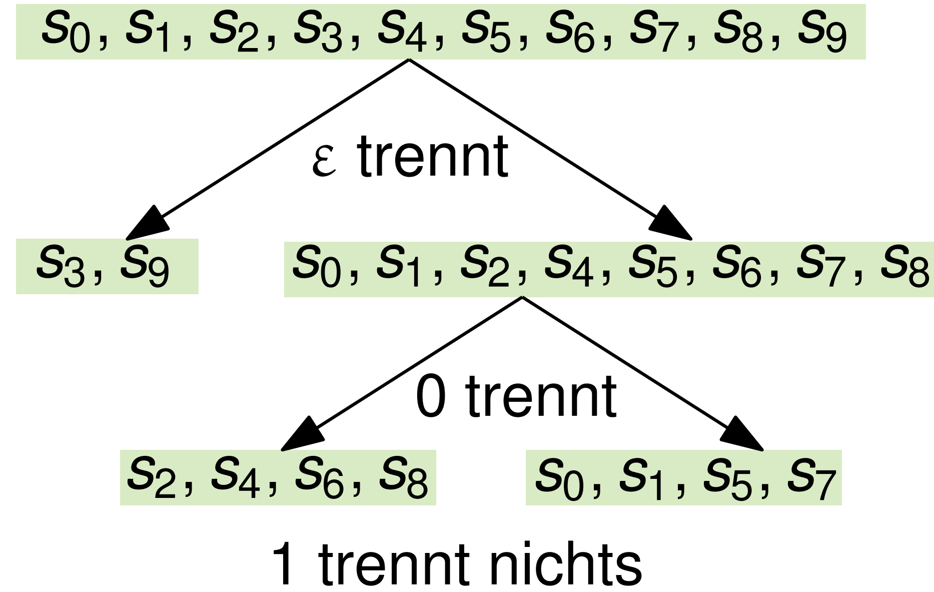


Betrachte Wort $w = 0$

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.

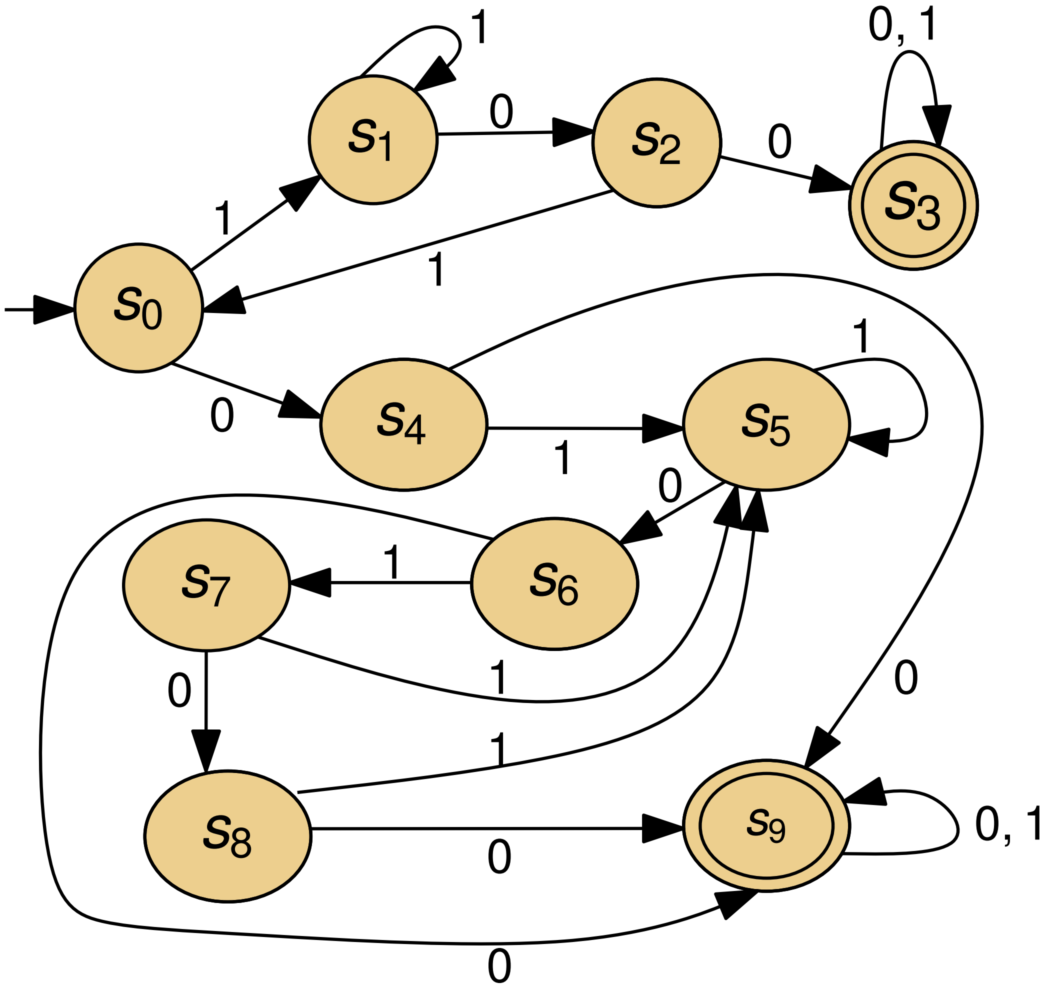


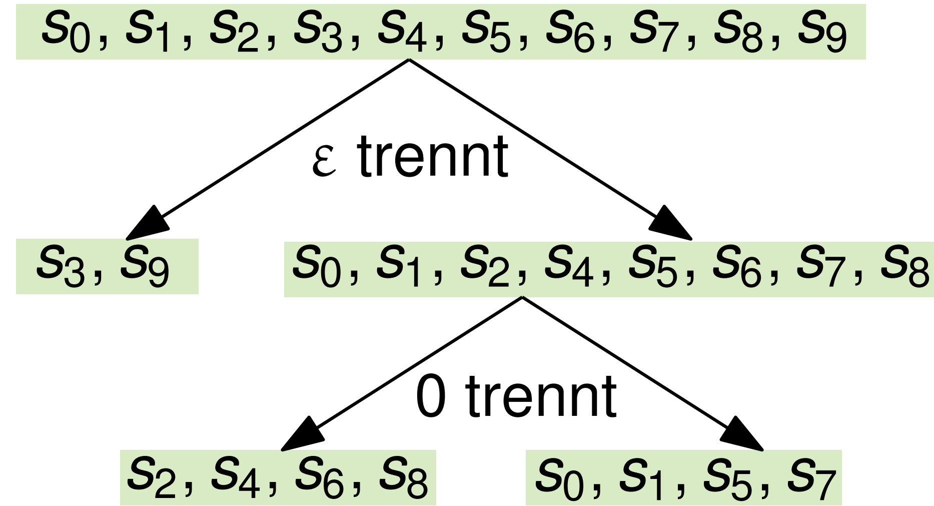


Betrachte Wort $w = 1$

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.





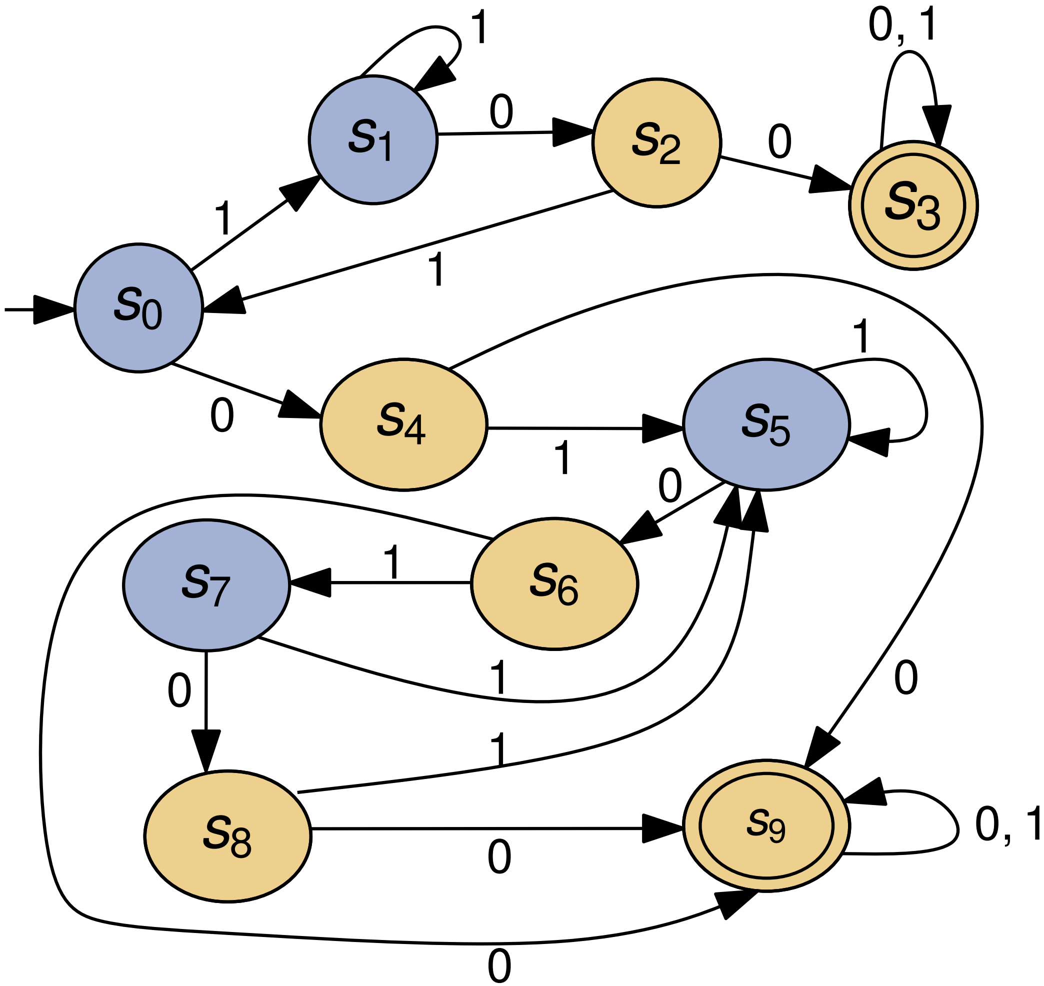
1 trennt nichts

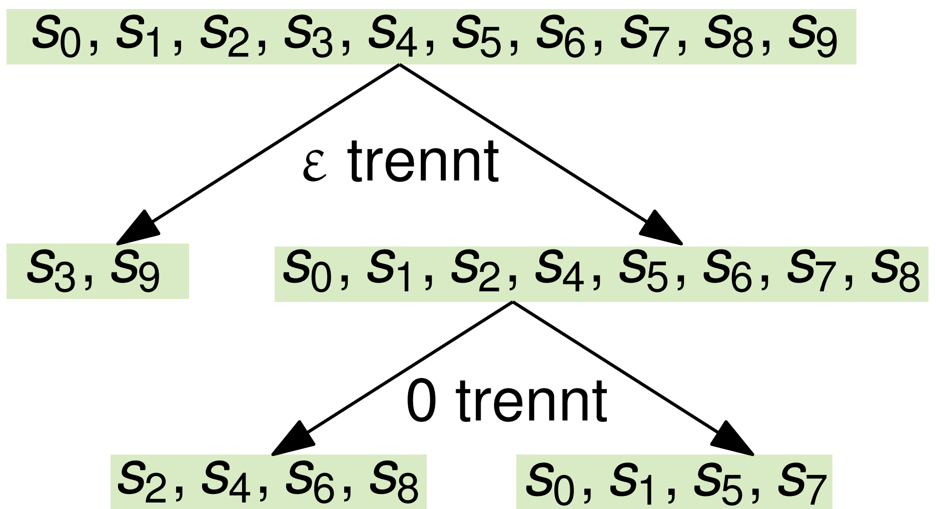
00 trennt nichts

Betrachte Wort $w =$ 00

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.





1 trennt nichts

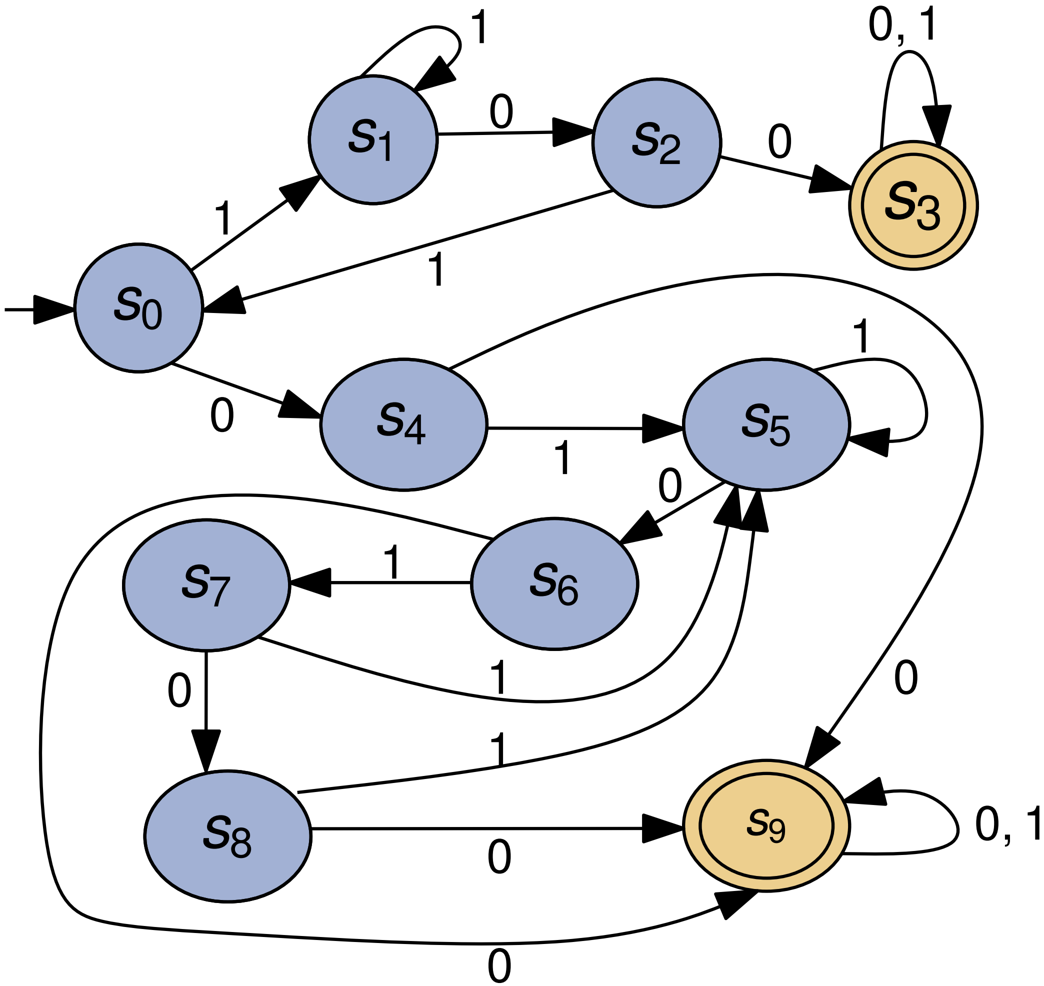
00 trennt nichts

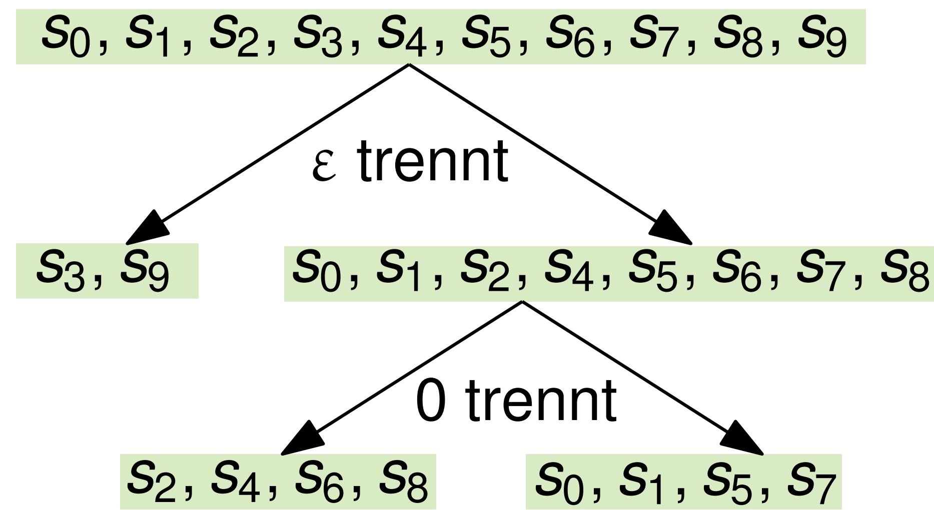
01 trennt nichts

Betrachte Wort $w = $ 01

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.





1 trennt nichts

00 trennt nichts

01 trennt nichts

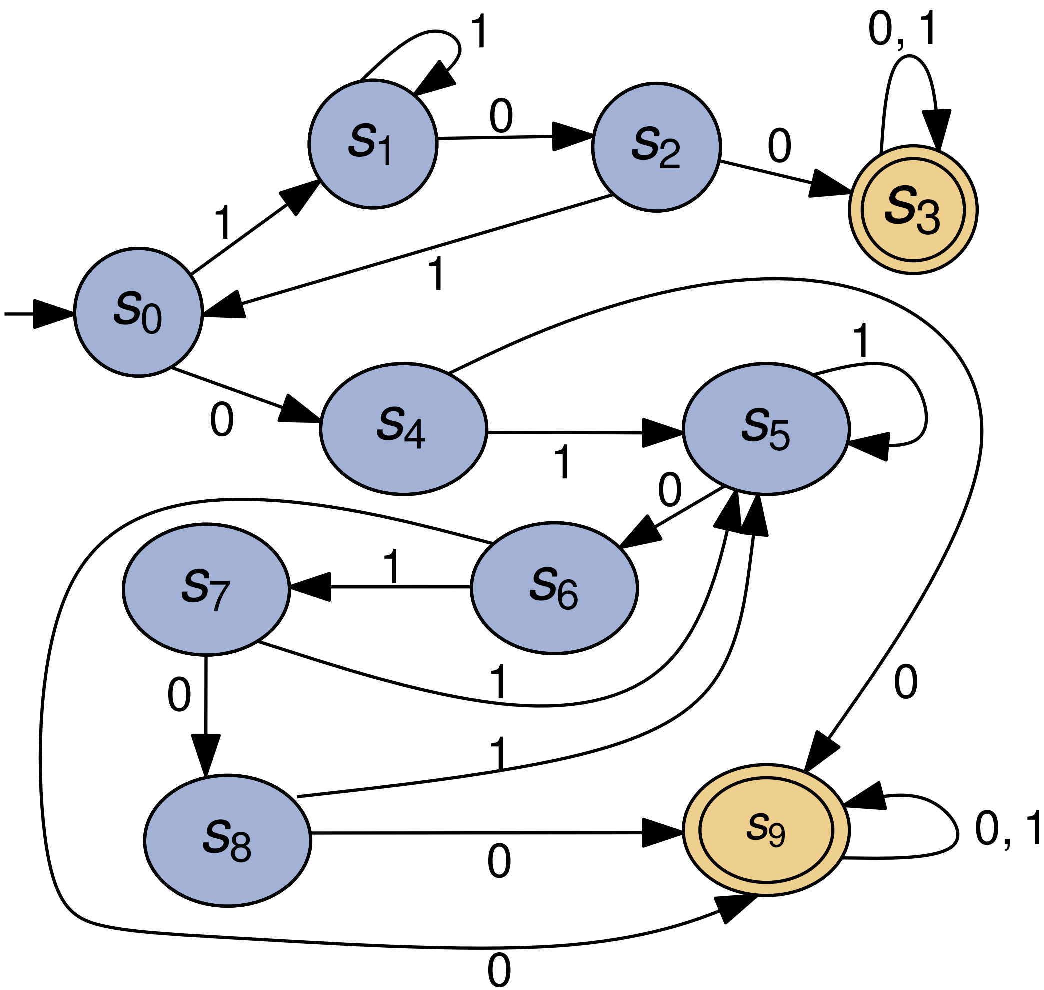
10 trennt nichts

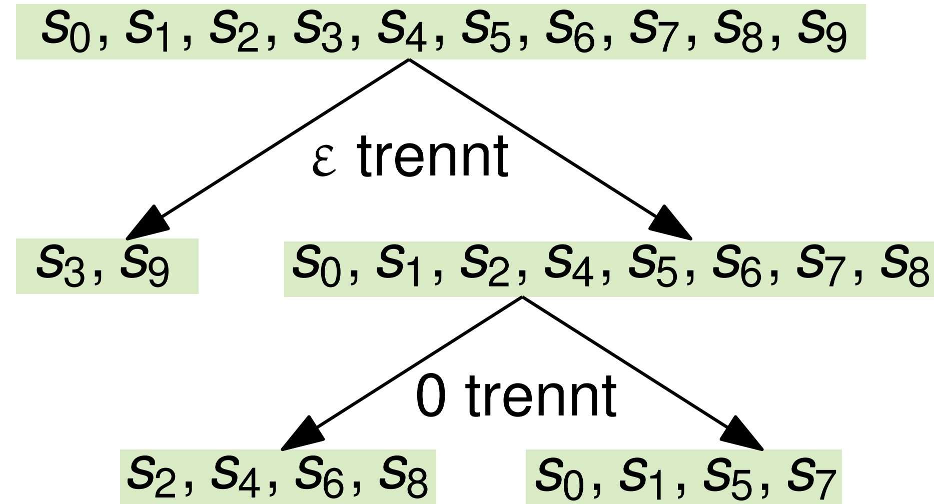
Betrachte Wort $w =$ 10

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik



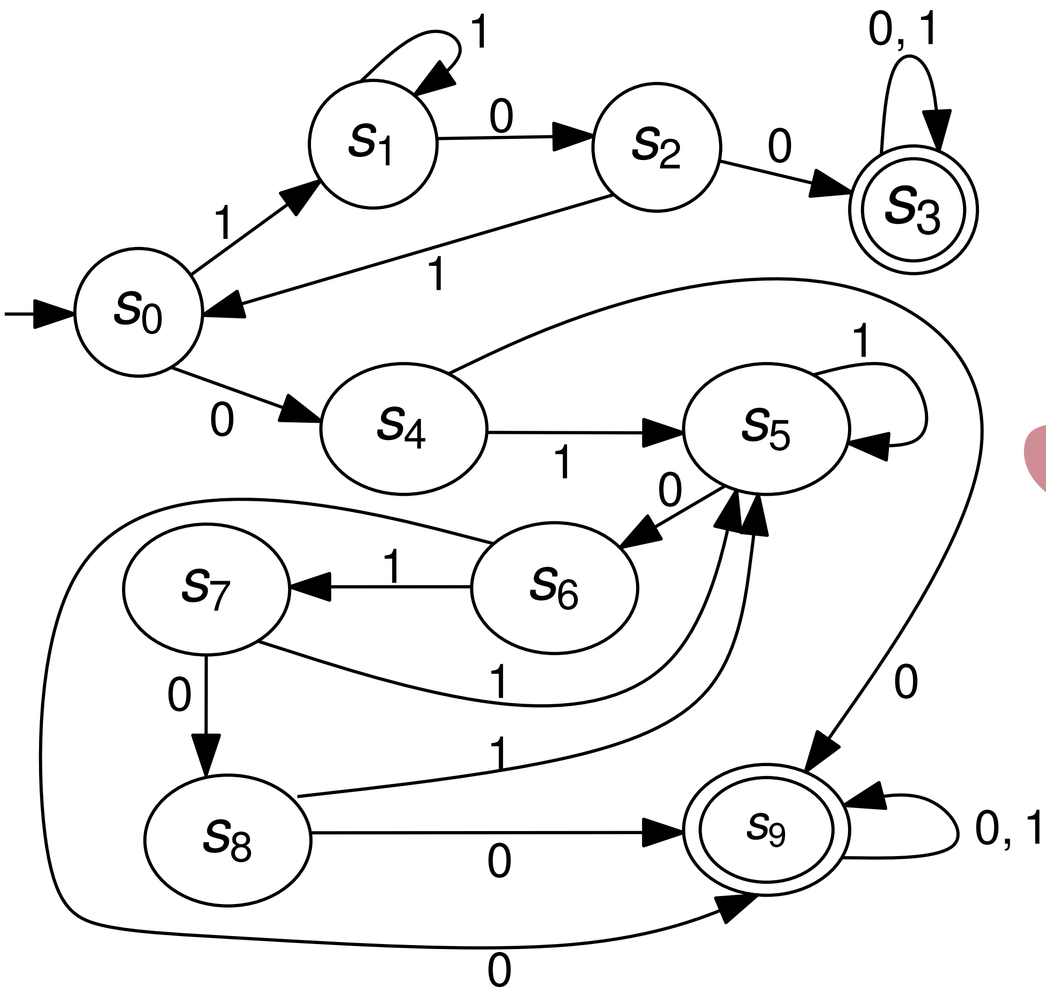


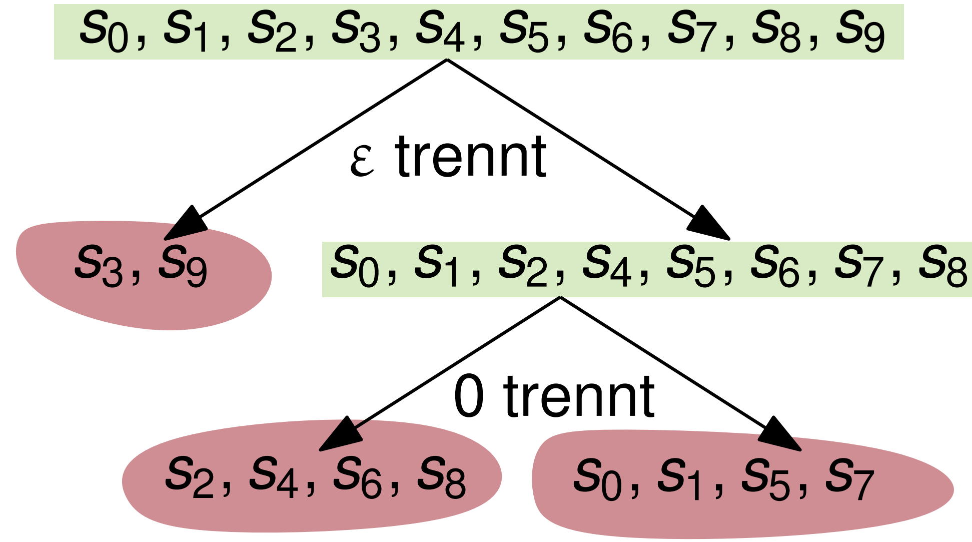
1 trennt nichts
00 trennt nichts
01 trennt nichts
10 trennt nichts
11 trennt nichts

Betrachte Wort $w = 11$

$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.



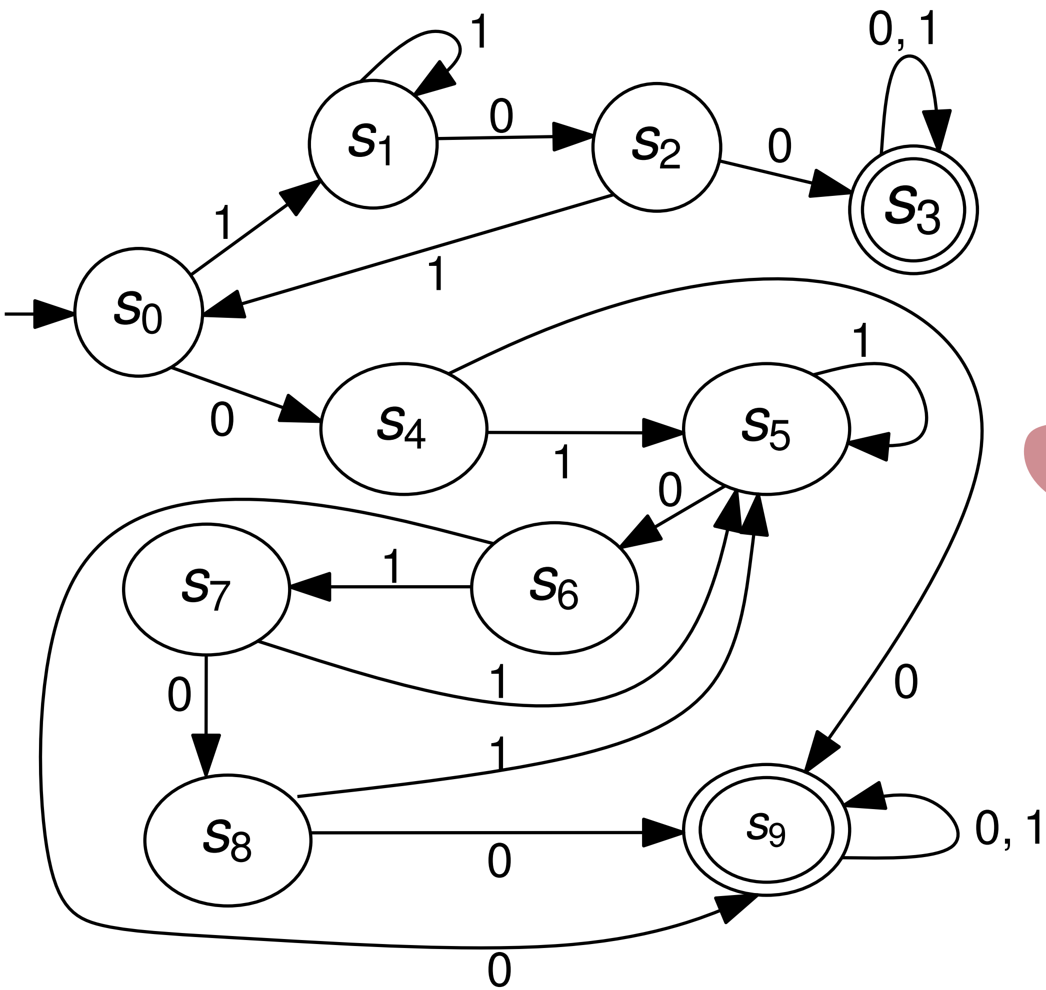


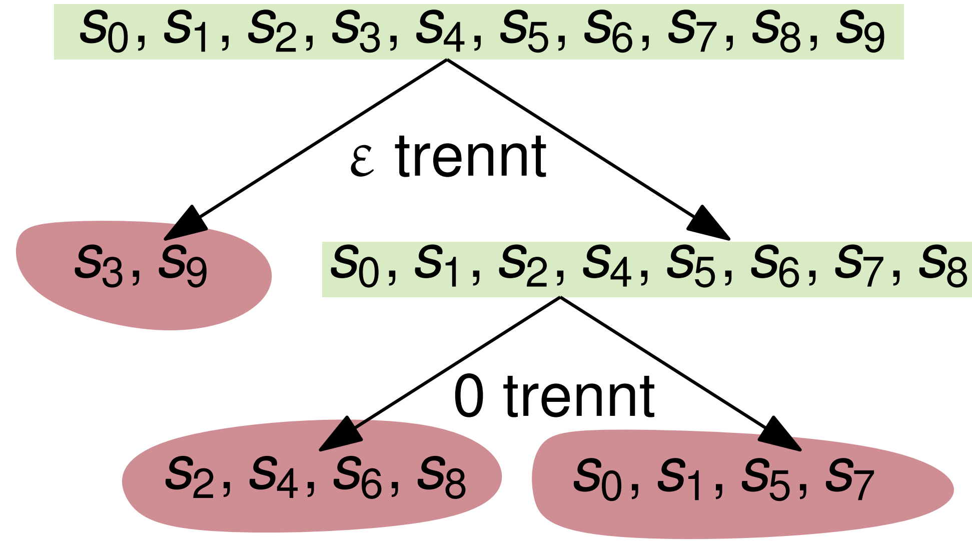
3 Äquivalenzklassen!

Betrachte Wort $w = 11$

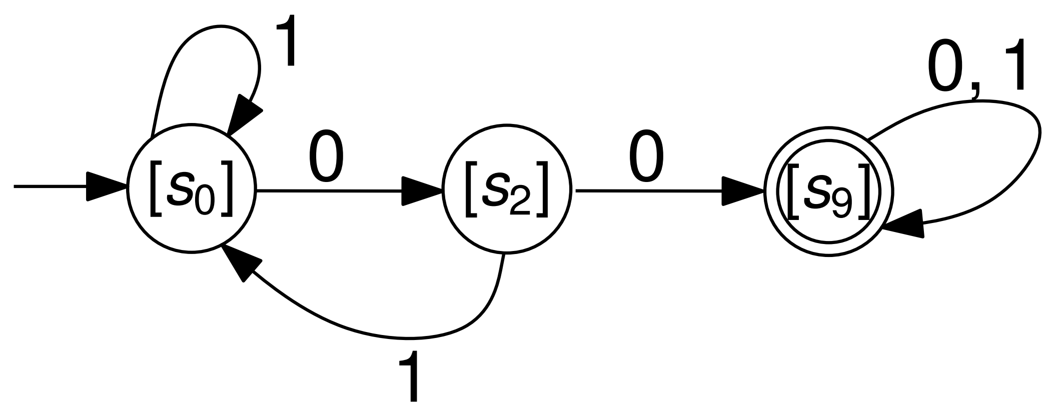
$\delta(s_i, w)$ endet in Endzustand.

$\delta(s_i, w)$ endet nicht in Endzustand.





3 Äquivalenzklassen!

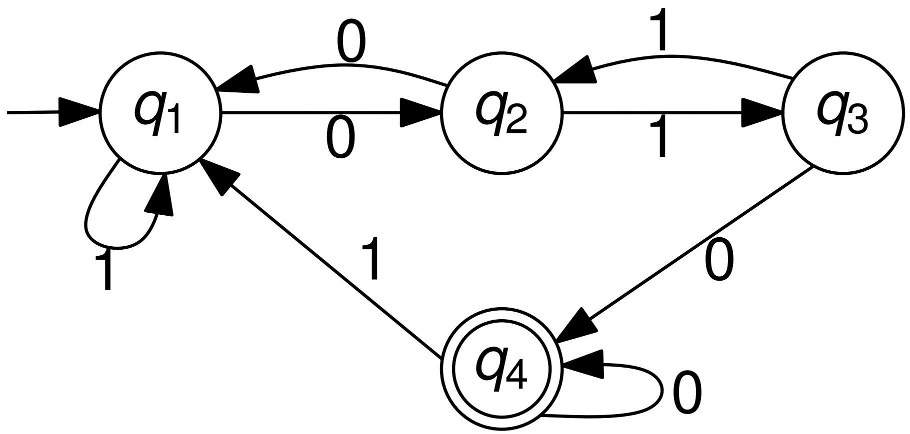


Betrachte Wort $w = 11$

$\delta(s_i, w)$ endet in Endzustand.

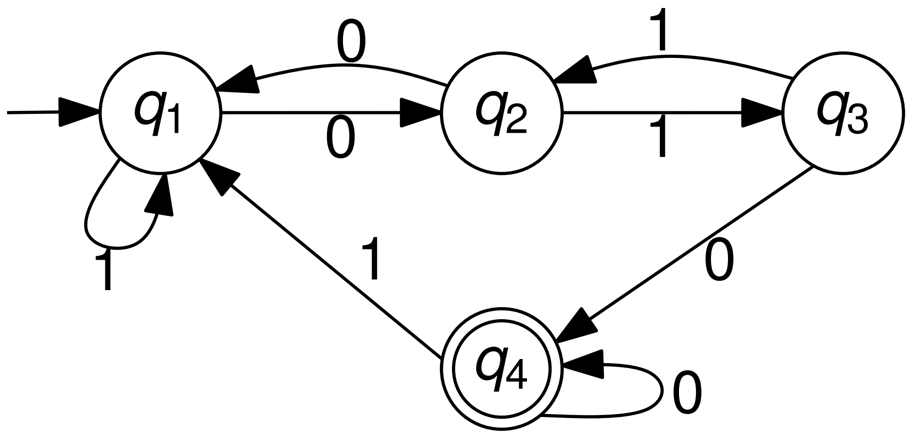
$\delta(s_i, w)$ endet nicht in Endzustand.

Minimierung von Automaten



Zeigen Sie, dass dieser Automat bezüglich der Äquivalenzklassenkonstruktion minimal ist.

Minimierung von Automaten

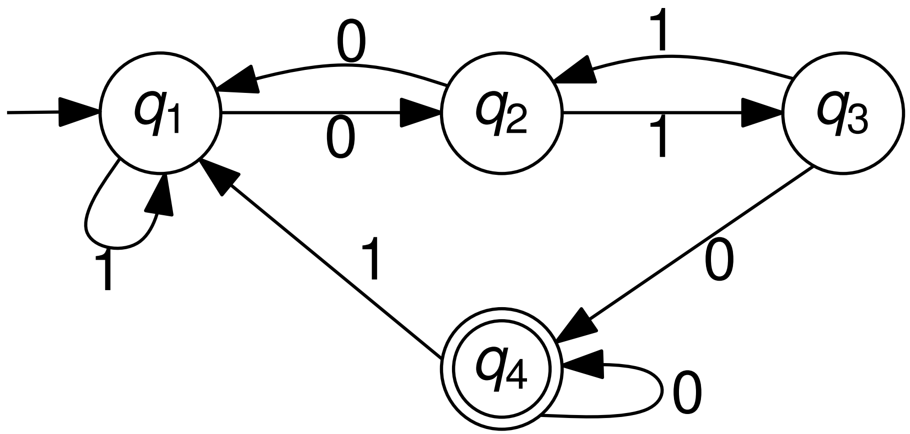


Zeigen Sie, dass dieser Automat bezüglich der Äquivalenzklassenkonstruktion minimal ist.

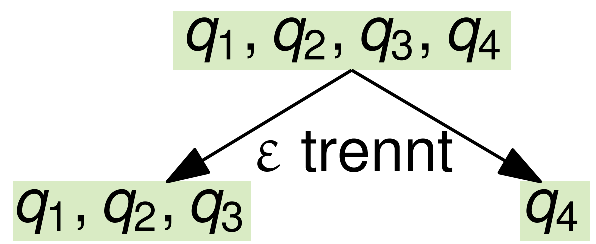
q_1, q_2, q_3, q_4

Betrachte: ε, 0, 1, 01, 10, 11,. . .

Minimierung von Automaten



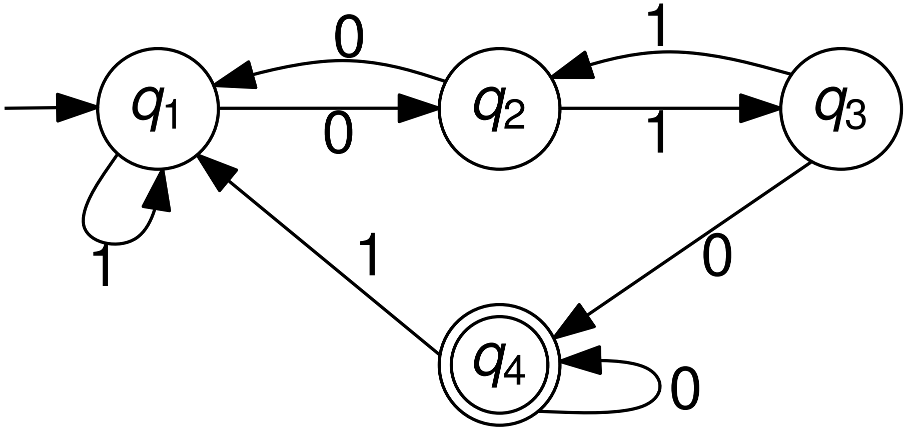
Zeigen Sie, dass dieser Automat bezüglich der Äquivalenzklassenkonstruktion minimal ist.



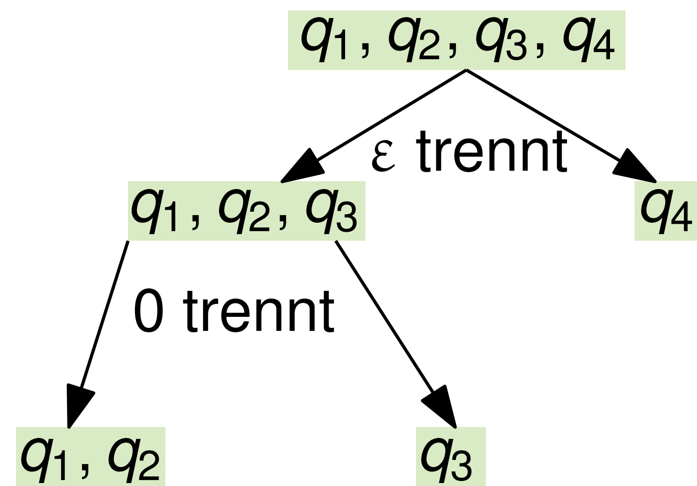
Betrachte: ε, 0, 1, 01, 10, 11,. . .

Minimierung von Automaten

Zeigen Sie, dass dieser Automat bezüglich der Äquivalenzklassenkonstruktion minimal ist.

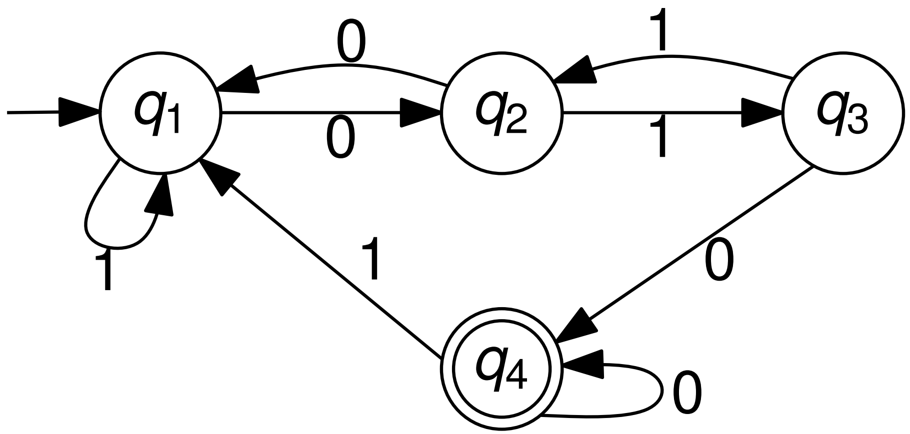


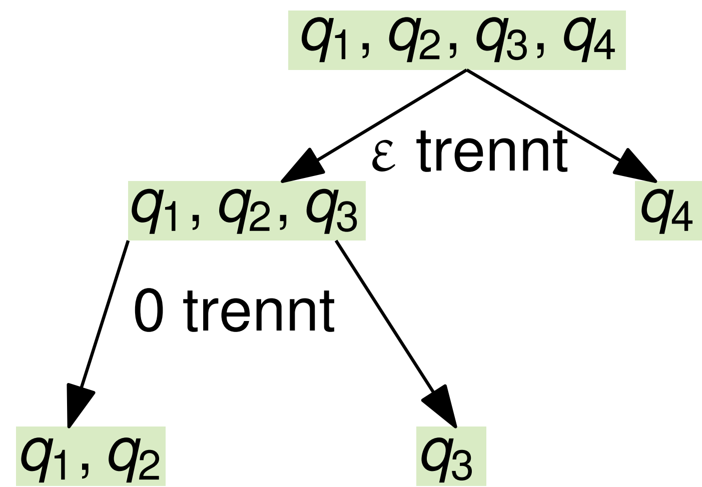
Betrachte: ε, 0, 1, 01, 10, 11,...



Minimierung von Automaten

Zeigen Sie, dass dieser Automat bezüglich der Äquivalenzklassenkonstruktion minimal ist.





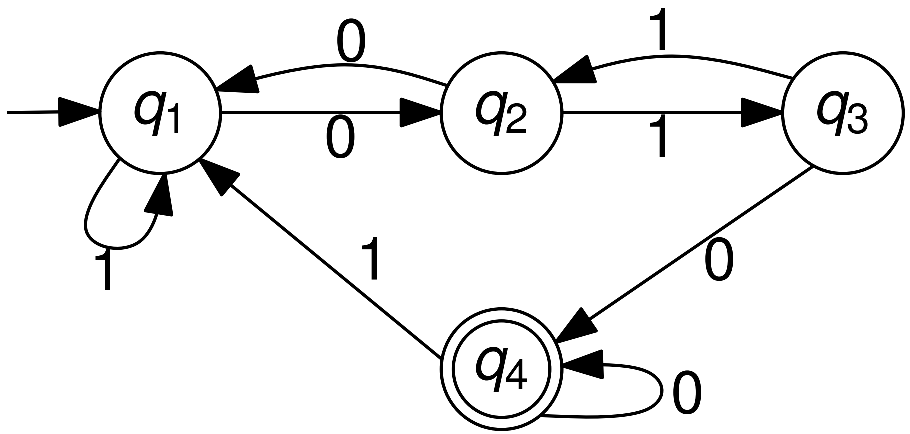
Betrachte: ε, 0, 1, 01, 10, 11,. . .

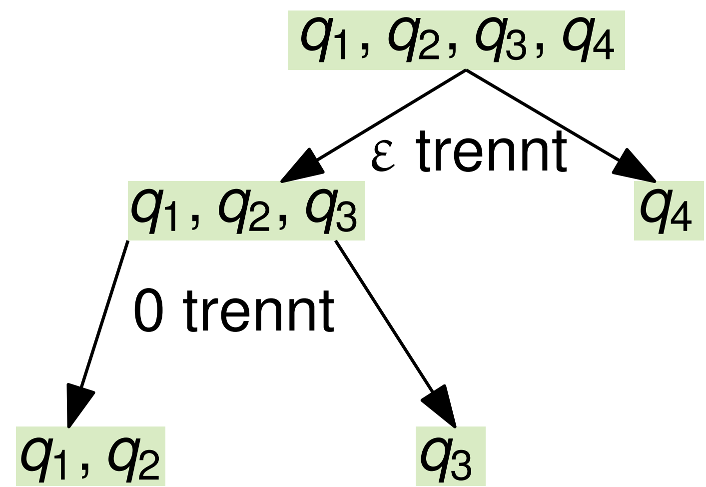
1 trennt nichts

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Minimierung von Automaten

Zeigen Sie, dass dieser Automat bezüglich der Äquivalenzklassenkonstruktion minimal ist.





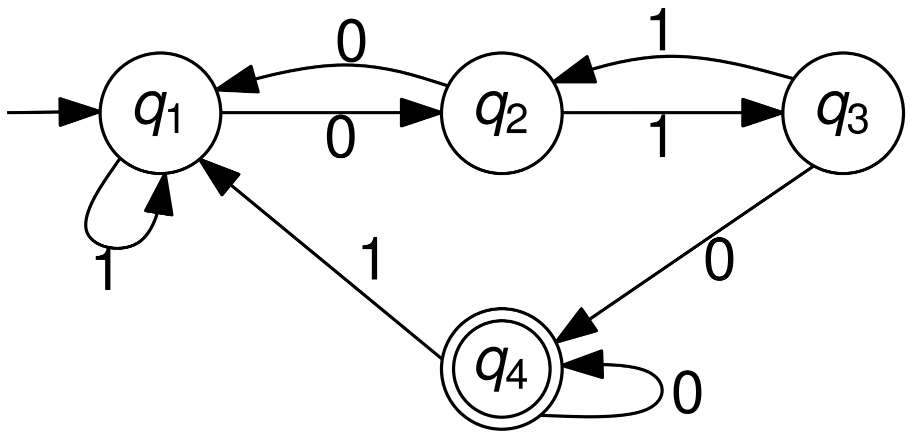
Betrachte: ε, 0, 1, 01, 10, 11,. . .

1 trennt nichts

01 trennt nichts

Minimierung von Automaten

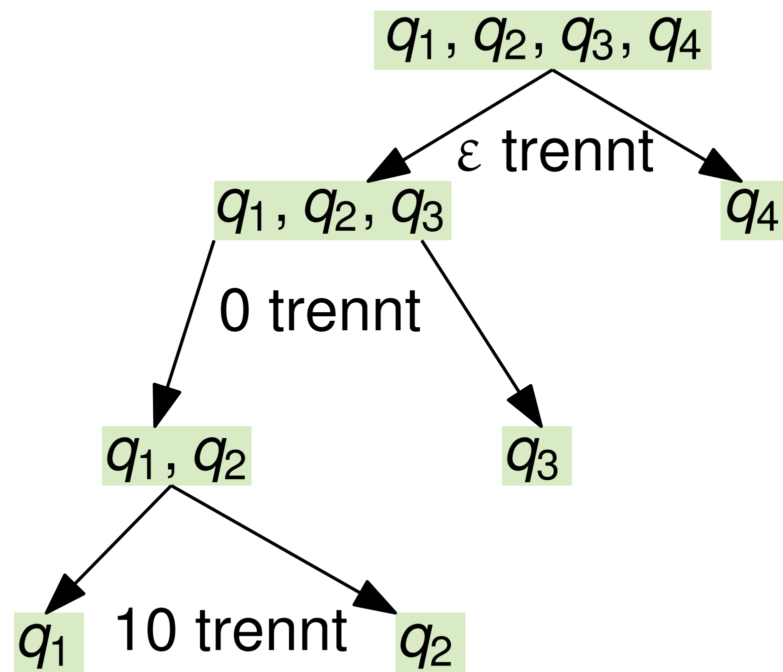
Zeigen Sie, dass dieser Automat bezüglich der Äquivalenzklassenkonstruktion minimal ist.



Betrachte: ε, 0, 1, 01, 10, 11,...

1 trennt nichts

01 trennt nichts



4 Äquivalenzklassen!

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Diagonalargument: nehme an, $(0, 1)$ wäre abzählbar mit Bijektion $f : \mathbb{N} \to (0, 1)$. Konstruiere Zahl $x \in (0, 1)$ mit $x \notin \mathrm{Bild}(f)$. Widerspr.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Diagonalargument: nehme an, $(0, 1)$ wäre abzählbar mit Bijektion $f : \mathbb{N} \to (0, 1)$. Konstruiere Zahl $x \in (0, 1)$ mit $x \notin \text{Bild}(f)$. Widerspr. ↯

		1	2	3	4	5	6	7	8	9	10
$f(1) =$	0,	9	9	6	8	8	3	6	0	4	7
$f(2) =$	0,	1	2	3	9	8	1	2	6	4	9
$f(3) =$	0,	8	2	7	3	7	4	5	8	0	1
$f(4) =$	0,	1	7	8	7	4	9	0	3	0	7
$f(5) =$	0,	7	9	8	4	3	2	9	3	2	3

$x = 0,$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Diagonalargument: nehme an, $(0,1)$ wäre abzählbar mit Bijektion $f : \mathbb{N} \to (0,1)$. Konstruiere Zahl $x \in (0,1)$ mit $x \notin \text{Bild}(f)$. Widerspr.

		1	2	3	4	5	6	7	8	9	10
$f(1) =$	0,	9	9	6	8	8	3	6	0	4	7
$f(2) =$	0,	1	2	3	9	8	1	2	6	4	9
$f(3) =$	0,	8	2	7	3	7	4	5	8	0	1
$f(4) =$	0,	1	7	8	7	4	9	0	3	0	7
$f(5) =$	0,	7	9	8	4	3	2	9	3	2	3

$\longrightarrow x \neq f(1)$

$x = 0,\ 7$

Idee: wähle 7 falls möglich, sonst 0.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Diagonalargument: nehme an, $(0,1)$ wäre abzählbar mit Bijektion $f : \mathbb{N} \to (0,1)$. Konstruiere Zahl $x \in (0,1)$ mit $x \notin \text{Bild}(f)$. Widerspr.

		1	2	3	4	5	6	7	8	9	10	
$f(1) =$	0,	9	9	6	8	8	3	6	0	4	7	$\to x \neq f(1)$
$f(2) =$	0,	1	2	3	9	8	1	2	6	4	9	$\to x \neq f(2)$
$f(3) =$	0,	8	2	7	3	7	4	5	8	0	1	
$f(4) =$	0,	1	7	8	7	4	9	0	3	0	7	
$f(5) =$	0,	7	9	8	4	3	2	9	3	2	3	
$x =$	0,	7	7									

Idee: wähle 7 falls möglich, sonst 0.

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Diagonalargument: nehme an, $(0, 1)$ wäre abzählbar mit Bijektion $f : \mathbb{N} \to (0, 1)$. Konstruiere Zahl $x \in (0, 1)$ mit $x \notin \text{Bild}(f)$. Widerspr.

		1	2	3	4	5	6	7	8	9	10	
$f(1) =$	0,	9	9	6	8	8	3	6	0	4	7	$\to x \neq f(1)$
$f(2) =$	0,	1	2	3	9	8	1	2	6	4	9	$\to x \neq f(2)$
$f(3) =$	0,	8	2	7	3	7	4	5	8	0	1	$\to x \neq f(3)$
$f(4) =$	0,	1	7	8	7	4	9	0	3	0	7	
$f(5) =$	0,	7	9	8	4	3	2	9	3	2	3	
$x =$	0,	7	7	0								

Idee: wähle 7 falls möglich, sonst 0.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Diagonalargument: nehme an, $(0, 1)$ wäre abzählbar mit Bijektion $f : \mathbb{N} \to (0, 1)$. Konstruiere Zahl $x \in (0, 1)$ mit $x \notin \text{Bild}(f)$. Widerspr.

		1	2	3	4	5	6	7	8	9	10	
$f(1) =$	0,	9	9	6	8	8	3	6	0	4	7	$\longrightarrow x \neq f(1)$
$f(2) =$	0,	1	2	3	9	8	1	2	6	4	9	$\longrightarrow x \neq f(2)$
$f(3) =$	0,	8	2	7	3	7	4	5	8	0	1	$\longrightarrow x \neq f(3)$
$f(4) =$	0,	1	7	8	7	4	9	0	3	0	7	$\longrightarrow x \neq f(4)$
$f(5) =$	0,	7	9	8	4	3	2	9	3	2	3	
$x =$	0,	7	7	0	0							

Idee: wähle 7 falls möglich, sonst 0.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Diagonalargument: nehme an, $(0,1)$ wäre abzählbar mit Bijektion $f : \mathbb{N} \to (0,1)$. Konstruiere Zahl $x \in (0,1)$ mit $x \notin \text{Bild}(f)$. Widerspr.

		1	2	3	4	5	6	7	8	9	10	
$f(1) =$	0,	9	9	6	8	8	3	6	0	4	7	$\longrightarrow x \neq f(1)$
$f(2) =$	0,	1	2	3	9	8	1	2	6	4	9	$\longrightarrow x \neq f(2)$
$f(3) =$	0,	8	2	7	3	7	4	5	8	0	1	$\longrightarrow x \neq f(3)$
$f(4) =$	0,	1	7	8	7	4	9	0	3	0	7	$\longrightarrow x \neq f(4)$
$f(5) =$	0,	7	9	8	4	3	2	9	3	2	3	$\longrightarrow x \neq f(5)$

$x = 0,\ 7\ 7\ 0\ 0\ 7\ \ldots$

Idee: wähle 7 falls möglich, sonst 0.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Diagonalargument: nehme an, $(0, 1)$ wäre abzählbar mit Bijektion $f : \mathbb{N} \to (0, 1)$. Konstruiere Zahl $x \in (0, 1)$ mit $x \notin \text{Bild}(f)$. Widerspr.

		1	2	3	4	5	6	7	8	9	10
$f(1) =$	0,	$\boxed{9}$	9	6	8	8	3	6	0	4	7
$f(2) =$	0,	1	$\boxed{2}$	3	9	8	1	2	6	4	9
$f(3) =$	0,	8	2	$\boxed{7}$	3	7	4	5	8	0	1
$f(4) =$	0,	1	7	8	$\boxed{7}$	4	9	0	3	0	7
$f(5) =$	0,	7	9	8	4	$\boxed{3}$	2	9	3	2	3

$x = 0,\ 7\ 7\ 0\ 0\ 7\ \ldots$

$\longrightarrow x \notin \text{Bild}(f)$

Allgemein: bezeichne y_i den Wert der i-ten Nachkommastelle von y.

$$x_i := \begin{cases} 7 & \text{falls } f(i)_i \neq 7 \\ 0 & \text{falls } f(i)_i = 7 \end{cases}$$

Idee: wähle 7 falls möglich, sonst 0.

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Cantors 2. Diagonalargument

Theorem: Die Menge der reellen Zahlen ist überabzählbar.

Eine Menge heißt abzählbar, wenn sie entweder endlich ist oder eine Bijektion zur Menge der natürlichen Zahlen existiert.

Diagonalargument: nehme an, $(0, 1)$ wäre abzählbar mit Bijektion $f : \mathbb{N} \to (0, 1)$. Konstruiere Zahl $x \in (0, 1)$ mit $x \notin$ Bild(f). Widerspr.

Wieso wir das machen...

In der Vorlesung wird bald die **Diagonalsprache** vorgestellt, die von Turingmaschinen nicht erkannt werden kann.
Deren Konstruktion ist sehr ähnlich zu Cantors 2. Diagonalargument!