

Theoretische Grundlagen der Informatik

Übung

1. Übungstermin · 26. Oktober
Torsten Ueckerdt

INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

Übungsleiter

Torsten Ueckerdt
torsten.ueckerdt@kit.edu
Raum 319

Guido Brückner
brueckner@kit.edu
Raum 317

Sprechzeiten: Termin nach Vereinbarung



Gliederung

Organisatorisches

- Übungsblätter

Inhalt

- Formale Sprachen und reguläre Ausdrücke
- Nicht-deterministische endliche Automaten
- Eigenschaften von endlichen Automaten
- Potenzmengenkonstruktion
- ε -Abschluss
- Kontextfreie Grammatiken

Übungsblatt (ÜB)

Aus- und Abgabe: jeden zweiten Dienstag

Tag der Abgabe für aktuelles ÜB

=

Tag der Ausgabe für nächstes ÜB



Ab 25% der erreichbaren Punkte gibt es 1 Bonuspunkt auf die **bestandene** Klausur.

- ab 50% gibt es 2 Bonuspunkte
- ab 75% gibt es 3 Bonuspunkte



Übungsblattabgabe

Titelblatt:

WebInScribe Deckblattgenerator benutzen!

Übungsblattabgabe

Titelblatt:

WebInScribe Deckblattgenerator benutzen!



Handschriftliche Abgabe!

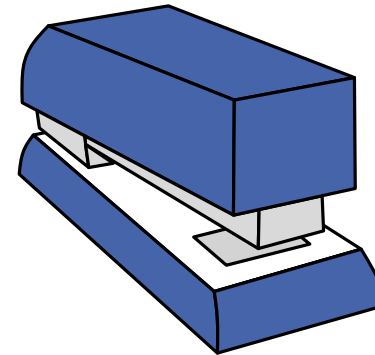
Übungsblattabgabe

Titelblatt:

WebInScribe Deckblattgenerator benutzen!



Handschriftliche Abgabe!



Übungsblatt heften!

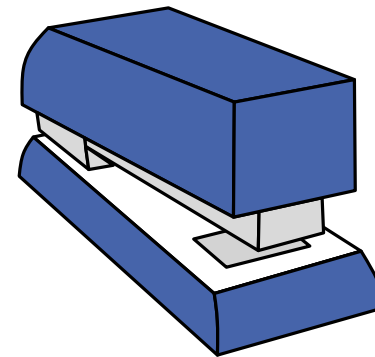
Übungsblattabgabe

Titelblatt:

WebInScribe Deckblattgenerator benutzen!



Handschriftliche Abgabe!



Übungsblatt heften!

Doppelabgabe ist erlaubt.

→ Beide müssen im selben Tutorium angemeldet sein.

Gliederung

Organisatorisches

- Übungsblätter

Inhalt

- Formale Sprachen und reguläre Ausdrücke
- Nicht-deterministische endliche Automaten
- Eigenschaften von endlichen Automaten
- Potenzmengenkonstruktion
- ε -Abschluss
- Kontextfreie Grammtiken

Formale Sprachen

Alphabet: Endliche Menge Σ an Zeichen/Symbolen.

Beispiel: $\Sigma = \{0, 1\}$

Wort: Ein Wort w über dem Alphabet Σ ist eine (endliche) Folge von Zeichen aus Σ .

Beispiel: $\Sigma = \{0, 1\}$, $w=0101$, $w=\varepsilon$

Leere Wort: ε

Alle Worte: Σ^*

Beispiel: $\Sigma = \{0, 1\}$, $\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$

Sprache: (beliebige) Menge an Worten über einem Alphabet Σ

Beispiel: $\Sigma = \{0, 1\}$

$$L_1 = \{00, 01, 10, 11\}$$

$$L_2 = \{w \in \Sigma^* \mid w \text{ endet mit } 0\}$$

Formale Sprachen

Seien $L, L_1, L_2 \subseteq \Sigma^*$ Sprachen.

Produktsprache $L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$

k -faches Produkt $L^k := \{w_1 \cdot w_2 \cdot \dots \cdot w_k \mid w_i \in L \text{ für } 1 \leq i \leq k\}$
 $L^0 := \{\varepsilon\}$

Kleene'scher Abschluss $L^* := \bigcup_{i \geq 0} L^i$

Positiver Abschluss $L^+ := \bigcup_{i \geq 1} L^i$

Quotientensprache $L_1/L_2 := \{w \in \Sigma^* \mid \exists z \in L_2 \text{ mit } w \cdot z \in L_1\}$

Komplementsprache $L^c := \Sigma^* \setminus L$

Beispiel: Produktsprache

$L_1 = \{01, 10, 11\}$

$L_2 = \{1, 00, 101\}$

Formale Sprachen

Seien $L, L_1, L_2 \subseteq \Sigma^*$ Sprachen.

Produktsprache $L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$

k -faches Produkt $L^k := \{w_1 \cdot w_2 \cdot \dots \cdot w_k \mid w_i \in L \text{ für } 1 \leq i \leq k\}$
 $L^0 := \{\varepsilon\}$

Kleene'scher Abschluss $L^* := \bigcup_{i \geq 0} L^i$

Positiver Abschluss $L^+ := \bigcup_{i \geq 1} L^i$

Quotientensprache $L_1 / L_2 := \{w \in \Sigma^* \mid \exists z \in L_2 \text{ mit } w \cdot z \in L_1\}$

Komplementsprache $L^c := \Sigma^* \setminus L$

Beispiel: Produktsprache

$L_1 = \{01, 10, 11\}$
 $L_2 = \{1, 00, 101\}$
 $L_1 \cdot L_2 = \{011, 0100, 01101, \dots\}$

Formale Sprachen

Seien $L, L_1, L_2 \subseteq \Sigma^*$ Sprachen.

Produktsprache $L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$

k -faches Produkt $L^k := \{w_1 \cdot w_2 \cdot \dots \cdot w_k \mid w_i \in L \text{ für } 1 \leq i \leq k\}$
 $L^0 := \{\varepsilon\}$

Kleene'scher Abschluss $L^* := \bigcup_{i \geq 0} L^i$

Positiver Abschluss $L^+ := \bigcup_{i \geq 1} L^i$

Quotientensprache $L_1/L_2 := \{w \in \Sigma^* \mid \exists z \in L_2 \text{ mit } w \cdot z \in L_1\}$

Komplementsprache $L^c := \Sigma^* \setminus L$

Beispiel: Produktsprache

$L_1 = \{01, 10, 11\}$
 $L_2 = \{1, 00, 101\}$
 $L_1 \cdot L_2 = \{011, 0100, 01101, 101, 1000, 10101, \dots\}$

Formale Sprachen

Seien $L, L_1, L_2 \subseteq \Sigma^*$ Sprachen.

Produktsprache $L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$

k -faches Produkt $L^k := \{w_1 \cdot w_2 \cdot \dots \cdot w_k \mid w_i \in L \text{ für } 1 \leq i \leq k\}$
 $L^0 := \{\varepsilon\}$

Kleene'scher Abschluss $L^* := \bigcup_{i \geq 0} L^i$

Positiver Abschluss $L^+ := \bigcup_{i \geq 1} L^i$

Quotientensprache $L_1 / L_2 := \{w \in \Sigma^* \mid \exists z \in L_2 \text{ mit } w \cdot z \in L_1\}$

Komplementsprache $L^c := \Sigma^* \setminus L$

Beispiel: Produktsprache

$L_1 = \{01, 10, 11\}$
 $L_2 = \{1, 00, 101\}$
 $L_1 \cdot L_2 = \{011, 0100, 01101, 101, 1000, 10101, 111, 1100, 11101\}$

Reguläre Sprachen

Eine Sprache $L \subseteq \Sigma^*$ heißt *regulär*, wenn für sie einer der folgenden Punkte gilt: (induktive Definition)

1. Verankerung:

- (a) $L = \{a\}$ mit $a \in \Sigma \cup \{\varepsilon\}$ oder
- (b) $L = \emptyset$

2. Induktion: Seien L_1, L_2 reguläre Sprachen

- (a) $L = L_1 \cdot L_2$ oder
- (b) $L = L_1 \cup L_2$ oder
- (c) $L = L_1^*$

Reguläre Sprachen

Eine Sprache $L \subseteq \Sigma^*$ heißt *regulär*, wenn für sie einer der folgenden Punkte gilt: (induktive Definition)

1. Verankerung:

- (a) $L = \{a\}$ mit $a \in \Sigma \cup \{\varepsilon\}$ oder
- (b) $L = \emptyset$

2. Induktion: Seien L_1, L_2 reguläre Sprachen

- (a) $L = L_1 \cdot L_2$ oder
- (b) $L = L_1 \cup L_2$ oder
- (c) $L = L_1^*$

Reguläre Ausdrücke:

- $a \in \Sigma \cup \{\varepsilon\}$ für $L = \{a\}$
- $(\alpha) \cup (\beta)$ für $L(\alpha) \cup L(\beta)$
- $(\alpha) \cdot (\beta)$ für $L(\alpha) \cdot L(\beta)$
- $(\alpha)^+$ für $L(\alpha)^+$
- $(\alpha)^*$ für $L(\alpha)^*$

Reguläre Sprachen

Eine Sprache $L \subseteq \Sigma^*$ heißt *regulär*, wenn für sie einer der folgenden Punkte gilt: (induktive Definition)

1. Verankerung:

- (a) $L = \{a\}$ mit $a \in \Sigma \cup \{\varepsilon\}$ oder
- (b) $L = \emptyset$

2. Induktion: Seien L_1, L_2 reguläre Sprachen

- (a) $L = L_1 \cdot L_2$ oder
- (b) $L = L_1 \cup L_2$ oder
- (c) $L = L_1^*$

Reguläre Ausdrücke:

- $a \in \Sigma \cup \{\varepsilon\}$ für $L = \{a\}$
- $(\alpha) \cup (\beta)$ für $L(\alpha) \cup L(\beta)$
- $(\alpha) \cdot (\beta)$ für $L(\alpha) \cdot L(\beta)$
- $(\alpha)^+$ für $L(\alpha)^+$
- $(\alpha)^*$ für $L(\alpha)^*$



Klammern werden häufig weggelassen.

Aufgabe

Seien A, B reguläre Ausdrücke. Welche regulären Ausdrücke beschreiben die gleiche Sprache?

(a) $(A^*)^* = A^*$

(b) $(A \cup B)^* = (A^* B^*)^*$

(c) $A^* \cup B^* = (A \cup B)^*$

Aufgabe

Seien A, B reguläre Ausdrücke. Welche regulären Ausdrücke beschreiben die gleiche Sprache?

(a) $(A^*)^* = A^*$

(b) $(A \cup B)^* = (A^* B^*)^*$

(c) $A^* \cup B^* = (A \cup B)^*$



3 min Zeit



Aufgabe

Seien A, B reguläre Ausdrücke. Welche regulären Ausdrücke beschreiben die gleiche Sprache?

(a) $(A^*)^* = A^*$



(b) $(A \cup B)^* = (A^* B^*)^*$



(c) $A^* \cup B^* = (A \cup B)^*$



Endliche Automaten

Definiert über einem Alphabet Σ

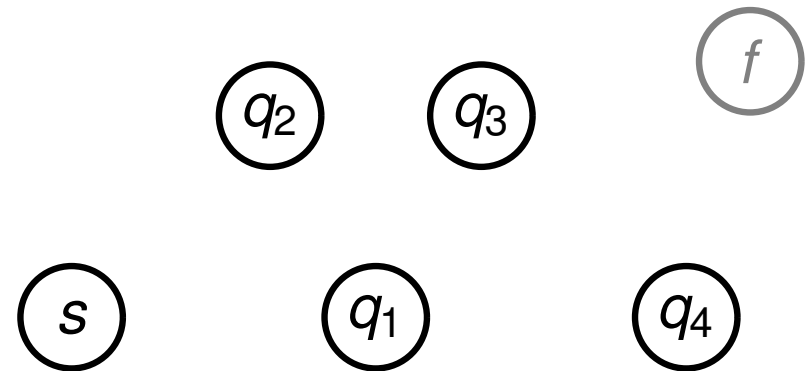
Bestehend aus:

Endliche Automaten

Definiert über einem Alphabet Σ

Bestehend aus:

→ Q : endliche Menge von *Zuständen*



Endliche Automaten

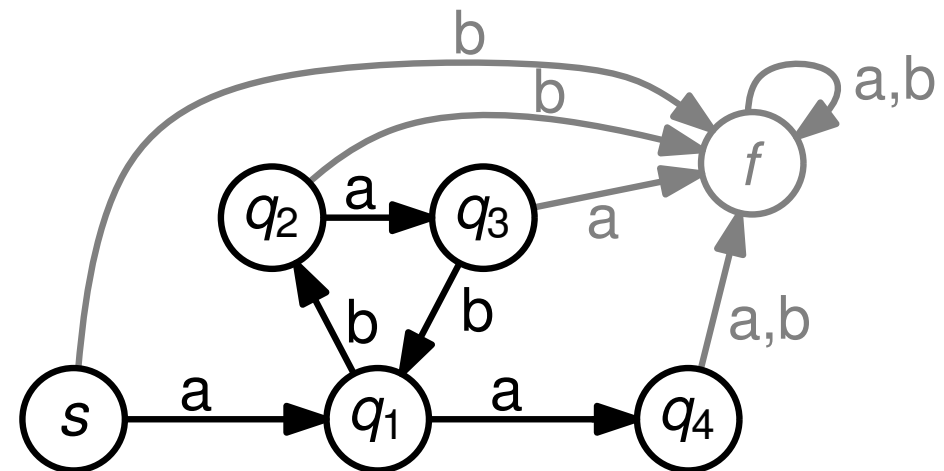
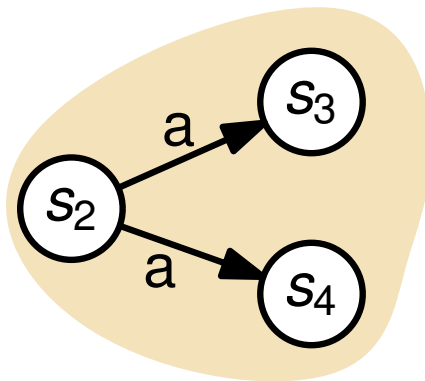
Definiert über einem Alphabet Σ

Bestehend aus:

→ Q : endliche Menge von *Zuständen*

→ Übergänge $\xrightarrow{\text{eindeutig}}$ $\delta: Q \times \Sigma \rightarrow Q$ deterministisch

$\xrightarrow{\text{mehrdeutig}}$ $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$ nicht-deterministisch



Endliche Automaten

Definiert über einem Alphabet Σ

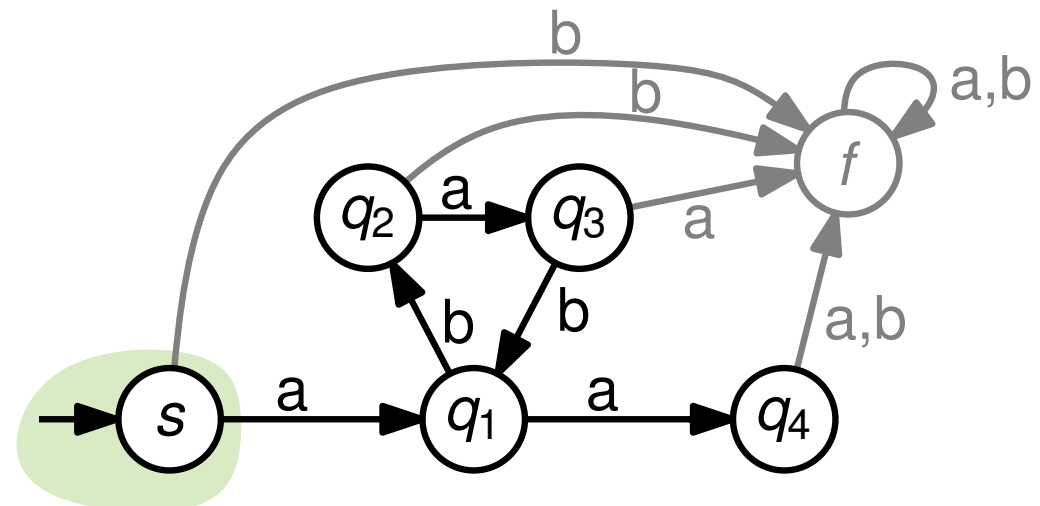
Bestehend aus:

→ Q : endliche Menge von *Zuständen*

→ Übergänge $\xrightarrow{\text{eindeutig}}$ $\delta: Q \times \Sigma \rightarrow Q$ deterministisch

$\xrightarrow{\text{mehrdeutig}}$ $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$ nicht-deterministisch

→ $s \in Q$: *Startzustand*



Endliche Automaten

Definiert über einem Alphabet Σ

Bestehend aus:

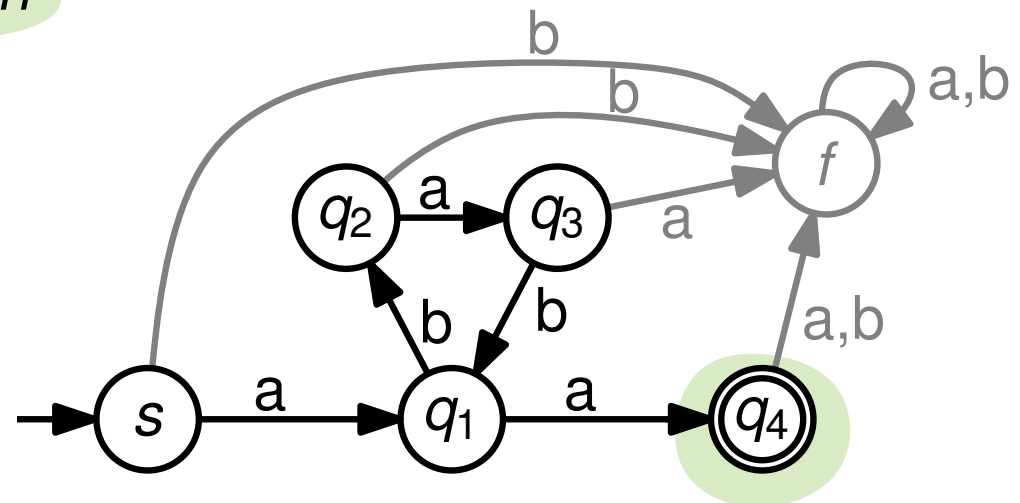
→ Q : endliche Menge von *Zuständen*

→ Übergänge $\xrightarrow{\text{eindeutig}}$ $\delta: Q \times \Sigma \rightarrow Q$ deterministisch

$\xrightarrow{\text{mehrdeutig}}$ $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$ nicht-deterministisch

→ $s \in Q$: *Startzustand*

→ $F \subseteq Q$, Menge von *Endzuständen*



Endliche Automaten

Definiert über einem Alphabet Σ

Bestehend aus:

→ Q : endliche Menge von *Zuständen*

→ Übergänge $\xrightarrow{\text{eindeutig}}$ $\delta: Q \times \Sigma \rightarrow Q$ deterministisch

$\xrightarrow{\text{mehrdeutig}}$ $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$ nicht-deterministisch

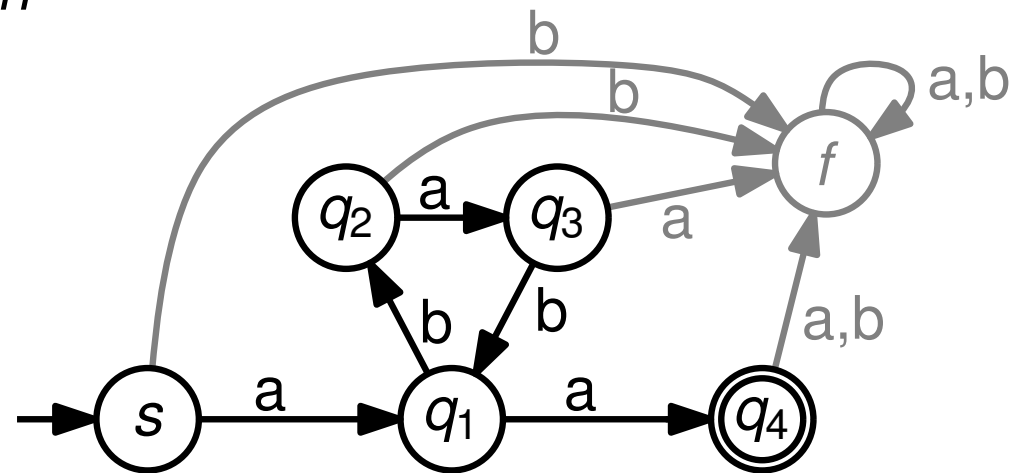
→ $s \in Q$: *Startzustand*

→ $F \subseteq Q$, Menge von *Endzuständen*

Akzeptanz des Wortes w :

DEA: Abarbeitung von w endet in Endzustand

NEA: Es gibt Abarbeitung von w , die in Endzustand endet



Endliche Automaten

Definiert über einem Alphabet Σ

Bestehend aus:

→ Q : endliche Menge von *Zuständen*

→ Übergänge $\xrightarrow{\text{eindeutig}}$ $\delta: Q \times \Sigma \rightarrow Q$ deterministisch

$\xrightarrow{\text{mehrdeutig}}$ $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$ nicht-deterministisch

→ $s \in Q$: *Startzustand*

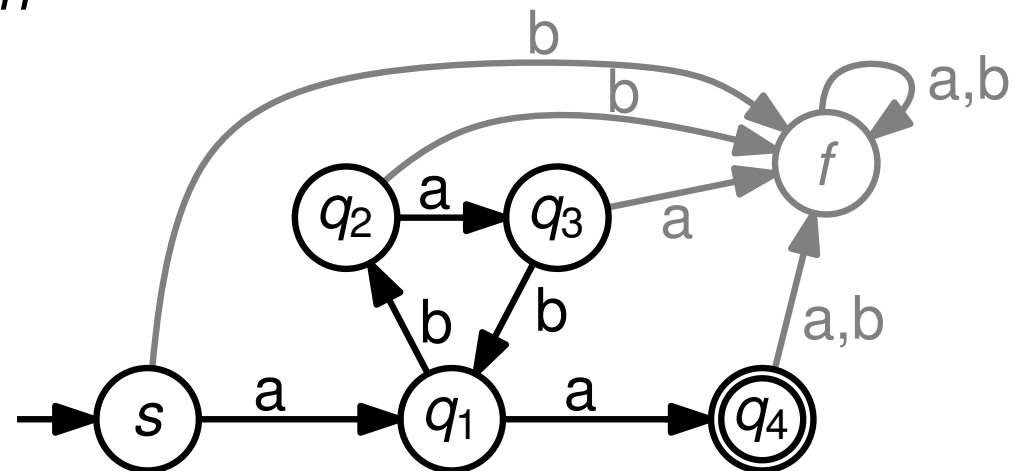
Welche Sprache erkennt der Automat?

→ $F \subseteq Q$, Menge von *Endzuständen*

Akzeptanz des Wortes w :

DEA: Abarbeitung von w endet in Endzustand

NEA: Es gibt Abarbeitung von w , die in Endzustand endet



Endliche Automaten

Definiert über einem Alphabet Σ

Bestehend aus:

→ Q : endliche Menge von *Zuständen*

→ Übergänge $\xrightarrow{\text{eindeutig}}$ $\delta: Q \times \Sigma \rightarrow Q$ deterministisch

$\xrightarrow{\text{mehrdeutig}}$ $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^Q$ nicht-deterministisch

→ $s \in Q$: *Startzustand*

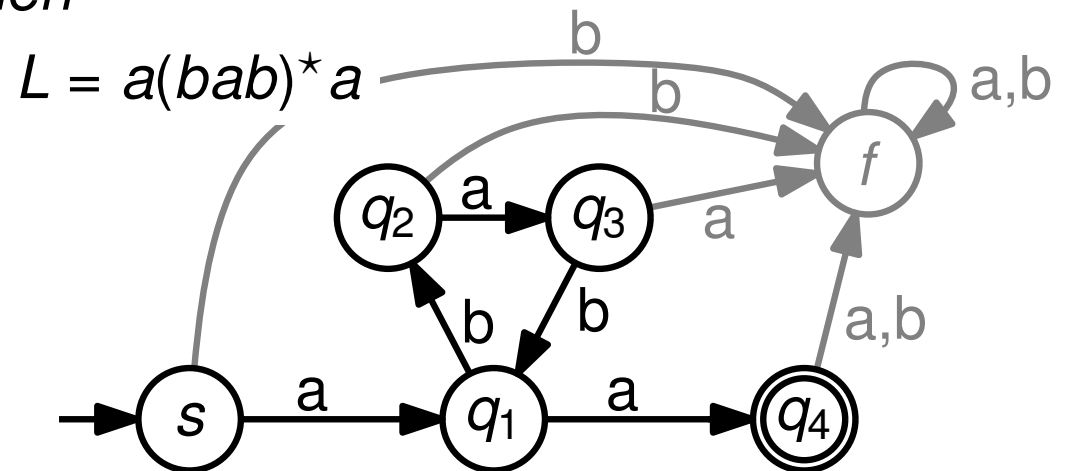
→ $F \subseteq Q$, Menge von *Endzuständen*

Akzeptanz des Wortes w :

DEA: Abarbeitung von w endet in Endzustand

NEA: Es gibt Abarbeitung von w , die in Endzustand endet

Welche Sprache erkennt der Automat?



Aufgabe

Konstruieren Sie einen nicht-deterministischen endlichen Automaten, der folgende Sprache erkennt:

$$L = (a \cup (ab(b)^*ba))^*$$

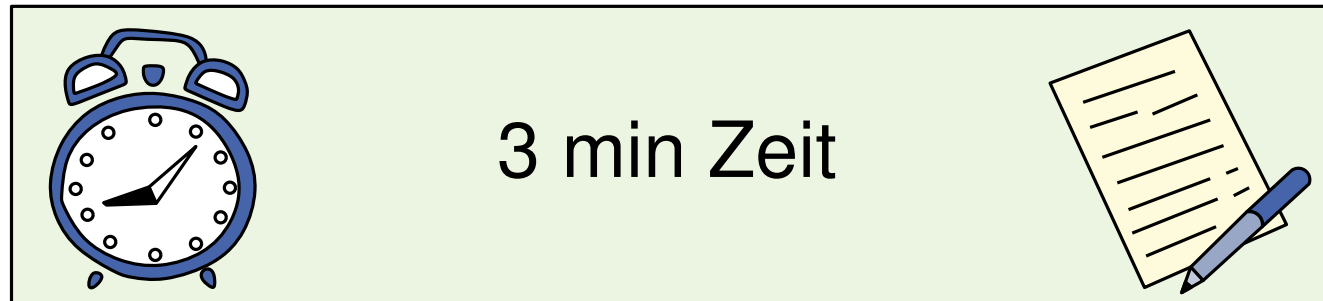
Verwenden Sie hierzu **maximal 3 Zustände + 1 Fehlerzustand**.

Aufgabe

Konstruieren Sie einen nicht-deterministischen endlichen Automaten, der folgende Sprache erkennt:

$$L = (a \cup (ab(b)^*ba))^*$$

Verwenden Sie hierzu **maximal 3 Zustände + 1 Fehlerzustand**.

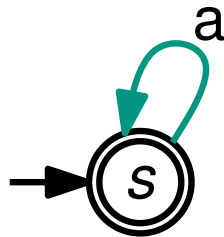


Aufgabe

Konstruieren Sie einen nicht-deterministischen endlichen Automaten, der folgende Sprache erkennt:

$$L = (a \cup (ab(b)^*ba))^*$$

Verwenden Sie hierzu **maximal 3 Zustände + 1 Fehlerzustand**.

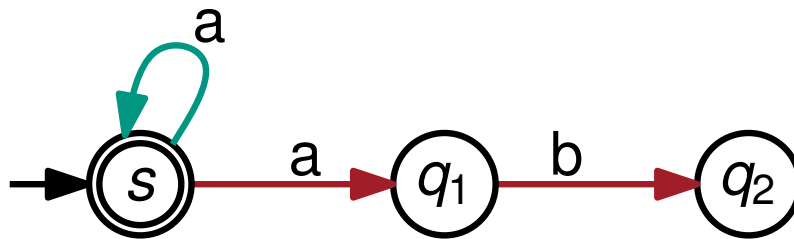


Aufgabe

Konstruieren Sie einen nicht-deterministischen endlichen Automaten, der folgende Sprache erkennt:

$$L = (\underline{a} \cup (\underline{ab}(b)^* \underline{ba}))^*$$

Verwenden Sie hierzu **maximal 3 Zustände + 1 Fehlerzustand**.

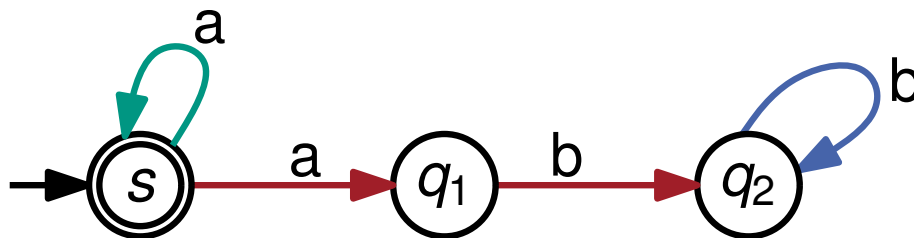


Aufgabe

Konstruieren Sie einen nicht-deterministischen endlichen Automaten, der folgende Sprache erkennt:

$$L = (\underline{a} \cup (\underline{ab} \underline{(b)^*} \underline{ba}))^*$$

Verwenden Sie hierzu **maximal 3 Zustände + 1 Fehlerzustand**.

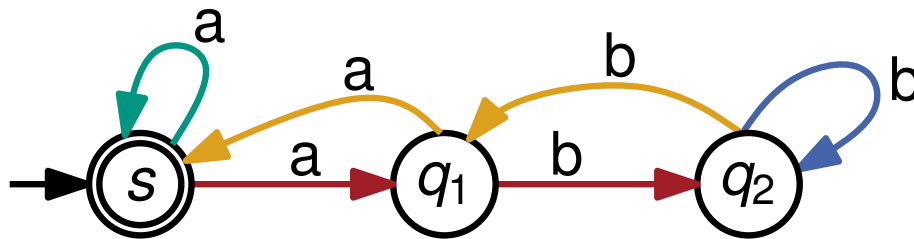


Aufgabe

Konstruieren Sie einen nicht-deterministischen endlichen Automaten, der folgende Sprache erkennt:

$$L = (\underline{a} \cup (\underline{ab}(b)^* \underline{ba}))^*$$

Verwenden Sie hierzu **maximal 3 Zustände + 1 Fehlerzustand**.

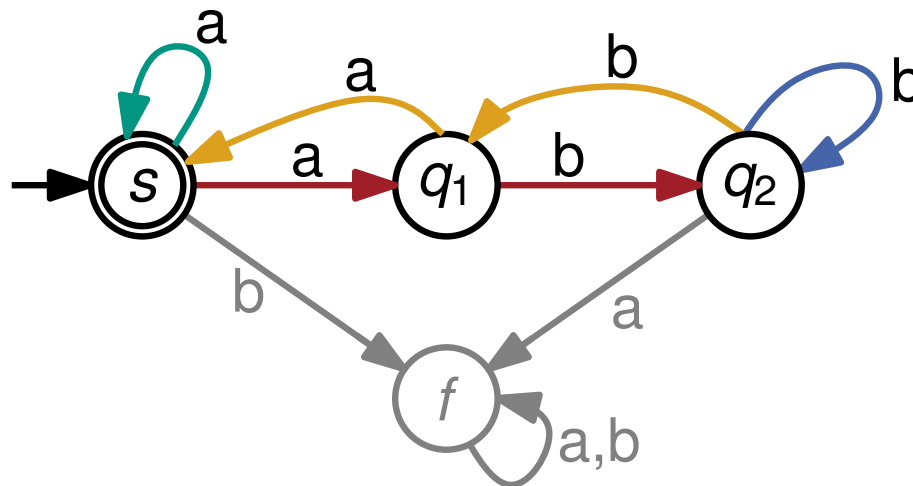


Aufgabe

Konstruieren Sie einen nicht-deterministischen endlichen Automaten, der folgende Sprache erkennt:

$$L = (\underline{a} \cup (\underline{ab}(b)^* \underline{ba}))^*$$

Verwenden Sie hierzu **maximal 3 Zustände + 1 Fehlerzustand**.



Größe von DEAs vs. NEAs

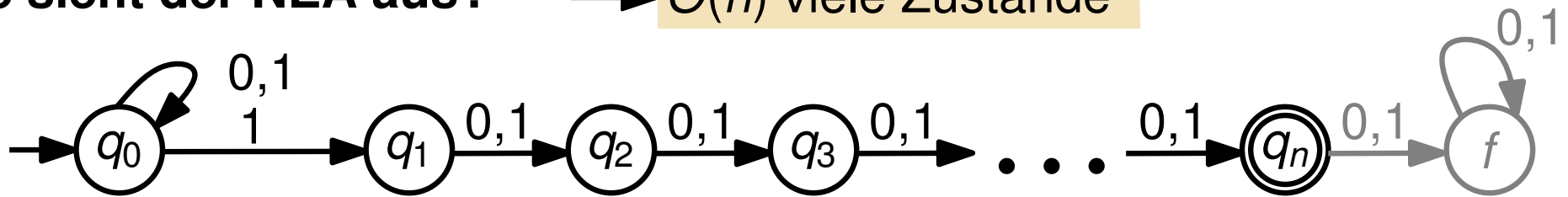
Betrachte: $L_n = \{w \in \{0, 1\}^* \mid \text{das } n\text{-letzte Zeichen von } w \text{ ist eine } 1\}$.

Wie sieht der NEA aus?

Größe von DEAs vs. NEAs

Betrachte: $L_n = \{w \in \{0, 1\}^* \mid \text{das } n\text{-letzte Zeichen von } w \text{ ist eine } 1\}$.

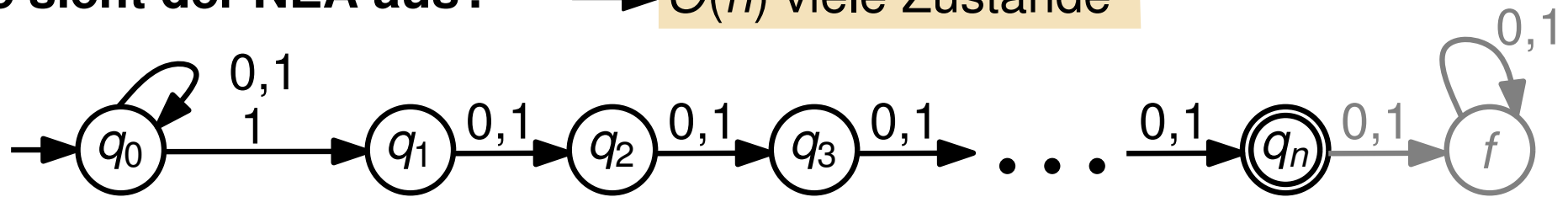
Wie sieht der NEA aus? $\rightarrow O(n)$ viele Zustände



Größe von DEAs vs. NEAs

Betrachte: $L_n = \{w \in \{0, 1\}^* \mid \text{das } n\text{-letzte Zeichen von } w \text{ ist eine } 1\}$.

Wie sieht der NEA aus? $\rightarrow O(n)$ viele Zustände

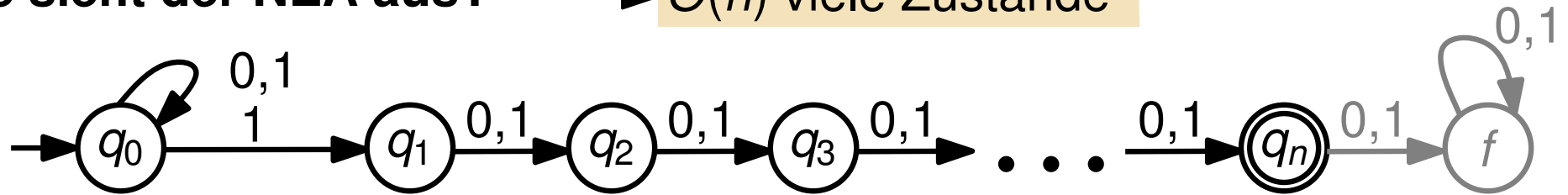


Zeige: Jeder DEA von L_n hat **mindestens 2^n Zustände.**

Größe von DEAs vs. NEAs

Betrachte: $L_n = \{w \in \{0, 1\}^* \mid \text{das } n\text{-letzte Zeichen von } w \text{ ist eine } 1\}$.

Wie sieht der NEA aus? $\rightarrow O(n)$ viele Zustände



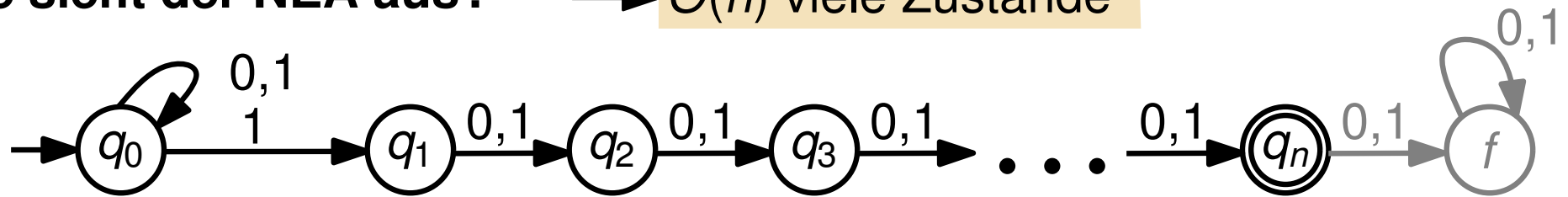
Zeige: Jeder DEA von L_n hat **mindestens 2^n Zustände**.

Annahme: Sei $\mathcal{D}_n = (\Sigma, Q, \delta, q_0, F)$ DEA, der L_n akzeptiert, mit $|Q| < 2^n$

Größe von DEAs vs. NEAs

Betrachte: $L_n = \{w \in \{0, 1\}^* \mid \text{das } n\text{-letzte Zeichen von } w \text{ ist eine } 1\}$.

Wie sieht der NEA aus? $\rightarrow O(n)$ viele Zustände



Zeige: Jeder DEA von L_n hat **mindestens 2^n Zustände**.

Annahme: Sei $\mathcal{D}_n = (\Sigma, Q, \delta, q_0, F)$ DEA, der L_n akzeptiert, mit $|Q| < 2^n$

Lemma: Wenn $|Q| < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

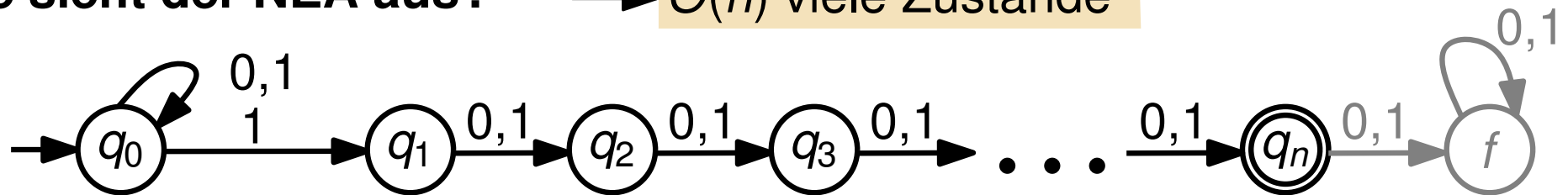
$$\delta(q_0, x 1 u) = \delta(q_0, y 0 v)$$

Noch zu zeigen.

Größe von DEAs vs. NEAs

Betrachte: $L_n = \{w \in \{0, 1\}^* \mid \text{das } n\text{-letzte Zeichen von } w \text{ ist eine } 1\}$.

Wie sieht der NEA aus? $\rightarrow O(n)$ viele Zustände



Zeige: Jeder DEA von L_n hat **mindestens 2^n Zustände**.

Annahme: Sei $\mathcal{D}_n = (\Sigma, Q, \delta, q_0, F)$ DEA, der L_n akzeptiert, mit $|Q| < 2^n$

Lemma: Wenn $|Q| < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x 1 u) = \delta(q_0, y 0 v) \quad \text{Noch zu zeigen.}$$

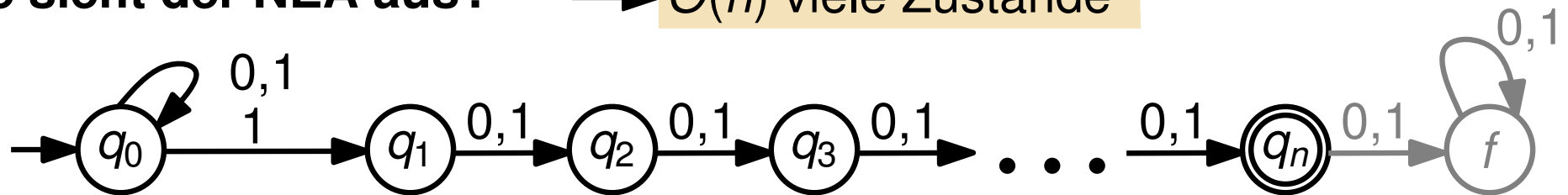
Dann gilt: Seien $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$ wie im Lemma gewählt.

$\rightarrow x 1 u \in L_n$ und $y 0 v \notin L_n$

Größe von DEAs vs. NEAs

Betrachte: $L_n = \{w \in \{0, 1\}^* \mid \text{das } n\text{-letzte Zeichen von } w \text{ ist eine } 1\}$.

Wie sieht der NEA aus? $\rightarrow O(n)$ viele Zustände



Zeige: Jeder DEA von L_n hat **mindestens 2^n Zustände**.


Annahme: Sei $\mathcal{D}_n = (\Sigma, Q, \delta, q_0, F)$ DEA, der L_n akzeptiert, mit $|Q| < 2^n$

Lemma: Wenn $|Q| < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x 1 u) = \delta(q_0, y 0 v) \quad \text{Noch zu zeigen.}$$

Dann gilt: Seien $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$ wie im Lemma gewählt.

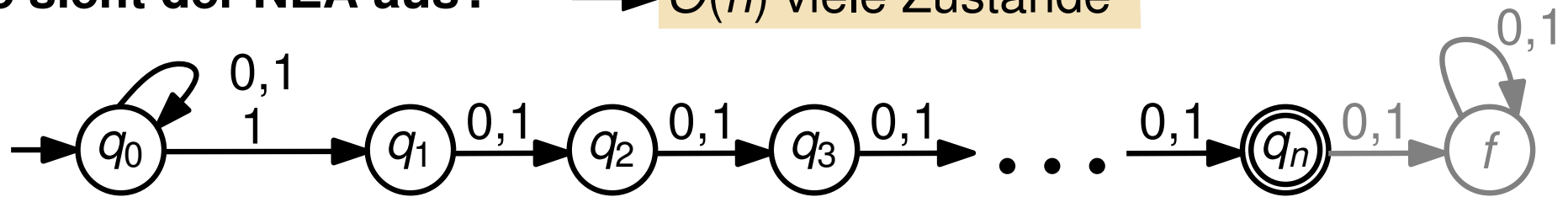
$\rightarrow x 1 u \in L_n$ und $y 0 v \notin L_n$

$\rightarrow \delta(q_0, x 1 u) \in F$ und $\delta(q_0, y 0 v) \notin F$  Lemma

Größe von DEAs vs. NEAs

Betrachte: $L_n = \{w \in \{0, 1\}^* \mid \text{das } n\text{-letzte Zeichen von } w \text{ ist eine } 1\}$.

Wie sieht der NEA aus? $\rightarrow O(n)$ viele Zustände



Zeige: Jeder DEA von L_n hat **mindestens 2^n Zustände**.


Annahme: Sei $\mathcal{D}_n = (\Sigma, Q, \delta, q_0, F)$ DEA, der L_n akzeptiert, mit $|Q| < 2^n$

Lemma: Wenn $|Q| < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x 1 u) = \delta(q_0, y 0 v) \quad \text{Noch zu zeigen.}$$

Dann gilt: Seien $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$ wie im Lemma gewählt.

$\rightarrow x 1 u \in L_n$ und $y 0 v \notin L_n$

$\rightarrow \delta(q_0, x 1 u) \in F$ und $\delta(q_0, y 0 v) \notin F$  Lemma

$\rightarrow \Omega(2^n)$ viele Zustände

Beweis zu Lemma

Lemma: Wenn $Q < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

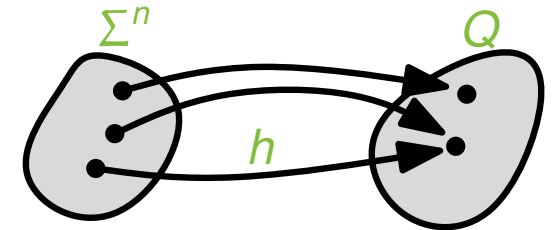
$$\delta(q_0, x 1 u) = \delta(q_0, y 0 v)$$

Beweis zu Lemma

Lemma: Wenn $Q < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x1u) = \delta(q_0, y0v)$$

Beweis: Sei $h: \Sigma^n \rightarrow Q$ sodass $h(z) = \delta(q_0, z)$
→ h ist nicht injektiv, da $|Q| < 2^n$



Beweis zu Lemma

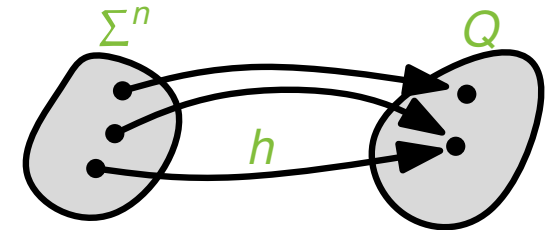
Lemma: Wenn $Q < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x1u) = \delta(q_0, y0v)$$

Beweis: Sei $h: \Sigma^n \rightarrow Q$ sodass $h(z) = \delta(q_0, z)$

→ h ist nicht injektiv, da $|Q| < 2^n$

∃ Sequenzen $\sigma_1 = a_1 \dots a_n$ und $\sigma_2 = b_1 \dots b_n$ mit $\sigma_1 \neq \sigma_2$ und $h(\sigma_1) = h(\sigma_2)$

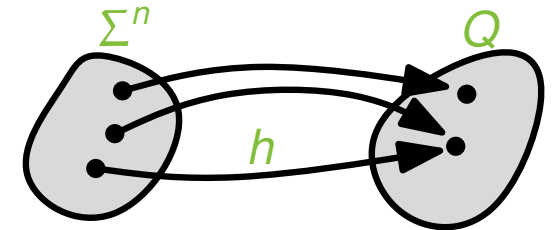


Beweis zu Lemma

Lemma: Wenn $Q < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x1u) = \delta(q_0, y0v)$$

Beweis: Sei $h: \Sigma^n \rightarrow Q$ sodass $h(z) = \delta(q_0, z)$
→ h ist nicht injektiv, da $|Q| < 2^n$



\exists Sequenzen $\sigma_1 = a_1 \dots a_n$ und $\sigma_2 = b_1 \dots b_n$ mit $\sigma_1 \neq \sigma_2$ und $h(\sigma_1) = h(\sigma_2)$

Annahme: $a_i = 1$ und $b_i = 0$

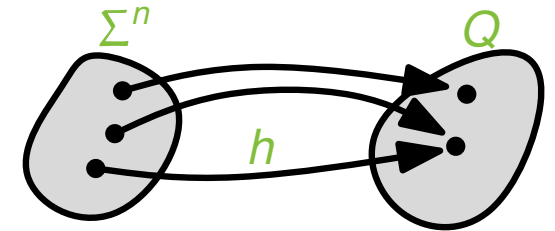
Sei $x = a_1 \dots a_{i-1}$ $u = a_{i+1} \dots a_n 0^{i-1}$ $y = b_1 \dots b_{i-1}$ $v = b_{i+1} \dots b_n 0^{i-1}$

Beweis zu Lemma

Lemma: Wenn $Q < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x 1 u) = \delta(q_0, y 0 v)$$

Beweis: Sei $h: \Sigma^n \rightarrow Q$ sodass $h(z) = \delta(q_0, z)$
 $\rightarrow h$ ist nicht injektiv, da $|Q| < 2^n$



\exists Sequenzen $\sigma_1 = a_1 \dots a_n$ und $\sigma_2 = b_1 \dots b_n$ mit $\sigma_1 \neq \sigma_2$ und $h(\sigma_1) = h(\sigma_2)$

Annahme: $a_i = 1$ und $b_i = 0$

Sei $x = a_1 \dots a_{i-1}$ $u = a_{i+1} \dots a_n 0^{i-1}$ $y = b_1 \dots b_{i-1}$ $v = b_{i+1} \dots b_n 0^{i-1}$

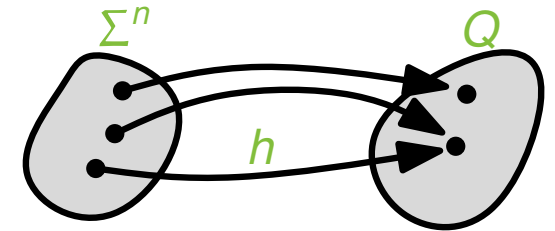
$\delta(q_0, x 1 u) = \delta(q_0, a_1 \dots a_{i-1} 1 a_{i+1} \dots a_n 0^{i-1})$

Beweis zu Lemma

Lemma: Wenn $Q < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x 1 u) = \delta(q_0, y 0 v)$$

Beweis: Sei $h: \Sigma^n \rightarrow Q$ sodass $h(z) = \delta(q_0, z)$
 $\rightarrow h$ ist nicht injektiv, da $|Q| < 2^n$



\exists Sequenzen $\sigma_1 = a_1 \dots a_n$ und $\sigma_2 = b_1 \dots b_n$ mit $\sigma_1 \neq \sigma_2$ und $h(\sigma_1) = h(\sigma_2)$

Annahme: $a_i = 1$ und $b_i = 0$

Sei $x = a_1 \dots a_{i-1}$ $u = a_{i+1} \dots a_n 0^{i-1}$ $y = b_1 \dots b_{i-1}$ $v = b_{i+1} \dots b_n 0^{i-1}$

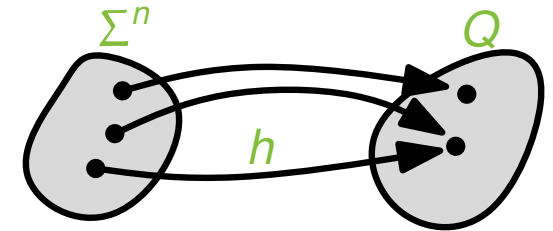
$$\begin{aligned} \delta(q_0, x 1 u) &= \delta(q_0, a_1 \dots a_{i-1} 1 a_{i+1} \dots a_n 0^{i-1}) \\ &= \delta(\delta(q_0, a_1 \dots a_{i-1} 1 a_{i+1} \dots a_n), 0^{i-1}) \end{aligned}$$

Beweis zu Lemma

Lemma: Wenn $Q < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x 1 u) = \delta(q_0, y 0 v)$$

Beweis: Sei $h: \Sigma^n \rightarrow Q$ sodass $h(z) = \delta(q_0, z)$
→ h ist nicht injektiv, da $|Q| < 2^n$



∃ Sequenzen $\sigma_1 = a_1 \dots a_n$ und $\sigma_2 = b_1 \dots b_n$ mit $\sigma_1 \neq \sigma_2$ und $h(\sigma_1) = h(\sigma_2)$

Annahme: $a_i = 1$ und $b_i = 0$

Sei $x = a_1 \dots a_{i-1}$ $u = a_{i+1} \dots a_n 0^{i-1}$ $y = b_1 \dots b_{i-1}$ $v = b_{i+1} \dots b_n 0^{i-1}$

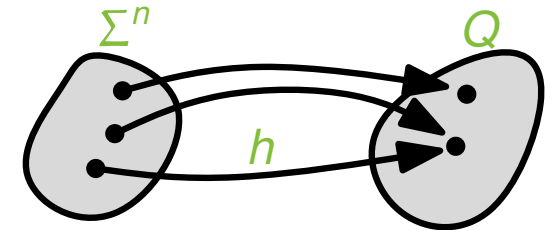
$$\begin{aligned} \delta(q_0, x 1 u) &= \delta(q_0, a_1 \dots a_{i-1} 1 a_{i+1} \dots a_n 0^{i-1}) \\ &= \delta(\delta(q_0, a_1 \dots a_{i-1} 1 a_{i+1} \dots a_n), 0^{i-1}) \\ &= \delta(\delta(q_0, b_1 \dots b_{i-1} 0 b_{i+1} \dots b_n), 0^{i-1}) \end{aligned}$$

Beweis zu Lemma

Lemma: Wenn $Q < 2^n$, dann gibt es $x, y \in \Sigma^*$ und $u, v \in \Sigma^{n-1}$, sodass

$$\delta(q_0, x 1 u) = \delta(q_0, y 0 v)$$

Beweis: Sei $h: \Sigma^n \rightarrow Q$ sodass $h(z) = \delta(q_0, z)$
 $\rightarrow h$ ist nicht injektiv, da $|Q| < 2^n$



\exists Sequenzen $\sigma_1 = a_1 \dots a_n$ und $\sigma_2 = b_1 \dots b_n$ mit $\sigma_1 \neq \sigma_2$ und $h(\sigma_1) = h(\sigma_2)$

Annahme: $a_i = 1$ und $b_i = 0$

Sei $x = a_1 \dots a_{i-1}$ $u = a_{i+1} \dots a_n 0^{i-1}$ $y = b_1 \dots b_{i-1}$ $v = b_{i+1} \dots b_n 0^{i-1}$

$$\begin{aligned} \delta(q_0, x 1 u) &= \delta(q_0, a_1 \dots a_{i-1} 1 a_{i+1} \dots a_n 0^{i-1}) \\ &= \delta(\delta(q_0, a_1 \dots a_{i-1} 1 a_{i+1} \dots a_n), 0^{i-1}) \\ &= \delta(\delta(q_0, b_1 \dots b_{i-1} 0 b_{i+1} \dots b_n), 0^{i-1}) \\ &= \delta(q_0, b_1 \dots b_{i-1} 0 b_{i+1} \dots b_n 0^{i-1}) = \delta(q_0, y 0 v) \end{aligned}$$

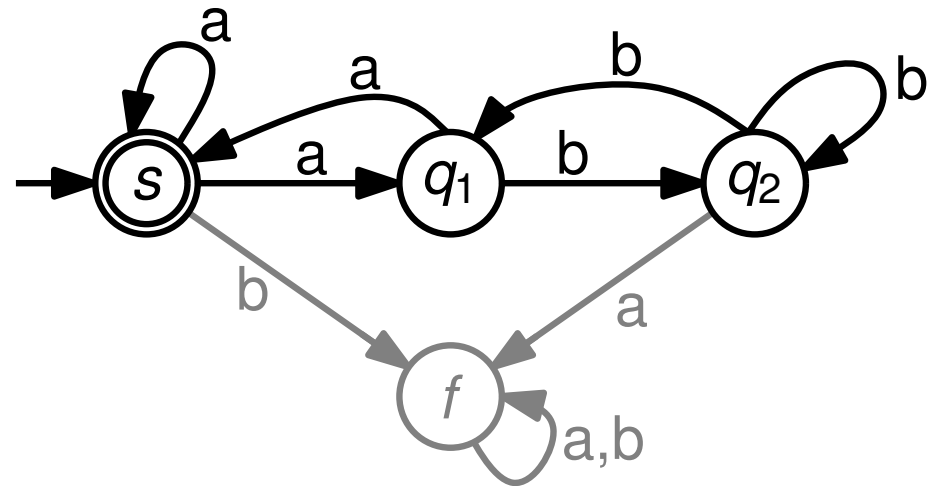
Potenzmengenkonstruktion

Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b

$$L = (a \cup (ab(b)^*ba))^*$$



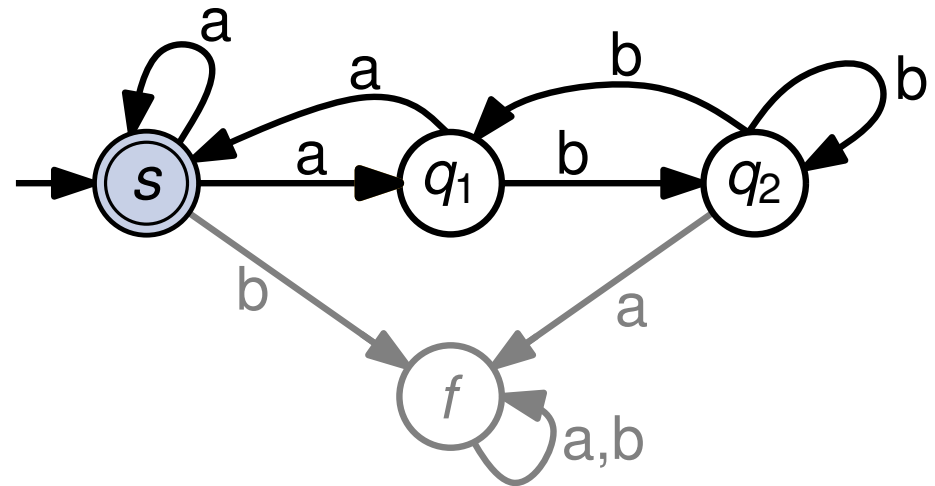
Potenzmengenkonstruktion

Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b
{s}		

$$L = (a \cup (ab(b)^*ba))^*$$



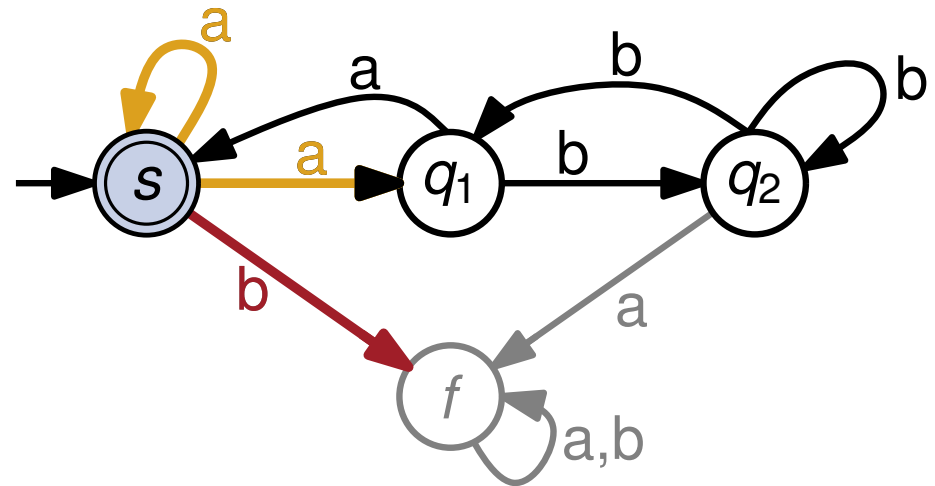
Potenzmengenkonstruktion

Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b
$\{s\}$	$\{s, q_1\}$	$\{f\}$
$\{s, q_1\}$		
$\{f\}$		

$$L = (a \cup (ab(b)^*ba))^*$$



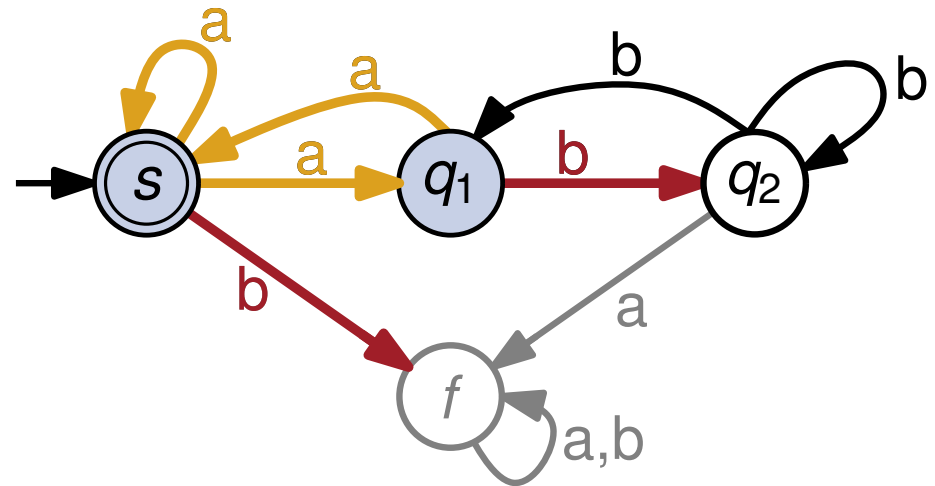
Potenzmengenkonstruktion

Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b
{s}	{s, q ₁ }	{f}
{s, q ₁ }	{s, q ₁ }	{f, q ₂ }
{f}		
{f, q ₂ }		

$$L = (a \cup (ab(b)^*ba))^*$$



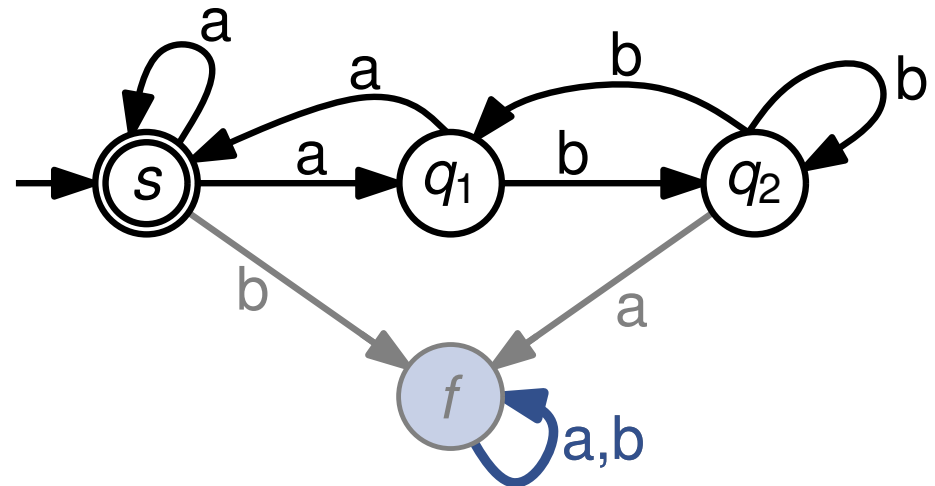
Potenzmengenkonstruktion

Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b
{s}	{s, q ₁ }	{f}
{s, q ₁ }	{s, q ₁ }	{f, q ₂ }
{f}	{f}	{f}
{f, q ₂ }		

$$L = (a \cup (ab(b)^*ba))^*$$



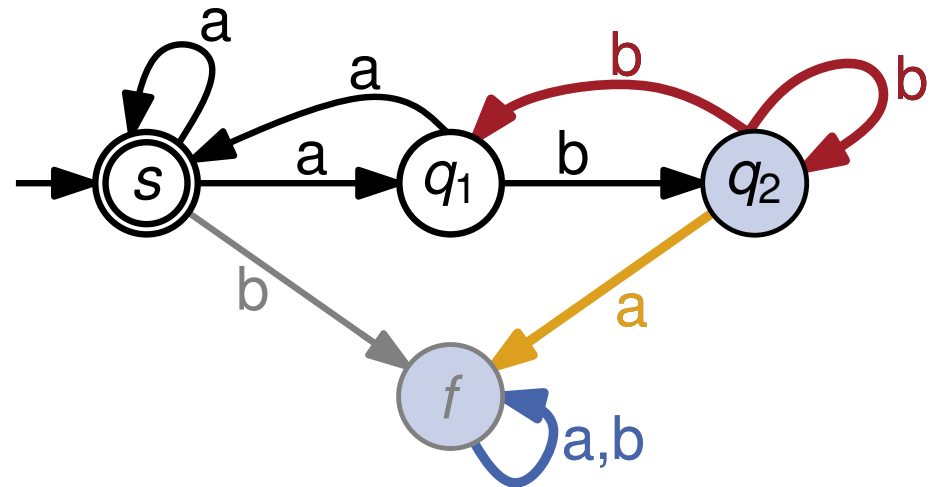
Potenzmengenkonstruktion

Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b
{s}	{s, q ₁ }	{f}
{s, q ₁ }	{s, q ₁ }	{f, q ₂ }
{f}	{f}	{f}
{f, q ₂ }	{f}	{f, q ₁ , q ₂ }
{f, q ₁ , q ₂ }		

$$L = (a \cup (ab(b)^*ba))^*$$



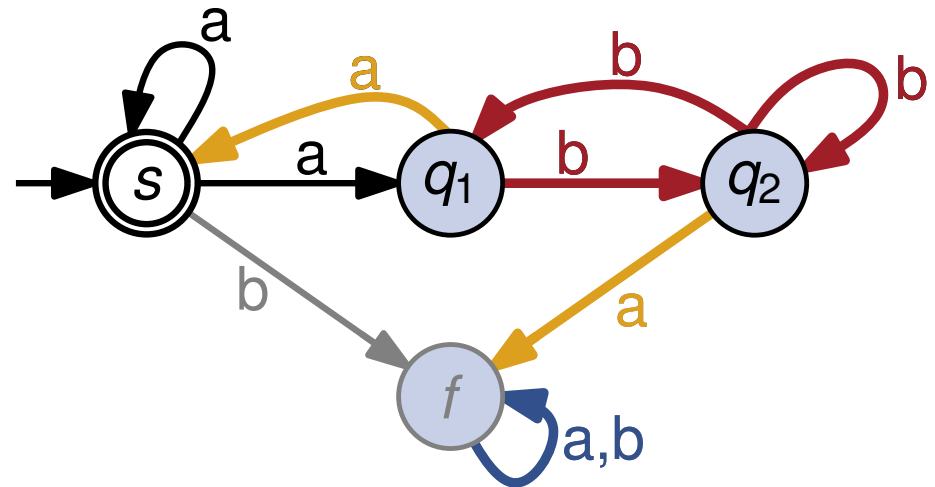
Potenzmengenkonstruktion

Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b
{s}	{s, q ₁ }	{f}
{s, q ₁ }	{s, q ₁ }	{f, q ₂ }
{f}	{f}	{f}
{f, q ₂ }	{f}	{f, q ₁ , q ₂ }
{f, q ₁ , q ₂ }	{f, s}	{f, q ₁ , q ₂ }
{f, s}		

$$L = (a \cup (ab(b)^*ba))^*$$



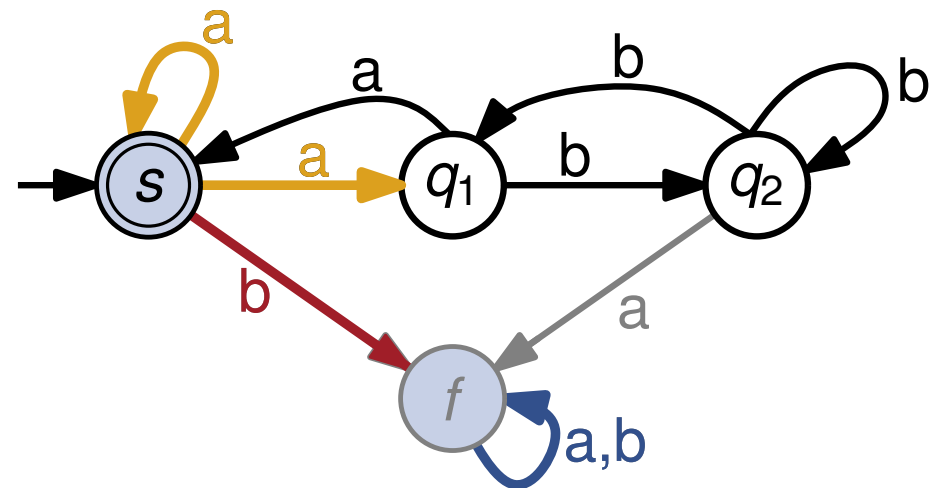
Potenzmengenkonstruktion

Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b
{s}	{s, q ₁ }	{f}
{s, q ₁ }	{s, q ₁ }	{f, q ₂ }
{f}	{f}	{f}
{f, q ₂ }	{f}	{f, q ₁ , q ₂ }
{f, q ₁ , q ₂ }	{f, s}	{f, q ₁ , q ₂ }
{f, s}	{f, s, q ₁ }	{f}
{f, s, q ₁ }		

$$L = (a \cup (ab(b)^*ba))^*$$



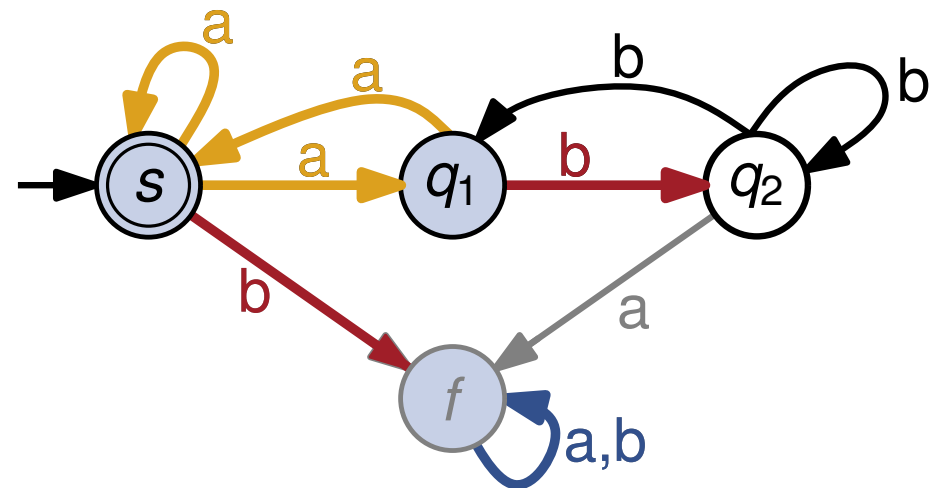
Potenzmengenkonstruktion

Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b
{s}	{s, q ₁ }	{f}
{s, q ₁ }	{s, q ₁ }	{f, q ₂ }
{f}	{f}	{f}
{f, q ₂ }	{f}	{f, q ₁ , q ₂ }
{f, q ₁ , q ₂ }	{f, s}	{f, q ₁ , q ₂ }
{f, s}	{f, s, q ₁ }	{f}
{f, s, q ₁ }	{f, s, q ₁ }	{f, q ₂ }

$$L = (a \cup (ab(b)^*ba))^*$$



Potenzmengenkonstruktion

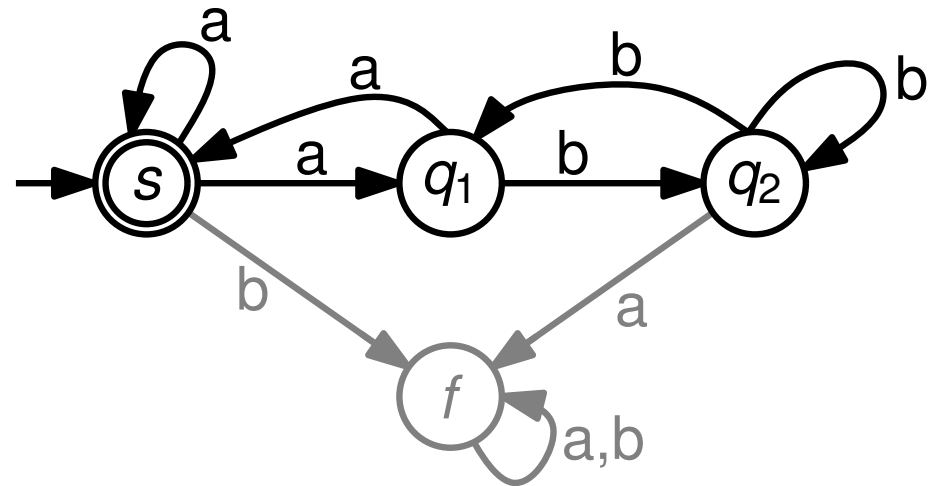
Jeder NEA kann in einen DEA überführt werden.

Potenzmengenkonstruktion: systematisches Verfahren für Transformation

Zustand	Übergang	
	a	b
$\{\underline{s}\}$	$\{s, q_1\}$	$\{f\}$
$\{\underline{s}, q_1\}$	$\{s, q_1\}$	$\{f, q_2\}$
$\{f\}$	$\{f\}$	$\{f\}$
$\{f, q_2\}$	$\{f\}$	$\{f, q_1, q_2\}$
$\{f, q_1, q_2\}$	$\{f, s\}$	$\{f, q_1, q_2\}$
$\{f, \underline{s}\}$	$\{f, s, q_1\}$	$\{f\}$
$\{f, \underline{s}, q_1\}$	$\{f, s, q_1\}$	$\{f, q_2\}$

$$L = (a \cup (ab(b)^*ba))^*$$

Endzustände: Alle Zustände, die Endzustand aus NEA enthalten.



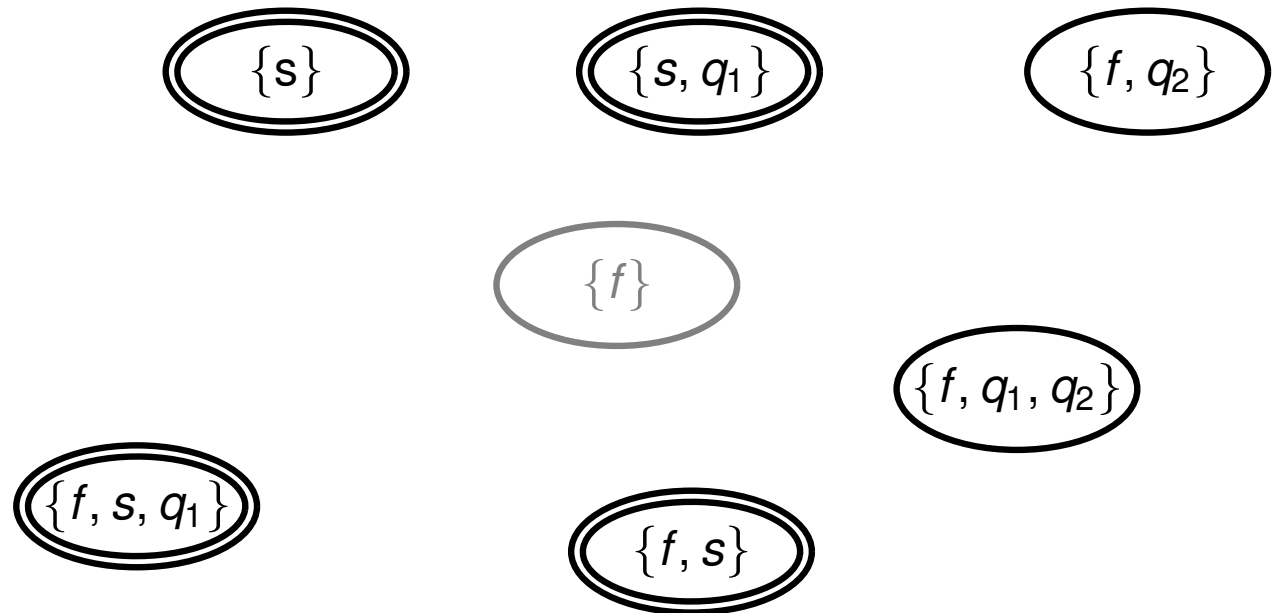
Potenzmengenkonstruktion

Zustand	Übergang	
	a	b
{s}	{s, q ₁ }	{f}
{s, q ₁ }	{s, q ₁ }	{f, q ₂ }
{f}	{f}	{f}
{f, q ₂ }	{f}	{f, q ₁ , q ₂ }
{f, q ₁ , q ₂ }	{f, s}	{f, q ₁ , q ₂ }
{f, s}	{f, s, q ₁ }	{f}
{f, s, q ₁ }	{f, s, q ₁ }	{f, q ₂ }

$$L = (a \cup (ab(b)^*ba))^*$$

Potenzmengenkonstruktion

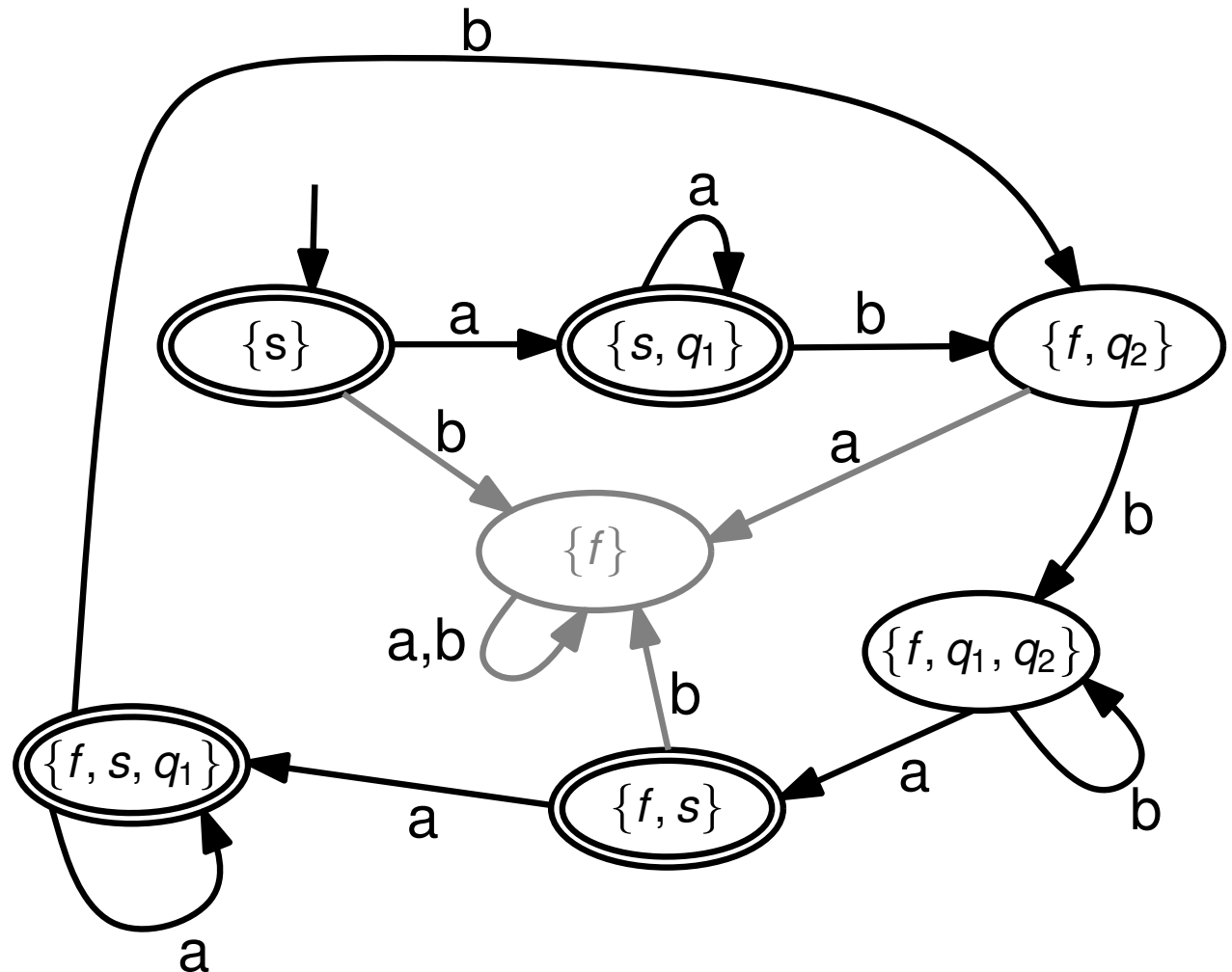
Zustand	Übergang	
	a	b
{s}	{s, q ₁ }	{f}
{s, q ₁ }	{s, q ₁ }	{f, q ₂ }
{f}	{f}	{f}
{f, q ₂ }	{f}	{f, q ₁ , q ₂ }
{f, q ₁ , q ₂ }	{f, s}	{f, q ₁ , q ₂ }
{f, s}	{f, s, q ₁ }	{f}
{f, s, q ₁ }	{f, s, q ₁ }	{f, q ₂ }



$$L = (a \cup (ab(b)^*ba))^*$$

Potenzmengenkonstruktion

Zustand	Übergang	
	a	b
$\{s\}$	$\{s, q_1\}$	$\{f\}$
$\{s, q_1\}$	$\{s, q_1\}$	$\{f, q_2\}$
$\{f\}$	$\{f\}$	$\{f\}$
$\{f, q_2\}$	$\{f\}$	$\{f, q_1, q_2\}$
$\{f, q_1, q_2\}$	$\{f, s\}$	$\{f, q_1, q_2\}$
$\{f, s\}$	$\{f, s, q_1\}$	$\{f\}$
$\{f, s, q_1\}$	$\{f, s, q_1\}$	$\{f, q_2\}$



$$L = (a \cup (ab(b)^*ba))^*$$

ϵ -Abschluss

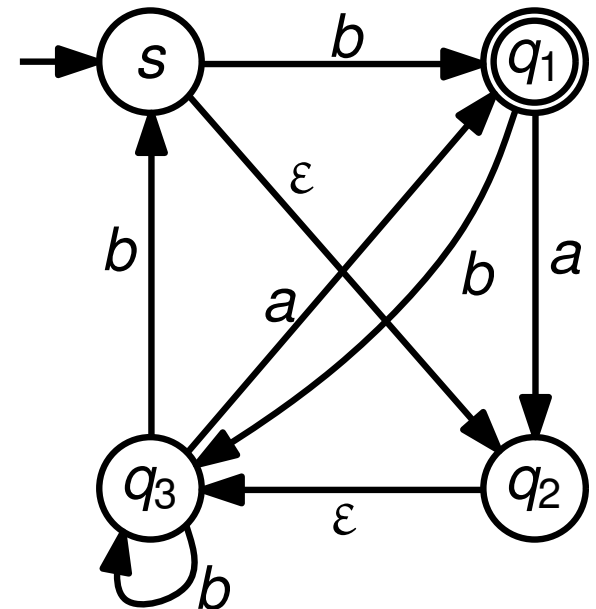
Jeder NEA kann in einen äquivalenten NEA ohne ϵ -Übergänge überführt werden, ohne die Anzahl Zustände zu erhöhen.

Verfahren: ϵ -Abschlüsse berechnen.

Idee: Finde via " $\epsilon^* X \epsilon^*$ " erreichbare Zustände

ϵ -Abschluss

	a	b
s		
q ₁		
q ₂		
q ₃		



Fehlende Übergänge führen in Fehlerzustand.

ϵ -Abschluss

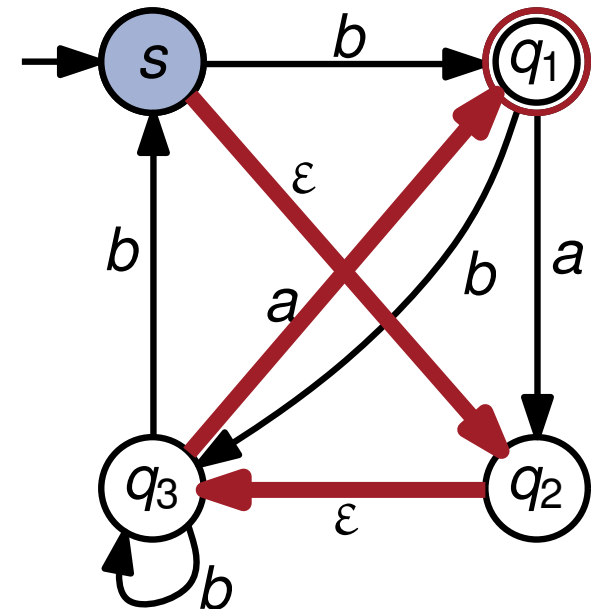
Jeder NEA kann in einen äquivalenten NEA ohne ϵ -Übergänge überführt werden, ohne die Anzahl Zustände zu erhöhen.

Verfahren: ϵ -Abschlüsse berechnen.

Idee: Finde via " $\epsilon^* X \epsilon^*$ " erreichbare Zustände

ϵ -Abschluss

	a	b
s	q_1	
q_1		
q_2		
q_3		



Fehlende Übergänge führen in Fehlerzustand.

ϵ -Abschluss

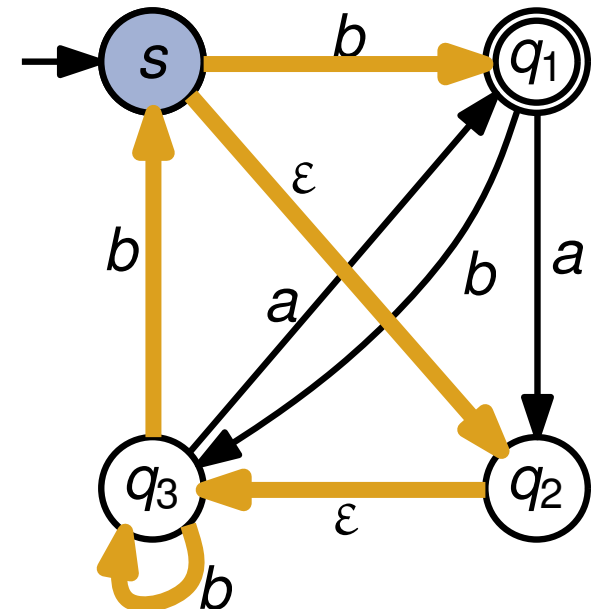
Jeder NEA kann in einen äquivalenten NEA ohne ϵ -Übergänge überführt werden, ohne die Anzahl Zustände zu erhöhen.

Verfahren: ϵ -Abschlüsse berechnen.

Idee: Finde via " $\epsilon^* X \epsilon^*$ " erreichbare Zustände

ϵ -Abschluss

	a	b
s	q_1	s, q_1, q_2, q_3
q_1		
q_2		
q_3		



Fehlende Übergänge führen in Fehlerzustand.

ϵ -Abschluss

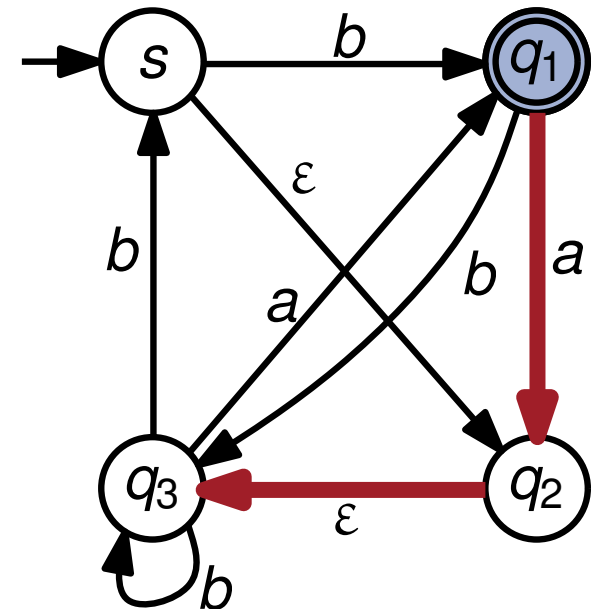
Jeder NEA kann in einen äquivalenten NEA ohne ϵ -Übergänge überführt werden, ohne die Anzahl Zustände zu erhöhen.

Verfahren: ϵ -Abschlüsse berechnen.

Idee: Finde via " $\epsilon^* X \epsilon^*$ " erreichbare Zustände

ϵ -Abschluss

	a	b
s	q_1	s, q_1, q_2, q_3
q_1	q_2, q_3	
q_2		
q_3		



Fehlende Übergänge führen in Fehlerzustand.

ϵ -Abschluss

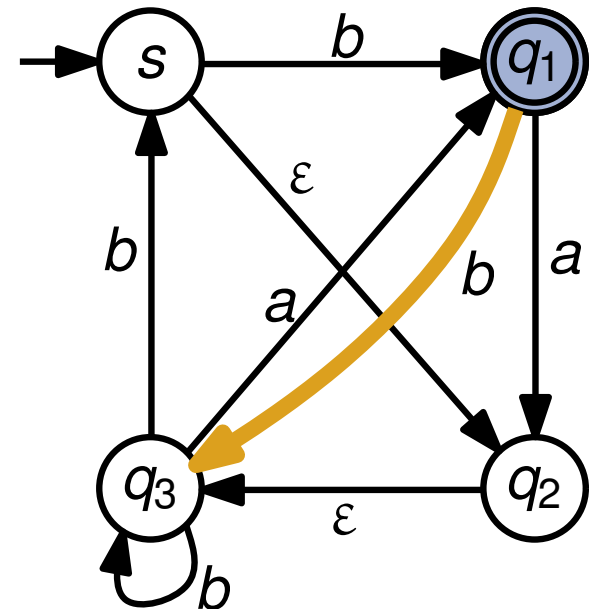
Jeder NEA kann in einen äquivalenten NEA ohne ϵ -Übergänge überführt werden, ohne die Anzahl Zustände zu erhöhen.

Verfahren: ϵ -Abschlüsse berechnen.

Idee: Finde via " $\epsilon^* X \epsilon^*$ " erreichbare Zustände

ϵ -Abschluss

	a	b
s	q_1	s, q_1, q_2, q_3
q_1	q_2, q_3	$, q_3$
q_2		
q_3		



Fehlende Übergänge führen in Fehlerzustand.

ϵ -Abschluss

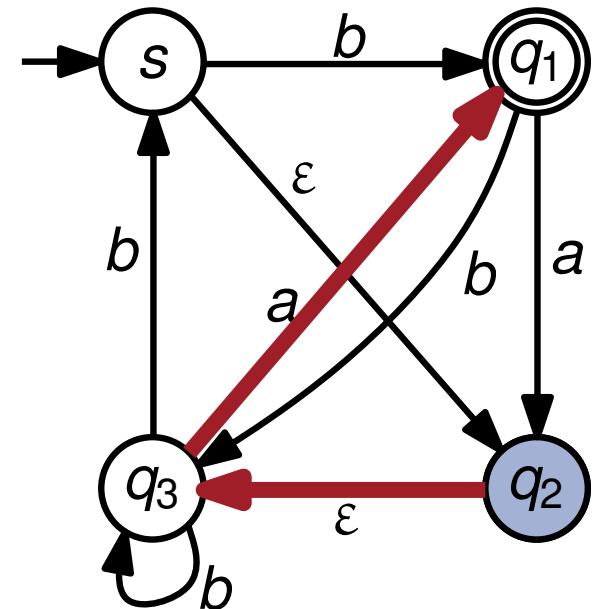
Jeder NEA kann in einen äquivalenten NEA ohne ϵ -Übergänge überführt werden, ohne die Anzahl Zustände zu erhöhen.

Verfahren: ϵ -Abschlüsse berechnen.

Idee: Finde via " $\epsilon^* X \epsilon^*$ " erreichbare Zustände

ϵ -Abschluss

	a	b
s	q_1	s, q_1, q_2, q_3
q_1	q_2, q_3	$, q_3$
q_2	q_1	
q_3		



Fehlende Übergänge führen in Fehlerzustand.

ϵ -Abschluss

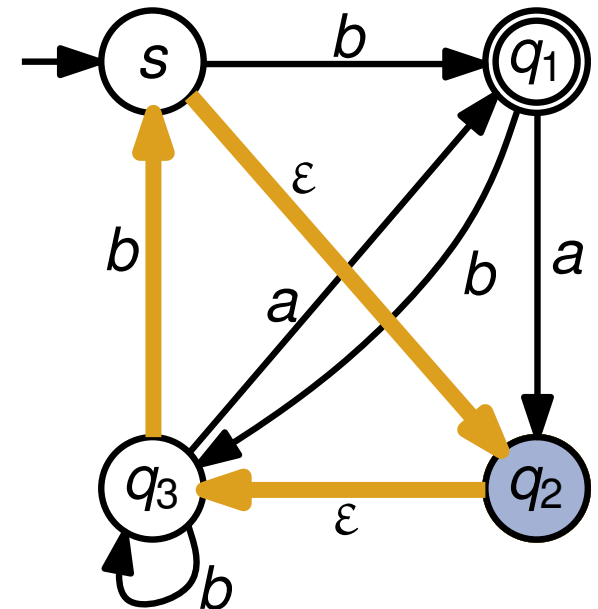
Jeder NEA kann in einen äquivalenten NEA ohne ϵ -Übergänge überführt werden, ohne die Anzahl Zustände zu erhöhen.

Verfahren: ϵ -Abschlüsse berechnen.

Idee: Finde via " $\epsilon^* X \epsilon^*$ " erreichbare Zustände

ϵ -Abschluss

	a	b
s	q_1	s, q_1, q_2, q_3
q_1	q_2, q_3	$, q_3$
q_2	q_1	s, q_2, q_3
q_3		



Fehlende Übergänge führen in Fehlerzustand.

ϵ -Abschluss

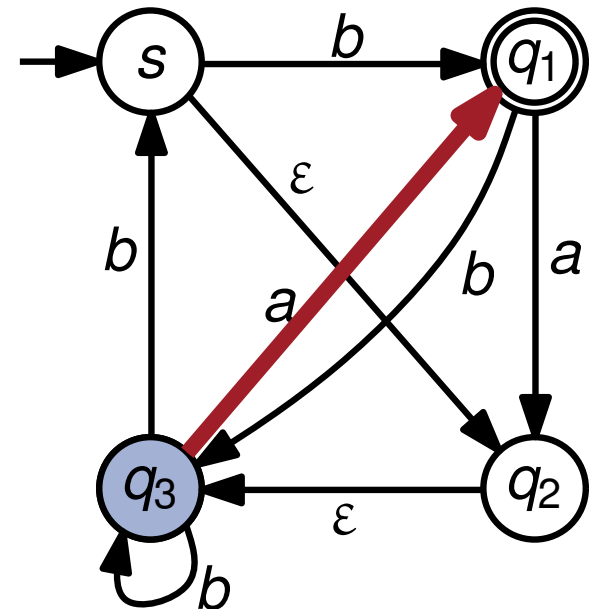
Jeder NEA kann in einen äquivalenten NEA ohne ϵ -Übergänge überführt werden, ohne die Anzahl Zustände zu erhöhen.

Verfahren: ϵ -Abschlüsse berechnen.

Idee: Finde via " $\epsilon^* X \epsilon^*$ " erreichbare Zustände

ϵ -Abschluss

	a	b
s	q_1	s, q_1, q_2, q_3
q_1	q_2, q_3	$, q_3$
q_2	q_1	s, q_2, q_3
q_3	q_1	



Fehlende Übergänge führen in Fehlerzustand.

ϵ -Abschluss

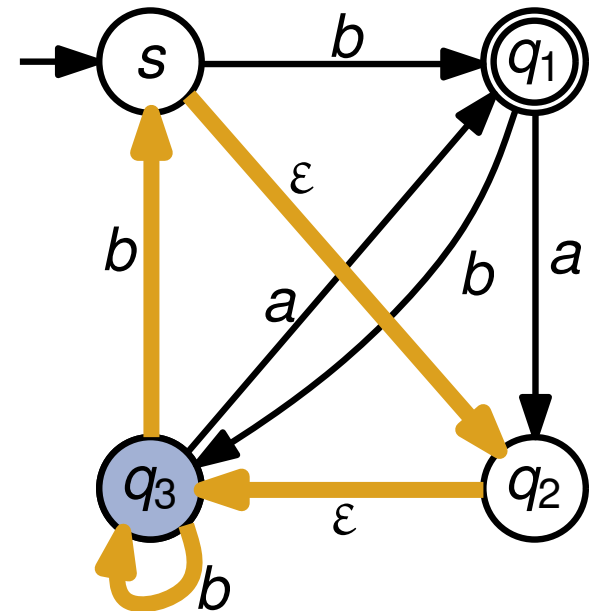
Jeder NEA kann in einen äquivalenten NEA ohne ϵ -Übergänge überführt werden, ohne die Anzahl Zustände zu erhöhen.

Verfahren: ϵ -Abschlüsse berechnen.

Idee: Finde via " $\epsilon^* X \epsilon^*$ " erreichbare Zustände

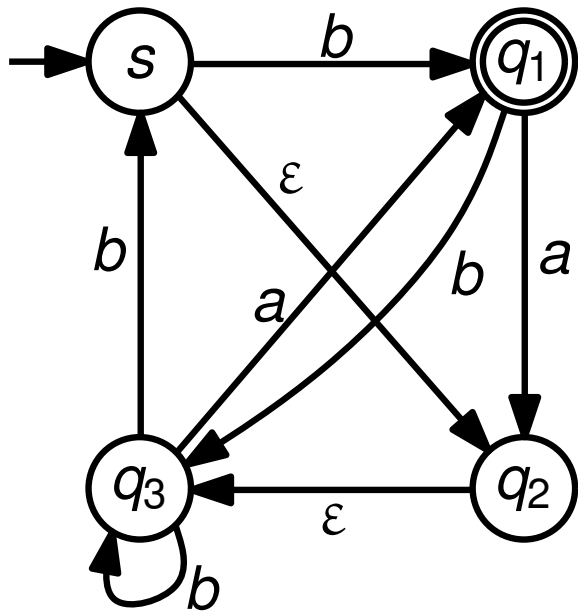
ϵ -Abschluss

	a	b
s	q_1	s, q_1, q_2, q_3
q_1	q_2, q_3	$, q_3$
q_2	q_1	s, q_2, q_3
q_3	q_1	s, q_2, q_3



Fehlende Übergänge führen in Fehlerzustand.

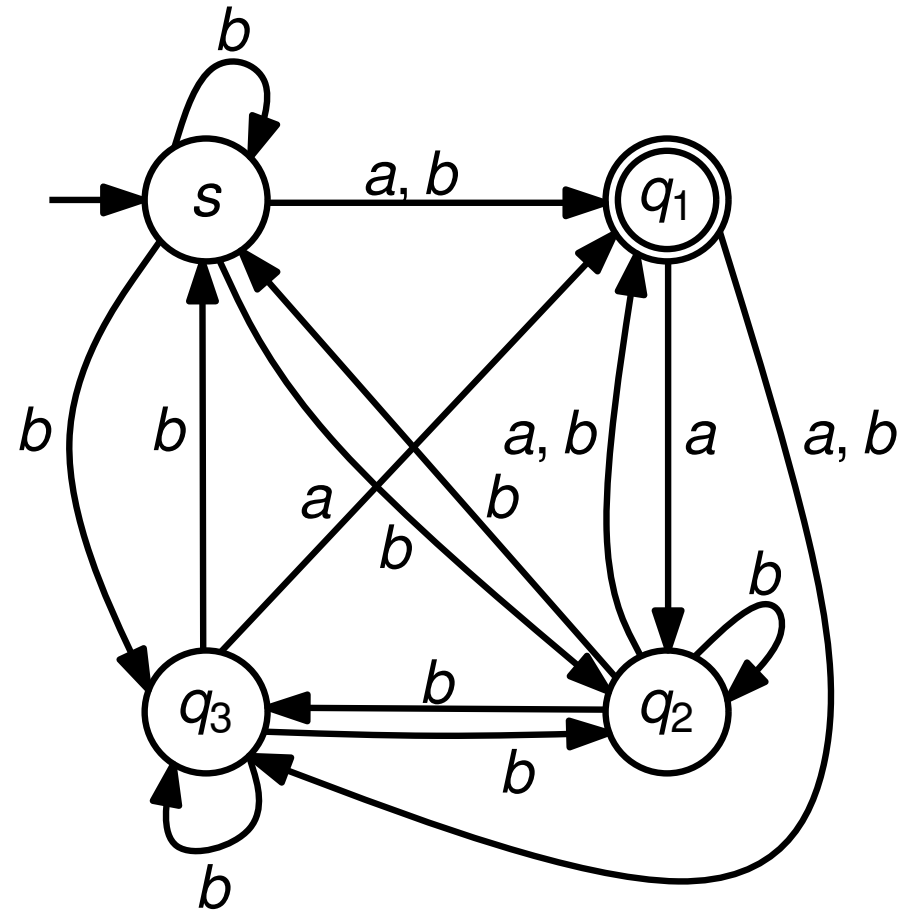
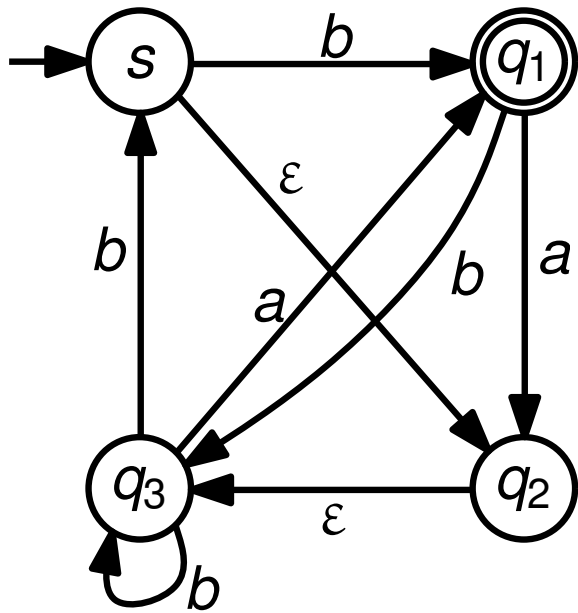
ε -Abschluss



ε -Abschluss

	a	b
s	q ₁	s, q ₁ , q ₂ , q ₃
q ₁	q ₂ , q ₃	q ₃
q ₂	q ₁	s, q ₂ , q ₃
q ₃	q ₁	s, q ₂ , q ₃

ϵ -Abschluss



ϵ -Abschluss

	a	b
s	q ₁	s, q ₁ , q ₂ , q ₃
q ₁	q ₂ , q ₃	q ₃
q ₂	q ₁	s, q ₂ , q ₃
q ₃	q ₁	s, q ₂ , q ₃

Kontextfreie Grammatiken

Kontextfreie Grammatik: $G = (\Sigma, V, S, R)$

Alphabet: Σ (Terminalalphabet), endlich

$$\Sigma = \{a, b\}$$

Variablen: V (Nichtterminale), $V \cap \Sigma = \emptyset$, endlich

$$V = \{A, S\}$$

Startsymbol: $S \in V$

Ableitungsregeln: $R \subset V \times (\Sigma \cup V)^*$

$$S \rightarrow aSb \mid A, \quad A \rightarrow aA \mid a$$

Kontextfreie Grammatiken

$$G = (\Sigma = \{a, b\}, V = \{S\}, S, R)$$

$$R = S \rightarrow aSb \mid ab$$

$$L(G) = \{a^i b^i \mid i \in \mathbb{N}\}$$

Kontextfreie Grammatiken

$$G = (\Sigma = \{a, b\}, V = \{S\}, S, R)$$

$$R = S \rightarrow aSb \mid ab$$

$$L(G) = \{a^i b^i \mid i \in \mathbb{N}\}$$

Ist $L(G)$ regulär?

Kontextfreie Grammatiken

$$G = (\Sigma = \{a, b\}, V = \{S\}, S, R)$$

$$R = S \rightarrow aSb \mid ab$$

$$L(G) = \{a^i b^i \mid i \in \mathbb{N}\}$$

Ist $L(G)$ regulär?

Nein!

Kontextfreie Grammatiken

$$G = (\Sigma = \{a, b\}, V = \{S\}, S, R)$$

$$R = S \rightarrow aSb \mid ab$$

$$L(G) = \{a^i b^i \mid i \in \mathbb{N}\}$$

Ist $L(G)$ regulär?

Nein!

Problem (informell): EA können nicht zählen (endliches Gedächtnis)

Kontextfreie Grammatiken

$$G = (\Sigma = \{a, b\}, V = \{S\}, S, R)$$

$$R = S \rightarrow aSb \mid ab$$

$$L(G) = \{a^i b^i \mid i \in \mathbb{N}\}$$

Ist $L(G)$ regulär?

Nein!

Problem (informell): EA können nicht zählen (endliches Gedächtnis)

Formales Hilfsmittel:

Satz (Pumping-Lemma).

Zu jeder regulären Sprache L existiert eine Zahl $n \in \mathbb{N}$, so dass jedes Wort $w \in L$ mit $|w| > n$ zerlegt werden kann in $w = uvx$ mit $|v|, |x| \leq n$, $v \neq \varepsilon$, so dass auch $uv^i x \in L$ für alle $i = 0, 1, \dots$

$$ux \in L \quad uvx \in L \quad uvvx \in L \quad uvvvx \in L \quad \dots$$