Script generated by TTT

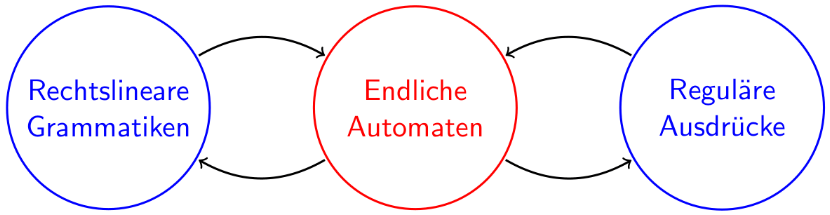
Title: Nipkow: Theo (29.04.2019)

Date: Mon Apr 29 14:06:15 CEST 2019

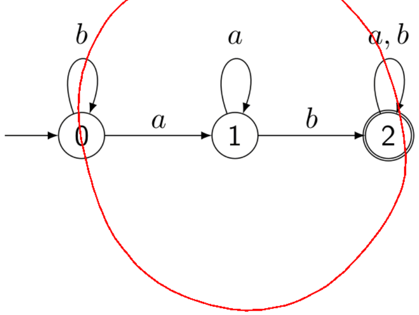
Duration: 95:07 min

Pages: 99

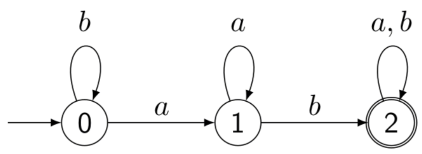
3. Reguläre Sprachen



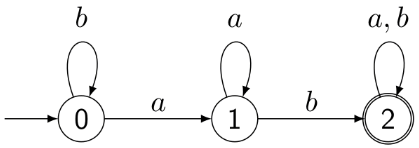
3.1 Deterministische endliche Automaten



3.1 Deterministische endliche Automaten



3.1 Deterministische endliche Automaten



Eingabewort $baba \rightsquigarrow$ Zustandsfolge 0,0,1,2,2.

Definition 3.1

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von Zuständen Q,

Definition 3.1

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von Zuständen Q,
- einem (endlichen) Eingabealphabet Σ,
- einer (totalen!) Übergangsfunktion $\delta : Q \times \Sigma \to Q$,
- einem Startzustand $q_0 \in Q$, und

Definition 3.1

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von Zuständen Q,
- einem (endlichen) Eingabealphabet Σ,
- einer (totalen!) Übergangsfunktion $\delta : Q \times \Sigma \to Q$,
- einem Startzustand $q_0 \in Q$, und
- einer Menge $F \subseteq Q$ von Endzuständen (akzeptierenden Zust.)

Die von M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\} ,$$

Definition 3.1

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von Zuständen Q,
- einem (endlichen) Eingabealphabet Σ,
- einer (totalen!) Übergangsfunktion $\delta : Q \times \Sigma \to Q$,
- einem Startzustand $q_0 \in Q$, und
- einer Menge $F \subseteq Q$ von Endzuständen (akzeptierenden Zust.)

Die von M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\} ,$$

wobei $\hat{\delta} : Q \times \Sigma^* \to Q$ rekursiv definiert ist durch

$$\begin{aligned} \hat{\delta}(q, \epsilon) &= q \\ \hat{\delta}(q, aw) &= \hat{\delta}(\delta(q, a), w) \quad \text{für } a \in \Sigma, w \in \Sigma^* \end{aligned}$$

Definition 3.1

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von Zuständen Q,
- einem (endlichen) Eingabealphabet Σ,
- einer (totalen!) Übergangsfunktion $\delta : Q \times \Sigma \to Q$,
- einem Startzustand $q_0 \in Q$, und
- einer Menge $F \subseteq Q$ von Endzuständen (akzeptierenden Zust.)

Die von M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\} ,$$

wobei $\hat{\delta} : Q \times \Sigma^* \to Q$ rekursiv definiert ist durch

$$\begin{aligned} \hat{\delta}(q, \epsilon) &= q \\ \hat{\delta}(q, aw) &= \hat{\delta}(\delta(q, a), w) \quad \text{für } a \in \Sigma, w \in \Sigma^* \end{aligned}$$

Definition 3.2

Eine Sprache ist regulär gdw sie von einem DFA akzeptiert wird.

Eigenschaften von $\hat{\delta}$:

- $\hat{\delta}(q, a) = \delta(q, a)$.

Definition 3.1

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von Zuständen Q,
- einem (endlichen) Eingabealphabet Σ,
- einer (totalen!) Übergangsfunktion $\delta : Q \times \Sigma \to Q$,
- einem Startzustand $q_0 \in Q$, und
- einer Menge $F \subseteq Q$ von Endzuständen (akzeptierenden Zust.)

Die von M akzeptierte Sprache ist

$$L(M) := \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\} ,$$

wobei $\hat{\delta} : Q \times \Sigma^* \to Q$ rekursiv definiert ist durch

$$\begin{aligned} \hat{\delta}(q, \epsilon) &= q \\ \hat{\delta}(q, aw) &= \hat{\delta}(\delta(q, a), w) \quad \text{für } a \in \Sigma, w \in \Sigma^* \end{aligned}$$

$\hat{\delta}(q, a) =$

$\hat{\delta}(q, a\varepsilon) =$

$\hat{\delta}(\underbrace{\delta(q, a)}, \varepsilon) =$

$\hat{\delta}(q, a)$

Definition 3.2

Eine Sprache ist regulär gdw sie von einem DFA akzeptiert wird.

Ind. nach Länge von w

$\hat{\delta}(q, aww') = \hat{\delta}(\delta(q,a), ww') =$ I.A.

$\hat{\delta}(\hat{\delta}(\delta(q,a), w), w')$

Eigenschaften von $\hat{\delta}$:

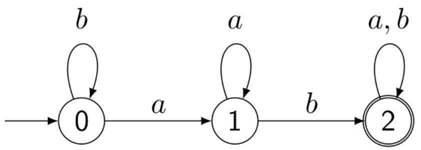
- $\hat{\delta}(q, a) = \delta(q, a)$.

$= \hat{\delta}(\hat{\delta}(q, aw), w')$

Graphische Darstellung

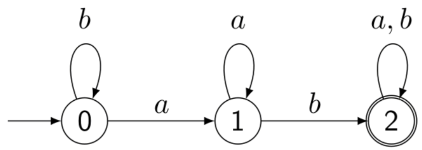
- Endliche Automaten können durch gerichtete und markierte Zustandsgraphen veranschaulicht werden:

Beispiel 3.3



Akzeptierte Sprache = alle Wörter über $\{a, b\}$, die ab enthalten.

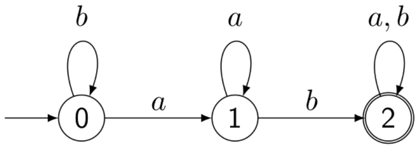
Beispiel 3.3



Akzeptierte Sprache = alle Wörter über $\{a, b\}$, die ab enthalten.

- Jedes Wort, das akzeptiert wird, enthält ab.

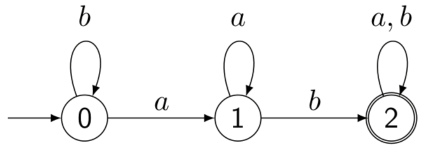
Beispiel 3.3



Akzeptierte Sprache = alle Wörter über $\{a, b\}$, die ab enthalten.

- Jedes Wort, das akzeptiert wird, enthält ab.
  Sei w ein Wort, welches akzeptiert wird.

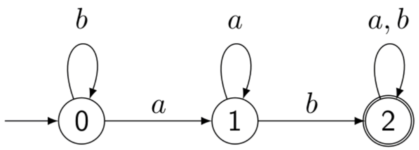
Beispiel 3.3



Akzeptierte Sprache = alle Wörter über $\{a, b\}$, die ab enthalten.

- Jedes Wort, das akzeptiert wird, enthält ab.
  Sei w ein Wort, welches akzeptiert wird.
  Betrachte in der Zustandsfolge für w den Zeitpunkt, in dem Zustand 1 zum letzten Mal besucht wird (existiert weil die Folge in Zustand 2 endet).

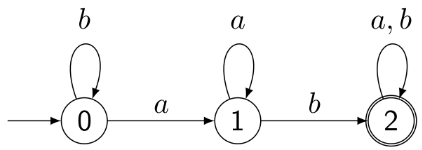
Beispiel 3.3



Akzeptierte Sprache = alle Wörter über $\{a, b\}$, die ab enthalten.

- Jedes Wort, das akzeptiert wird, enthält ab.
  Sei w ein Wort, welches akzeptiert wird.
  Betrachte in der Zustandsfolge für w den Zeitpunkt, in dem Zustand 1 zum letzten Mal besucht wird (existiert weil die Folge in Zustand 2 endet).
  Ummitelbar davor wird a gelesen und unmittelbar danach b

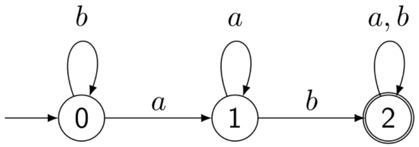
Beispiel 3.3



Akzeptierte Sprache = alle Wörter über $\{a, b\}$, die ab enthalten.

- Jedes Wort, das akzeptiert wird, enthält ab.
  Sei w ein Wort, welches akzeptiert wird.
  Betrachte in der Zustandsfolge für w den Zeitpunkt, in dem Zustand 1 zum letzten Mal besucht wird (existiert weil die Folge in Zustand 2 endet).
  Ummitelbar davor wird a gelesen und unmittelbar danach b
- Jedes Wort, das ab enthält, wird akzeptiert.

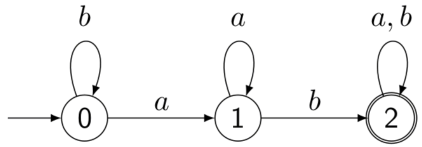
Beispiel 3.3



Akzeptierte Sprache = alle Wörter über $\{a, b\}$, die ab enthalten.

- Jedes Wort, das akzeptiert wird, enthält ab.
  Sei w ein Wort, welches akzeptiert wird.
  Betrachte in der Zustandsfolge für w den Zeitpunkt, in dem Zustand 1 zum letzten Mal besucht wird (existiert weil die Folge in Zustand 2 endet).
  Ummitelbar davor wird a gelesen und unmittelbar danach b
- Jedes Wort, das ab enthält, wird akzeptiert.
  Für alle $q \in \{0, 1, 2\}$ gilt: $\hat{\delta}(q, ab) = 2$.
  Für alle $w' \in \{a, b\}^*$ gilt: $\hat{\delta}(2, w') = 2$.

Beispiel 3.3



Akzeptierte Sprache = alle Wörter über $\{a, b\}$, die ab enthalten.

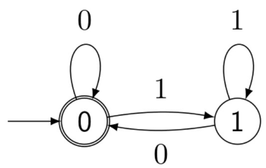
- Jedes Wort, das akzeptiert wird, enthält ab.
  Sei w ein Wort, welches akzeptiert wird.
  Betrachte in der Zustandsfolge für w den Zeitpunkt, in dem Zustand 1 zum letzten Mal besucht wird (existiert weil die Folge in Zustand 2 endet).
  Ummitelbar davor wird a gelesen und unmittelbar danach b
- Jedes Wort, das ab enthält, wird akzeptiert.
  Für alle $q \in \{0, 1, 2\}$ gilt: $\hat{\delta}(q, ab) = 2$.
  Für alle $w' \in \{a, b\}^*$ gilt: $\hat{\delta}(2, w') = 2$
  $\Longrightarrow \hat{\delta}(0, wabw') = \hat{\delta}(\hat{\delta}(\hat{\delta}(0, w), ab), w') = \hat{\delta}(2, w') = 2$

Beispiel 3.4

Für $w \in \{0, 1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl, zB $\#100 = 4$.

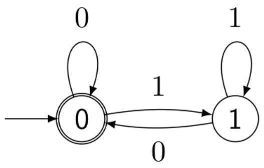
Beispiel 3.4

Für $w \in \{0, 1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl, zB $\#100 = 4$.



Beispiel 3.4

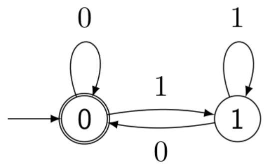
Für $w \in \{0,1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl,
zB $\#100 = 4$.



Der DFA akzeptiert genau die $w \in \{0,1\}^*$ so dass $\#w$ gerade ist.

Beispiel 3.4

Für $w \in \{0,1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl,
zB $\#100 = 4$.

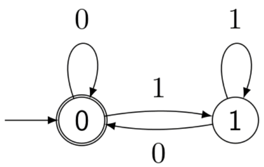


Der DFA akzeptiert genau die $w \in \{0,1\}^*$ so dass $\#w$ gerade ist.

Beweis:
Für alle $w \neq \epsilon$, $\#w$ ist gerade gdw w endet mit 0.

Beispiel 3.4

Für $w \in \{0,1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl,
zB $\#100 = 4$.

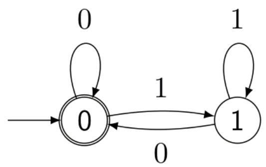


Der DFA akzeptiert genau die $w \in \{0,1\}^*$ so dass $\#w$ gerade ist.

Beweis:
Für alle $w \neq \epsilon$, $\#w$ ist gerade gdw w endet mit 0.
1. $\hat{\delta}(0, w0) =$

Beispiel 3.4

Für $w \in \{0,1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl,
zB $\#100 = 4$.

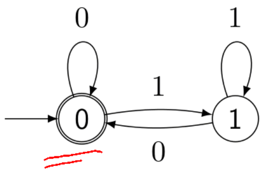


Der DFA akzeptiert genau die $w \in \{0,1\}^*$ so dass $\#w$ gerade ist.

Beweis:
Für alle $w \neq \epsilon$, $\#w$ ist gerade gdw w endet mit 0.
1. $\hat{\delta}(0, w0) = \delta(\hat{\delta}(0, w), 0)$

Beispiel 3.4

Für $w \in \{0,1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl,
zB $\#100 = 4$.

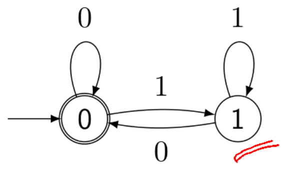


Der DFA akzeptiert genau die $w \in \{0,1\}^*$ so dass $\#w$ gerade ist.

Beweis:
Für alle $w \neq \epsilon$, $\#w$ ist gerade gdw w endet mit 0.
1. $\hat{\delta}(0, w0) = \delta(\hat{\delta}(0,w), 0) = 0 \in F$, daher $w0 \in L(A)$

Beispiel 3.4

Für $w \in \{0,1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl,
zB $\#100 = 4$.

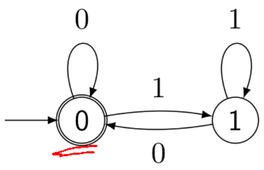


Der DFA akzeptiert genau die $w \in \{0,1\}^*$ so dass $\#w$ gerade ist.

Beweis:
Für alle $w \neq \epsilon$, $\#w$ ist gerade gdw w endet mit 0.
1. $\hat{\delta}(0, w0) = \delta(\hat{\delta}(0,w), 0) = 0 \in F$, daher $w0 \in L(A)$
2. $\hat{\delta}(0, w1) = \delta(\hat{\delta}(0,w), 1) = 1 \notin F$,

Beispiel 3.4

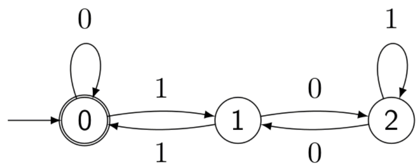
Für $w \in \{0,1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl,
zB $\#100 = 4$.



Der DFA akzeptiert genau die $w \in \{0,1\}^*$ so dass $\#w$ gerade ist.

Beweis:
Für alle $w \neq \epsilon$, $\#w$ ist gerade gdw w endet mit 0.
1. $\hat{\delta}(0, w0) = \delta(\hat{\delta}(0,w), 0) = 0 \in F$, daher $w0 \in L(A)$
2. $\hat{\delta}(0, w1) = \delta(\hat{\delta}(0,w), 1) = 1 \notin F$, daher $w1 \notin L(A)$

Beispiel 3.5



Der DFA akzeptiert genau die Wörter $w \in \{0,1\}^*$ mit: $\#w$ ist ein Vielfaches von 3.

3.2 Nichtdeterministische endliche Automaten (NFAs)

3.2 Nichtdeterministische endliche Automaten (NFAs)

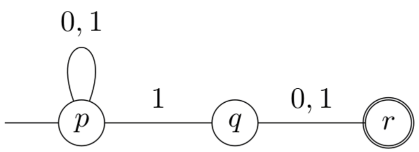
Verallgemeinerung von DFAs: Aus einem Zustand eines NFAs können $0, 1,$ oder mehr Transitionen mit derselben Beschriftung ausgehen.

3.2 Nichtdeterministische endliche Automaten (NFAs)

Verallgemeinerung von DFAs: Aus einem Zustand eines NFAs können $0, 1,$ oder mehr Transitionen mit derselben Beschriftung ausgehen.

Beispiel 3.6

Erkennung der Binärzahlen, deren vorletzte Ziffer 1 ist:

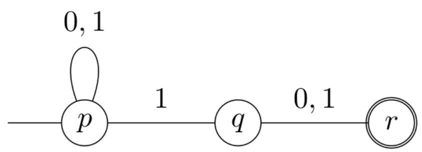


3.2 Nichtdeterministische endliche Automaten (NFAs)

Verallgemeinerung von DFAs: Aus einem Zustand eines NFAs können $0, 1,$ oder mehr Transitionen mit derselben Beschriftung ausgehen.

Beispiel 3.6

Erkennung der Binärzahlen, deren vorletzte Ziffer 1 ist:



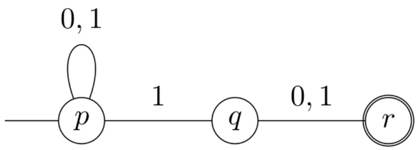
Wort wird akzepiert gdw es einen Weg zum Endzustand gibt.

3.2 Nichtdeterministische endliche Automaten (NFAs)

Verallgemeinerung von DFAs: Aus einem Zustand eines NFAs können $0, 1,$ oder mehr Transitionen mit derselben Beschriftung ausgehen.

Beispiel 3.6

Erkennung der Binärzahlen, deren vorlezte Ziffer 1 ist:



Wort wird akzepiert gdw es einen Weg zum Endzustand gibt.

Intuitive Vorstellung:

- Der Automat durchsucht alle möglichen Wege
- Der Automat "rät" den richtigen Weg.

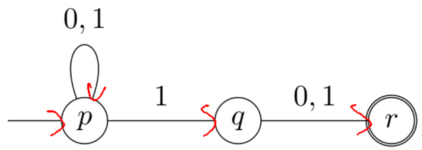
3.2 Nichtdeterministische endliche Automaten (NFAs)

Verallgemeinerung von DFAs: Aus einem Zustand eines NFAs können $0, 1,$ oder mehr Transitionen mit derselben Beschriftung ausgehen.

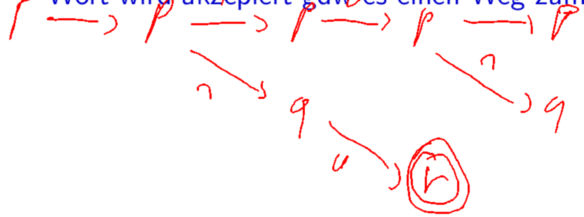
Beispiel 3.6

Erkennung der Binärzahlen, deren vorlezte Ziffer 1 ist:

0101



Wort wird akzepiert gdw es einen Weg zum Endzustand gibt.

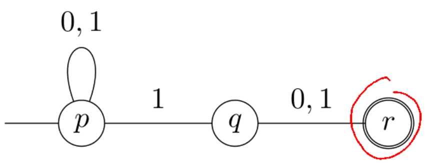


3.2 Nichtdeterministische endliche Automaten (NFAs)

Verallgemeinerung von DFAs: Aus einem Zustand eines NFAs können $0, 1,$ oder mehr Transitionen mit derselben Beschriftung ausgehen.

Beispiel 3.6

Erkennung der Binärzahlen, deren vorlezte Ziffer 1 ist:



Wort wird akzepiert gdw es einen Weg zum Endzustand gibt.

Definition 3.7

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

- Q, Σ, q_0 und F sind wie bei einem DFA

Definition 3.7

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

- Q, Σ, q_0 und F sind wie bei einem DFA
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$
  $\mathcal{P}(Q)$ = Menge aller Teilmengen von $Q = 2^Q$.

Definition 3.7

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

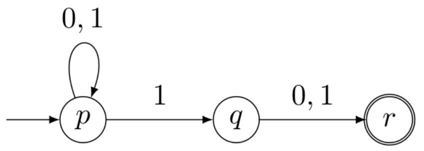
- Q, Σ, q_0 und F sind wie bei einem DFA
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$
  $\mathcal{P}(Q)$ = Menge aller Teilmengen von $Q = 2^Q$.
  Alternative: Relation $\delta \subseteq Q \times \Sigma \times Q$.

Definition 3.7

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

- Q, Σ, q_0 und F sind wie bei einem DFA
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$
  $\mathcal{P}(Q)$ = Menge aller Teilmengen von $Q = 2^Q$.
  Alternative: Relation $\delta \subseteq Q \times \Sigma \times Q$.

Beispiel

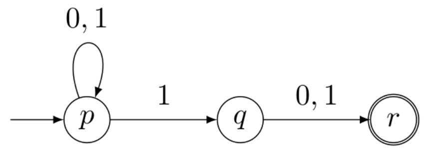


Definition 3.7

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

- Q, Σ, q_0 und F sind wie bei einem DFA
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$
  $\mathcal{P}(Q)$ = Menge aller Teilmengen von $Q = 2^Q$.
  Alternative: Relation $\delta \subseteq Q \times \Sigma \times Q$.

Beispiel



Σ

Q

δ	0	1
p	$\{p\}$	$\{p, q\}$
q	$\{r\}$	$\{r\}$
r	$\emptyset$	$\emptyset$

Erweiterung von $\delta : \quad Q \times \Sigma \to \mathcal{P}(Q)$
auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$:

Erweiterung von $\delta : \quad Q \times \Sigma \to \mathcal{P}(Q)$
auf $\bar{\delta} : \underline{\mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)}$:

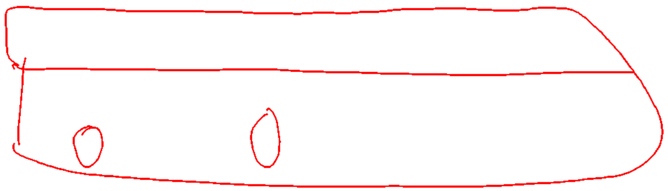
$$\bar{\delta}(S, a) := \bigcup_{q \in S} \delta(q, a)$$

Erweiterung von $\delta : \quad Q \times \Sigma \to \mathcal{P}(Q)$
auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$:

$$\bar{\delta}(S, a) := \bigcup_{q \in S} \delta(q, a)$$

$\Rightarrow \hat{\bar{\delta}} : \mathcal{P}(Q) \times \Sigma^* \to \mathcal{P}(Q)$

Erweiterung von $\delta : \quad Q \times \Sigma \to \mathcal{P}(Q)$
auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$:



Graphische Darstellung

- Endliche Automaten können durch gerichtete und markierte Zustandsgraphen veranschaulicht werden:

Erweiterung von $\delta : \quad Q \times \Sigma \to \mathcal{P}(Q)$
auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$:

$$\bar{\delta}(S, a) := \bigcup_{q \in S} \delta(q, a)$$

$\Rightarrow \hat{\bar{\delta}} : \mathcal{P}(Q) \times \Sigma^* \to \mathcal{P}(Q)$

$S \subseteq Q$

Intuition:

$\hat{\bar{\delta}}(S, w)$ ist Menge aller Zustände,
die sich von einem Zustand in S aus mit w erreichen lassen.

Erweiterung von $\delta : \quad Q \times \Sigma \to \mathcal{P}(Q)$
auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$:

$$\bar{\delta}(S, a) := \bigcup_{q \in S} \delta(q, a)$$

$\Rightarrow \hat{\bar{\delta}} : \mathcal{P}(Q) \times \Sigma^* \to \mathcal{P}(Q)$

Intuition:

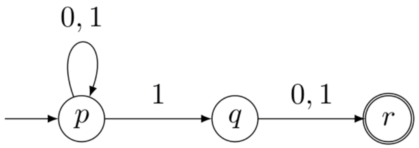
$\hat{\bar{\delta}}(S, w)$ ist Menge aller Zustände,
die sich von einem Zustand in S aus mit w erreichen lassen.

Die von $N = (Q, \Sigma, \delta, q_0, F)$ akzeptierte Sprache ist

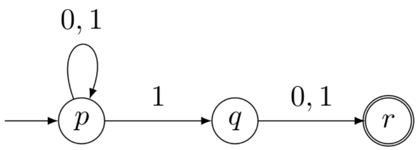
$$L(N) := \{w \in \Sigma^* \mid \hat{\bar{\delta}}(\{q_0\}, w) \cap F \neq \emptyset\}$$

Um Tod durch Notation zu vermeiden schreiben wir oft nur δ statt $\bar{\delta}$ und $\hat{\delta}$ statt $\hat{\bar{\delta}}$.

Beispiel

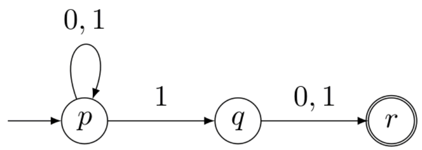


Beispiel



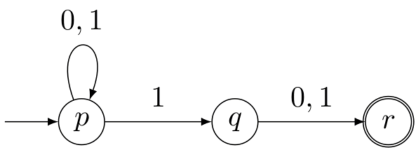
$$\begin{aligned} & \hat{\bar{\delta}}(\{p,q\},10) \\ = \quad & \hat{\bar{\delta}}(\bar{\delta}(\{p,q\},1),0) \end{aligned}$$

Beispiel



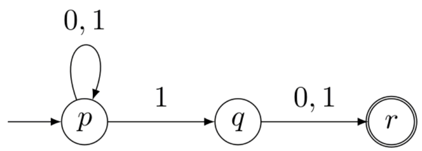
$$\begin{aligned} & \hat{\bar{\delta}}(\{p,q\},10) \\ = \quad & \hat{\bar{\delta}}(\bar{\delta}(\{p,q\},1),0) \\ = \quad & \hat{\bar{\delta}}(\delta(p,1) \cup \delta(q,1),0) \end{aligned}$$

Beispiel



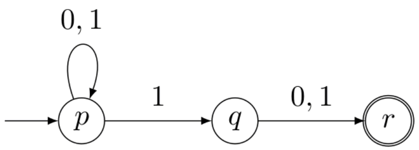
$$\begin{aligned} & \hat{\bar{\delta}}(\{p,q\},10) \\ = \quad & \hat{\bar{\delta}}(\bar{\delta}(\{p,q\},1),0) \\ = \quad & \hat{\bar{\delta}}(\delta(p,1) \cup \delta(q,1),0) \\ = \quad & \hat{\bar{\delta}}(\{p,q,r\},0) \end{aligned}$$

Beispiel



$$\begin{aligned} & \hat{\bar{\delta}}(\{p,q\},10) \\ = \quad & \hat{\bar{\delta}}(\bar{\delta}(\{p,q\},1),0) \\ = \quad & \hat{\bar{\delta}}(\delta(p,1) \cup \delta(q,1),0) \\ = \quad & \hat{\bar{\delta}}(\{p,q,r\},0) \qquad 0 \in \Sigma^* \\ = \quad & \bar{\delta}(\{p,q,r\},0) \qquad 0 \in \Sigma \end{aligned}$$

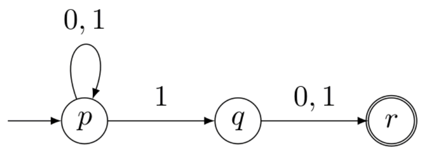
Beispiel



$$
\begin{aligned}
& \hat{\bar{\delta}}(\{p,q\},10) \\
= \quad & \hat{\bar{\delta}}(\bar{\delta}(\{p,q\},1),0) \\
= \quad & \hat{\bar{\delta}}(\delta(p,1) \cup \delta(q,1),0) \\
= \quad & \hat{\bar{\delta}}(\{p,q,r\},0) \\
= \quad & \bar{\delta}(\{p,q,r\},0) \\
= \quad & \delta(p,0) \cup \delta(q,0) \cup \delta(r,0)
\end{aligned}
$$

$\{p\}$ $\{r\}$ $\emptyset$

Beispiel



$$
\begin{aligned}
& \hat{\bar{\delta}}(\{p,q\},10) \\
= \quad & \hat{\bar{\delta}}(\bar{\delta}(\{p,q\},1),0) \\
= \quad & \hat{\bar{\delta}}(\delta(p,1) \cup \delta(q,1),0) \\
= \quad & \hat{\bar{\delta}}(\{p,q,r\},0) \\
= \quad & \bar{\delta}(\{p,q,r\},0) \\
= \quad & \delta(p,0) \cup \delta(q,0) \cup \delta(r,0) \\
= \quad & \{p\} \cup \{r\} \cup \emptyset \; = \; \{p,r\}
\end{aligned}
$$

- NFA wird (typischerweise) nicht direkt als Recognizer verwendet

- NFA wird (typischerweise) nicht direkt als Recognizer verwendet
- sonder ist meist Zwischenstation auf dem Weg zum DFA

- NFA wird (typischerweise) nicht direkt als Recognizer verwendet
- sonder ist meist Zwischenstation auf dem Weg zum DFA oder kompakte Darstellung einer Sprache

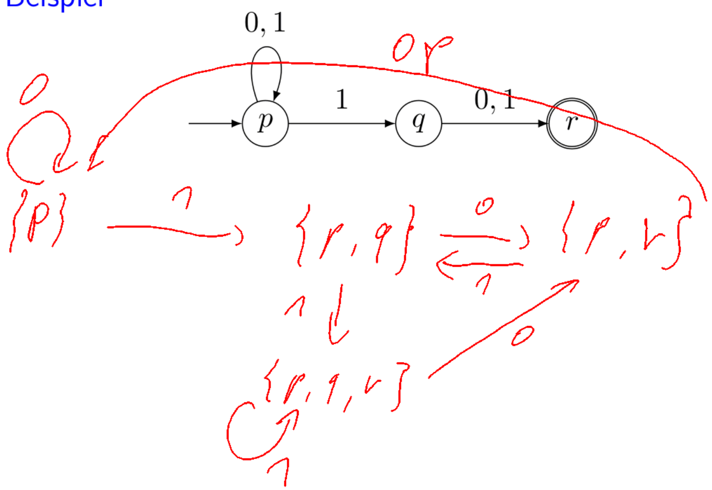
3.3 Äquivalenz von NFA und DFA

Satz 3.8
Für jeden NFA N gibt es einen DFA M mit $L(N) = L(M)$.

3.3 Äquivalenz von NFA und DFA

Satz 3.8
Für jeden NFA N gibt es einen DFA M mit $L(N) = L(M)$.

Beispiel



Beweis:
Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.

Beweis:
Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.
Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$\bar{\delta} : P(Q) \times \Sigma \to P(Q)$

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Beweis:
Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.

Beweis:
Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.
Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Dann gilt:

$$\begin{array}{rcll} w \in L(N) & \Leftrightarrow & \hat{\hat{\delta}}(\{q_0\}, w) \cap F \neq \emptyset & \text{Def.} \\ & \Leftrightarrow & \hat{\bar{\delta}}(\{q_0\}, w) \in F_M & \text{Def.} \\ & \Leftrightarrow & w \in L(M) & \text{Def.} \end{array} \quad \square$$

Beweis:
Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.
Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Dann gilt:

$$\begin{array}{rcll} w \in L(N) & \Leftrightarrow & \hat{\hat{\delta}}(\{q_0\}, w) \cap F \neq \emptyset & \text{Def.} \\ & \Leftrightarrow & \hat{\bar{\delta}}(\{q_0\}, w) \in F_M & \text{Def.} \\ & \Leftrightarrow & w \in L(M) & \text{Def.} \end{array} \quad \square$$

Dies nennt man die Potenzmengen- oder Teilmengenkonstruktion.

Beweis:
Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.
Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Dann gilt:

$$\begin{aligned} w \in L(N) &\Leftrightarrow \hat{\hat{\delta}}(\{q_0\}, w) \cap F \neq \emptyset && \text{Def.} \\ &\Leftrightarrow \hat{\bar{\delta}}(\{q_0\}, w) \in F_M && \text{Def.} \\ &\Leftrightarrow w \in L(M) && \text{Def.} \quad \square \end{aligned}$$

Dies nennt man die Potenzmengen- oder Teilmengenkonstruktion.

In der Praxis werden nur die aus q_0 erreichbaren Zuständen konstruiert.

Beweis:
Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.
Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Dann gilt:

$$\begin{aligned} w \in L(N) &\Leftrightarrow \hat{\hat{\delta}}(\{q_0\}, w) \cap F \neq \emptyset && \text{Def.} \\ &\Leftrightarrow \hat{\bar{\delta}}(\{q_0\}, w) \in F_M && \text{Def.} \\ &\Leftrightarrow w \in L(M) && \text{Def.} \quad \square \end{aligned}$$

Dies nennt man die Potenzmengen- oder Teilmengenkonstruktion.

In der Praxis werden nur die aus q_0 erreichbaren Zuständen konstruiert.

Trotzdem: Für einen NFA mit n Zuständen kann der entsprechende DFA bis zu 2^n Zustände haben.

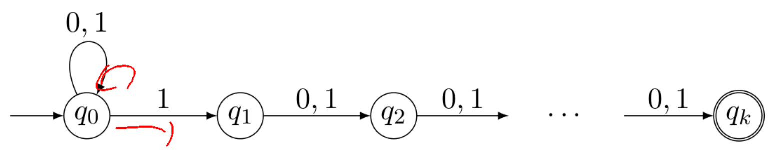
Beispiel 3.9

$$L_k := \{w \in \{0,1\}^* \mid \text{das } k\text{-letzte Bit von } w \text{ ist } 1\}$$

Beispiel 3.9

$$L_k := \{w \in \{0,1\}^* \mid \text{das } k\text{-letzte Bit von } w \text{ ist } 1\}$$

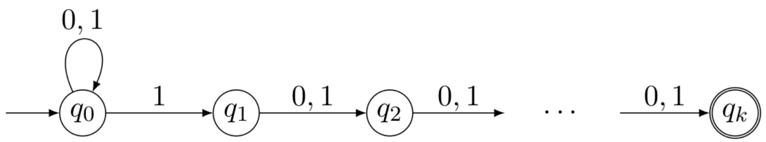
Ein NFA für diese Sprache:



Beispiel 3.9

$$L_k := \{w \in \{0,1\}^* \mid \text{das } k\text{-letzte Bit von } w \text{ ist } 1\}$$

Ein NFA für diese Sprache:

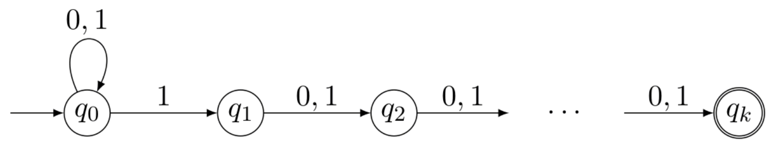


Die Potenzmengenkonstruktion liefert einen DFA für L_k mit 2^k Zuständen. Geht es kompakter?

Beispiel 3.9

$$L_k := \{w \in \{0,1\}^* \mid \text{das } k\text{-letzte Bit von } w \text{ ist } 1\}$$

Ein NFA für diese Sprache:



Die Potenzmengenkonstruktion liefert einen DFA für L_k mit 2^k Zuständen. Geht es kompakter?

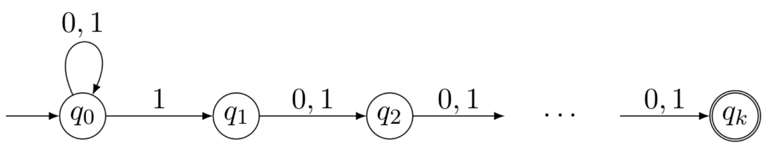
Lemma 3.10

Jeder DFA M mit $L(M) = L_k$ hat mindestens 2^k Zuständen.

Beispiel 3.9

$$L_k := \{w \in \{0,1\}^* \mid \text{das } k\text{-letzte Bit von } w \text{ ist } 1\}$$

Ein NFA für diese Sprache:



Die Potenzmengenkonstruktion liefert einen DFA für L_k mit 2^k Zuständen. Geht es kompakter?

Lemma 3.10

Jeder DFA M mit $L(M) = L_k$ hat mindestens 2^k Zuständen.

Im *schlimmsten* Fall ist ein exponentieller Sprung unvemeidlich.

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$: wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$:
  wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.
  Annahme: Es gibt $w_1, w_2 \in \{0,1\}^k$ mit $w_1 \neq w_2$ aber $\hat{\delta}(q_0, w_1) = \hat{\delta}(q_0, w_2)$. Wir leiten einen Widerspruch ab:

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$:
  wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.
  Annahme: Es gibt $w_1, w_2 \in \{0,1\}^k$ mit $w_1 \neq w_2$ aber $\hat{\delta}(q_0, w_1) = \hat{\delta}(q_0, w_2)$. Wir leiten einen Widerspruch ab:
- Sei $w_1 = wa_i \dots a_k$ und $w_2 = wb_i \dots b_k$ mit $a_i \neq b_i$

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$:
  wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.
  Annahme: Es gibt $w_1, w_2 \in \{0,1\}^k$ mit $w_1 \neq w_2$ aber $\hat{\delta}(q_0, w_1) = \hat{\delta}(q_0, w_2)$. Wir leiten einen Widerspruch ab:
- Sei $w_1 = w\overset{1}{a_i} \dots a_k$ und $w_2 = w\underset{0}{b_i} \dots b_k$ mit $a_i \neq b_i$
  OE sei $a_i = 1$, $b_i = 0$.

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$:
  wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.
  Annahme: Es gibt $w_1, w_2 \in \{0,1\}^k$ mit $w_1 \neq w_2$ aber $\hat{\delta}(q_0, w_1) = \hat{\delta}(q_0, w_2)$. Wir leiten einen Widerspruch ab:
- Sei $w_1 = wa_i \dots a_k$ und $w_2 = wb_i \dots b_k$ mit $a_i \neq b_i$
  OE sei $a_i = 1$, $b_i = 0$.
  k-letzte Bit
  Es gilt einerseits: $w_1 0^{i-1} = \underline{wa_i \dots a_k} 0^{i-1} \in L_k$

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$:
  wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.
  Annahme: Es gibt $w_1, w_2 \in \{0,1\}^k$ mit $w_1 \neq w_2$ aber $\hat{\delta}(q_0, w_1) = \hat{\delta}(q_0, w_2)$. Wir leiten einen Widerspruch ab:
- Sei $w_1 = wa_i \dots a_k$ und $w_2 = wb_i \dots b_k$ mit $a_i \neq b_i$

  OE sei $a_i = 1,\ b_i = 0$.

  Es gilt einerseits: $w_1 0^{i-1} = wa_i \dots a_k 0^{i-1} \in L_k$
  $w_2 0^{i-1} = wb_i \dots b_k 0^{i-1} \notin L_k$

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$:
  wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.
  Annahme: Es gibt $w_1, w_2 \in \{0,1\}^k$ mit $w_1 \neq w_2$ aber $\hat{\delta}(q_0, w_1) = \hat{\delta}(q_0, w_2)$. Wir leiten einen Widerspruch ab:
- Sei $w_1 = wa_i \dots a_k$ und $w_2 = wb_i \dots b_k$ mit $a_i \neq b_i$

  OE sei $a_i = 1,\ b_i = 0$.

  Es gilt einerseits: $w_1 0^{i-1} = wa_i \dots a_k 0^{i-1} \in L_k$
  $w_2 0^{i-1} = wb_i \dots b_k 0^{i-1} \notin L_k$

  Aber es gilt auch: $\boxed{\hat{\delta}(q_0, w_1 0^{i-1})} = \hat{\delta}(\hat{\delta}(q_0, w_1), 0^{i-1}) =$
  $\hat{\delta}(\hat{\delta}(q_0, w_2), 0^{i-1}) = \boxed{\hat{\delta}(q_0, w_2 0^{i-1})}$ ↯

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$:
  wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.
  Annahme: Es gibt $w_1, w_2 \in \{0,1\}^k$ mit $w_1 \neq w_2$ aber $\hat{\delta}(q_0, w_1) = \hat{\delta}(q_0, w_2)$. Wir leiten einen Widerspruch ab:
- Sei $w_1 = wa_i \dots a_k$ und $w_2 = wb_i \dots b_k$ mit $a_i \neq b_i$

  OE sei $a_i = 1,\ b_i = 0$.

  Es gilt einerseits: $w_1 0^{i-1} = wa_i \dots a_k 0^{i-1} \in L_k$
  $w_2 0^{i-1} = wb_i \dots b_k 0^{i-1} \notin L_k$

  Aber es gilt auch: $\hat{\delta}(q_0, w_1 0^{i-1}) = \hat{\delta}(\hat{\delta}(q_0, w_1), 0^{i-1}) =$
  $\hat{\delta}(\hat{\delta}(q_0, w_2), 0^{i-1}) = \hat{\delta}(q_0, w_2 0^{i-1})$ ↯
- Es folgt: M hat mindestens 2^k Zustände.

□

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$:
  wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.
  Annahme: Es gibt $w_1, w_2 \in \{0,1\}^k$ mit $w_1 \neq w_2$ aber $\hat{\delta}(q_0, w_1) = \hat{\delta}(q_0, w_2)$. Wir leiten einen Widerspruch ab:
- Sei $w_1 = wa_i \dots a_k$ und $w_2 = wb_i \dots b_k$ mit $a_i \neq b_i$

  OE sei $a_i = 1,\ b_i = 0$.

  Es gilt einerseits: $w_1 0^{i-1} = wa_i \dots a_k 0^{i-1} \in L_k$
  $w_2 0^{i-1} = wb_i \dots b_k 0^{i-1} \notin L_k$

  Aber es gilt auch: $\hat{\delta}(q_0, w_1 0^{i-1}) = \hat{\delta}(\hat{\delta}(q_0, w_1), 0^{i-1}) =$
  $\hat{\delta}(\hat{\delta}(q_0, w_2), 0^{i-1}) = \hat{\delta}(q_0, w_2 0^{i-1})$ ↯
- Es folgt: M hat mindestens 2^k Zustände.

□

Beweis:

Sei M ein DFA mit $L(M) = L_k$.

- Zuerst zeigen wir für alle $w_1, w_2 \in \{0,1\}^k$:
  wenn $w_1 \neq w_2$ dann $\hat{\delta}(q_0, w_1) \neq \hat{\delta}(q_0, w_2)$.
  Annahme: Es gibt $w_1, w_2 \in \{0,1\}^k$ mit $w_1 \neq w_2$ aber $\hat{\delta}(q_0, w_1) = \hat{\delta}(q_0, w_2)$. Wir leiten einen Widerspruch ab:
- Sei $w_1 = wa_i \ldots a_k$ und $w_2 = wb_i \ldots b_k$ mit $a_i \neq b_i$

  OE sei $a_i = 1,\ b_i = 0$.

  Es gilt einerseits: $w_1 0^{i-1} = wa_i \ldots a_k 0^{i-1} \in L_k$
  $w_2 0^{i-1} = wb_i \ldots b_k 0^{i-1} \notin L_k$

3.4 NFAs mit ϵ-Übergängen

- Diese Erweiterung von NFAs macht sie nicht mächtiger (man bekommt wieder genau die regulären Sprachen)
- aber man kann manche Sprachen einfacher beschreiben als nur mit NFAs.

Definition 3.11

Ein NFA mit ϵ-Übergängen (auch ϵ-NFA) ist ein NFA mit einem speziellen Symbol $\epsilon \notin \Sigma$ und mit

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q) .$$

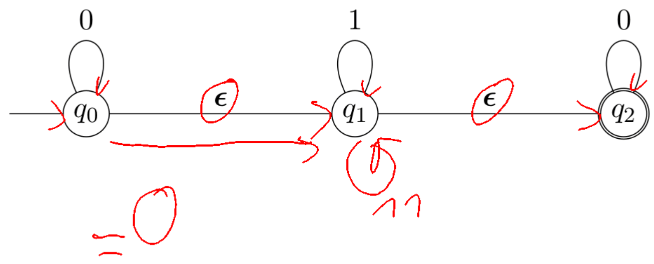
Ein ϵ-Übergang darf ausgeführt werden, ohne dass ein Eingabezeichen gelesen wird.

Definition 3.11

Ein NFA mit ϵ-Übergängen (auch ϵ-NFA) ist ein NFA mit einem speziellen Symbol $\epsilon \notin \Sigma$ und mit

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q) .$$

Ein ϵ-Übergang darf ausgeführt werden, ohne dass ein Eingabezeichen gelesen wird.

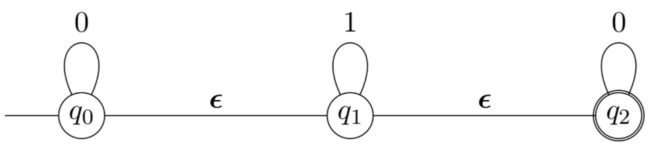


Definition 3.11

Ein NFA mit $\boldsymbol{\epsilon}$-Übergängen (auch $\boldsymbol{\epsilon}$-NFA) ist ein NFA mit einem speziellen Symbol $\boldsymbol{\epsilon} \notin \Sigma$ und mit

$$\delta : Q \times (\Sigma \cup \{\boldsymbol{\epsilon}\}) \to \mathcal{P}(Q)\ .$$

Ein $\boldsymbol{\epsilon}$-Übergang darf ausgeführt werden, ohne dass ein Eingabezeichen gelesen wird.



Akzeptiert: ϵ, 00, 11, ... Nicht akzeptiert: , ...

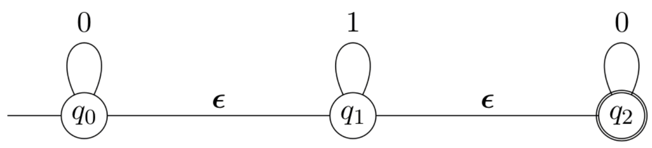
0101

Definition 3.11

Ein NFA mit $\boldsymbol{\epsilon}$-Übergängen (auch $\boldsymbol{\epsilon}$-NFA) ist ein NFA mit einem speziellen Symbol $\boldsymbol{\epsilon} \notin \Sigma$ und mit

$$\delta : Q \times (\Sigma \cup \{\boldsymbol{\epsilon}\}) \to \mathcal{P}(Q)\ .$$

Ein $\boldsymbol{\epsilon}$-Übergang darf ausgeführt werden, ohne dass ein Eingabezeichen gelesen wird.



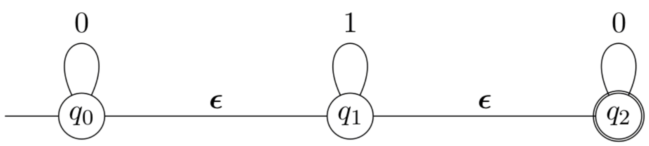
Akzeptiert: ϵ, 00, 11, ... Nicht akzeptiert: 101, ...

Definition 3.11

Ein NFA mit $\boldsymbol{\epsilon}$-Übergängen (auch $\boldsymbol{\epsilon}$-NFA) ist ein NFA mit einem speziellen Symbol $\boldsymbol{\epsilon} \notin \Sigma$ und mit

$$\delta : Q \times (\Sigma \cup \{\boldsymbol{\epsilon}\}) \to \mathcal{P}(Q)\ .$$

Ein $\boldsymbol{\epsilon}$-Übergang darf ausgeführt werden, ohne dass ein Eingabezeichen gelesen wird.



Akzeptiert: ϵ, 00, 11, ... Nicht akzeptiert: 101, ...

Bemerkung: $\boldsymbol{\epsilon} \neq \epsilon$; $\boldsymbol{\epsilon}$ ist ein einzelnes Symbol, ϵ das leere Wort.

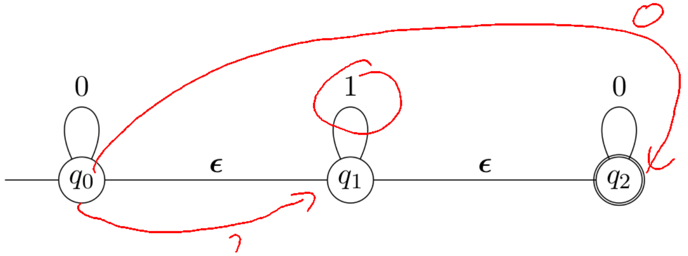
Formal betrachten wir einen $\boldsymbol{\epsilon}$-NFA $N = (Q, \Sigma, \delta, q_0, F)$ als kompakte Repräsentation eines $\boldsymbol{\epsilon}$-freien NFA $N' = (Q, \Sigma, \delta', q_0, F')$

Definition 3.11

Ein NFA mit ϵ-Übergängen (auch ϵ-NFA) ist ein NFA mit einem speziellen Symbol $\epsilon \notin \Sigma$ und mit

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q) \; .$$

Ein ϵ-Übergang darf ausgeführt werden, ohne dass ein Eingabezeichen gelesen wird.



Akzeptiert: ϵ, 00, 11, ... Nicht akzeptiert: 101, ...

Bemerkung: $\boldsymbol{\epsilon} \neq \epsilon$; $\boldsymbol{\epsilon}$ ist ein einzelnes Symbol, ϵ das leere Wort.

Formal betrachten wir einen ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ als kompakte Repräsentation eines ϵ-freien NFA $N' = (Q, \Sigma, \delta', q_0, F')$

Formal betrachten wir einen ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ als kompakte Repräsentation eines ϵ-freien NFA $N' = (Q, \Sigma, \delta', q_0, F')$

- $\delta' : Q \times \Sigma \to \mathcal{P}(Q)$:

$$\delta'(q, a) := \bigcup_{i \geq 0, j \geq 0} \hat{\delta}(\{q\}, \epsilon^i a \epsilon^j) \; .$$

- Falls N das leere Wort ϵ akzeptiert, also falls

$$\exists i \geq 0.\ \hat{\delta}(\{q_0\}, \epsilon^i) \cap F \neq \emptyset$$

dann setze $F' := F \cup \{q_0\}$, sonst setze $F' := F$.

Formal betrachten wir einen ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ als kompakte Repräsentation eines ϵ-freien NFA $N' = (Q, \Sigma, \delta', q_0, F')$

Formal betrachten wir einen ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ als kompakte Repräsentation eines ϵ-freien NFA $N' = (Q, \Sigma, \delta', q_0, F')$

- $\delta' : Q \times \Sigma \to \mathcal{P}(Q)$:

$$\delta'(q, a) := \bigcup_{i \geq 0, j \geq 0} \hat{\delta}(\{q\}, \boldsymbol{\epsilon}^i a \boldsymbol{\epsilon}^j) \ .$$

Formal betrachten wir einen ϵ-NFA $N = (Q, \Sigma, \delta, q_0, F)$ als kompakte Repräsentation eines ϵ-freien NFA $N' = (Q, \Sigma, \delta', q_0, F')$

- $\delta' : Q \times \Sigma \to \mathcal{P}(Q)$:

$$\delta'(q, a) := \bigcup_{i \geq 0, j \geq 0} \hat{\delta}(\{q\}, \boldsymbol{\epsilon}^i a \boldsymbol{\epsilon}^j) \ .$$

- Falls N das leere Wort ϵ akzeptiert, also falls

$$\exists i \geq 0.\ \hat{\delta}(\{q_0\}, \boldsymbol{\epsilon}^i) \cap F \neq \emptyset$$

  dann setze $F' := F \cup \{q_0\}$, sonst setze $F' := F$.

Damit gilt per definitionem:

Jeder ϵ-NFA ist äquivalent zu einem NFA.

Ab jetzt: "ϵ-Übergang" und "ϵ-NFA".

Fazit: Die Automatentypen

- DFA
- NFA
- ϵ-NFA

sind gleich mächtig.

3.5 Rechtslinearen Grammatiken

Erinnerung: rechtslineare Grammatiken haben nur Produktionen der Form $A \to a$, $A \to aB$ und $S \to \epsilon$.