Theoretische Informatik I

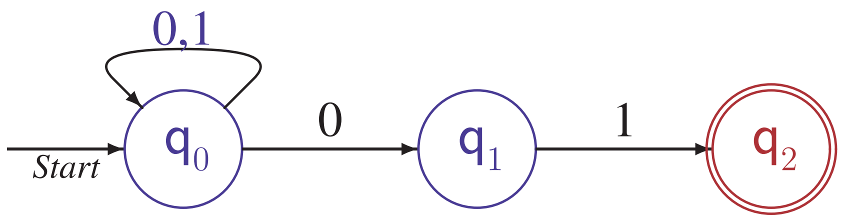
Einheit 2.2

Nichtdeterministische Automaten

1. Arbeitsweise
2. Akzeptierte Sprache
3. Äquivalenz zu deterministischen Automaten

Was ist Nichtdetermismus?

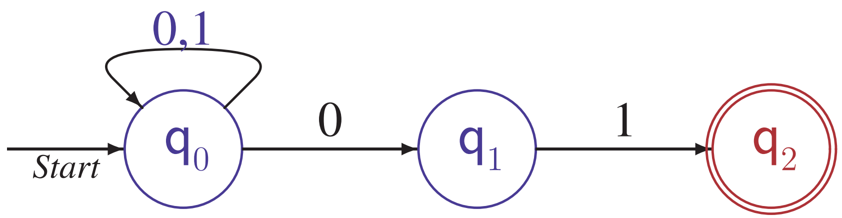
- **Verhalten nicht eindeutig bestimmt**
  - Automat wählt Folgezustand aus mehreren Möglichkeiten



  - Automat erkennt Strings, die mit `01` enden
  - Eine 0 kann das erste Symbol des Endes `01` sein ... oder auch nicht

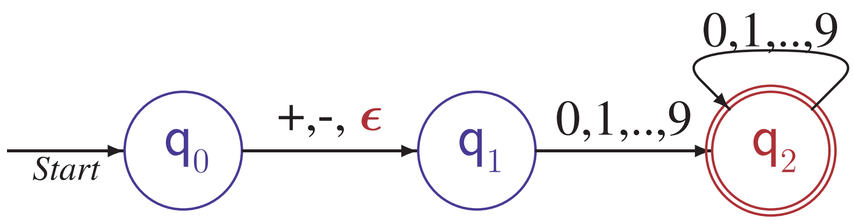
Was ist Nichtdetermismus?

- **Verhalten nicht eindeutig bestimmt**
  - Automat wählt Folgezustand aus mehreren Möglichkeiten



  - Automat erkennt Strings, die mit `01` enden
  - Eine 0 kann das erste Symbol des Endes `01` sein ... oder auch nicht

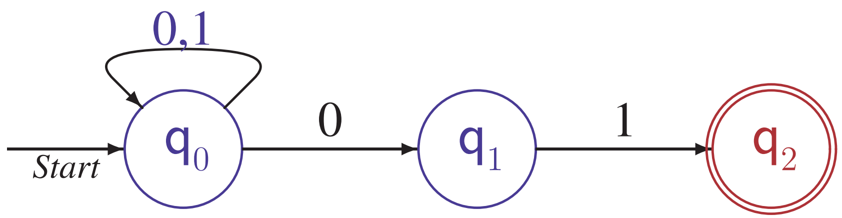
- **Spontane Übergänge zwischen Zuständen**
  - Automat geht ohne Eingabe in anderen Zustand über (**ϵ-Übergang**)



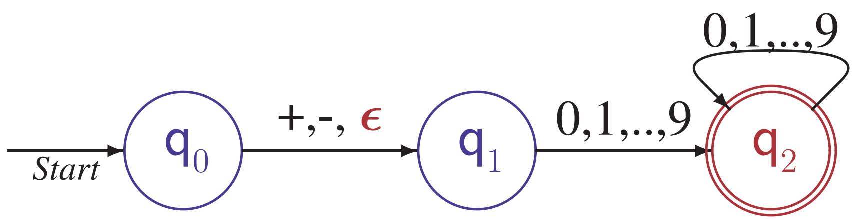
  - Automat erkennt ganze Zahlen mit und ohne Vorzeichen

Was ist Nichtdetermismus?

- **Verhalten nicht eindeutig bestimmt**
  - Automat wählt Folgezustand aus mehreren Möglichkeiten



  - Automat erkennt Strings, die mit `01` enden
  - Eine 0 kann das erste Symbol des Endes `01` sein ... oder auch nicht
- **Spontane Übergänge zwischen Zuständen**
  - Automat geht ohne Eingabe in anderen Zustand über (**ϵ-Übergang**)



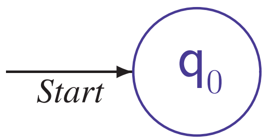
  - Automat erkennt ganze Zahlen mit und ohne Vorzeichen
- **Hilfreiches Modell für Entwurfsphase**
  - Elegantere Beschreibungsform, leichter als korrekt nachzuweisen
  - Begrenzte physikalische Realisierung durch Parallelrechner möglich

Nichtdeterministische Automaten – wozu?

- **Elegante Form der Textsuche in Dokumenten**
  – Viele verschiedene Wörter in großen Textsammlungen (Internet)
  – Leichte Beschreibung der Suchanfrage
  – Deterministisches Erkennungsverfahren mühsam zu beschreiben

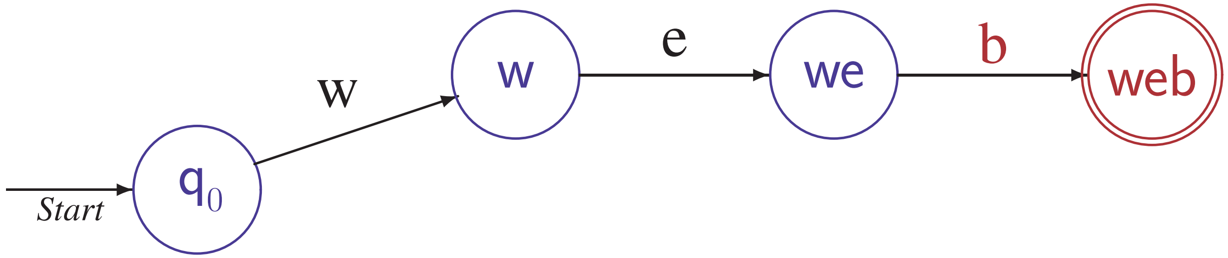
Nichtdeterministische Automaten – wozu?

- **Elegante Form der Textsuche in Dokumenten**
  - Viele verschiedene Wörter in großen Textsammlungen (Internet)
  - Leichte Beschreibung der Suchanfrage
  - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
  - z.B. Suche nach den Wörtern `web` und `ebay` am Ende eines Wortes



Nichtdeterministische Automaten – wozu?

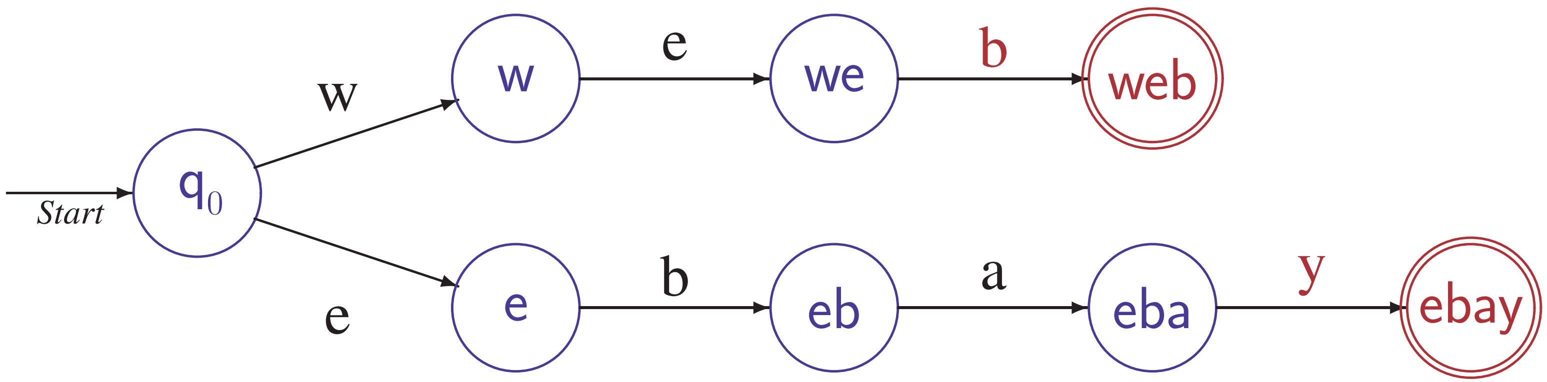
- **Elegante Form der Textsuche in Dokumenten**
  - Viele verschiedene Wörter in großen Textsammlungen (Internet)
  - Leichte Beschreibung der Suchanfrage
  - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
  - z.B. Suche nach den Wörtern `web` und `ebay` am Ende eines Wortes



  - Ein `w` könnte der Anfang von `web` sein

Nichtdeterministische Automaten – wozu?

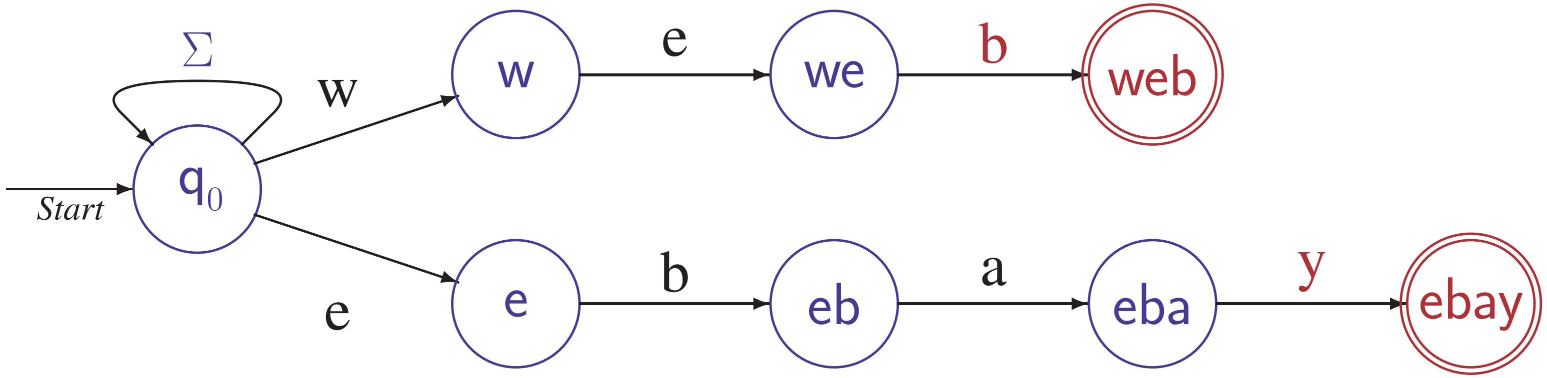
- **Elegante Form der Textsuche in Dokumenten**
  - Viele verschiedene Wörter in großen Textsammlungen (Internet)
  - Leichte Beschreibung der Suchanfrage
  - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
  - z.B. Suche nach den Wörtern `web` und `ebay` am Ende eines Wortes



  - Ein `w` könnte der Anfang von `web` sein
  - Ein `e` könnte der Anfang von `ebay` sein

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

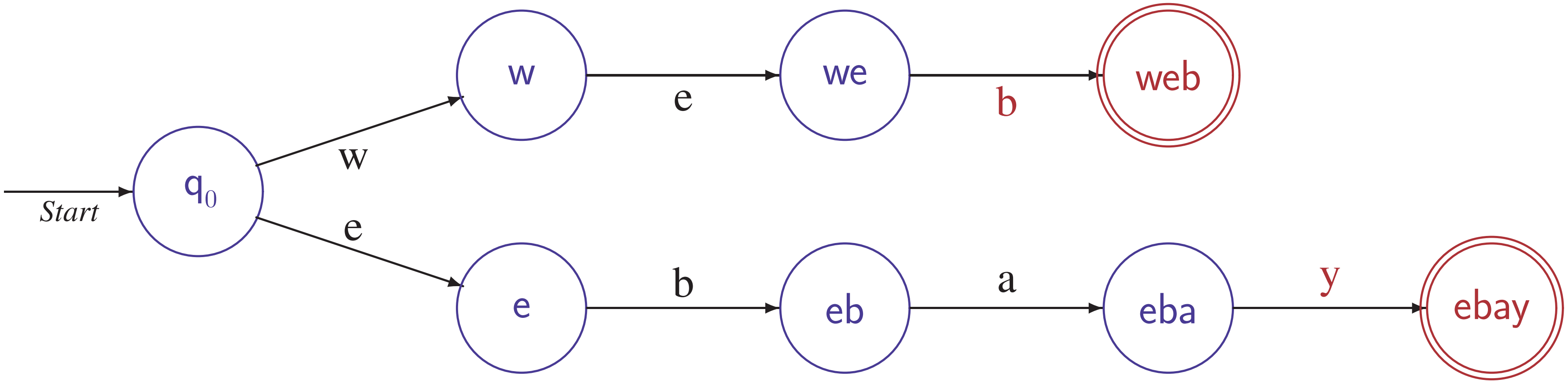
- **Elegante Form der Textsuche in Dokumenten**
  – Viele verschiedene Wörter in großen Textsammlungen (Internet)
  – Leichte Beschreibung der Suchanfrage
  – Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
  – z.B. Suche nach den Wörtern `web` und `ebay` am Ende eines Wortes



  – Ein `w` könnte der Anfang von `web` sein
  – Ein `e` könnte der Anfang von `ebay` sein
  – Aber vor den Wörtern könnte noch etwas anderes stehen

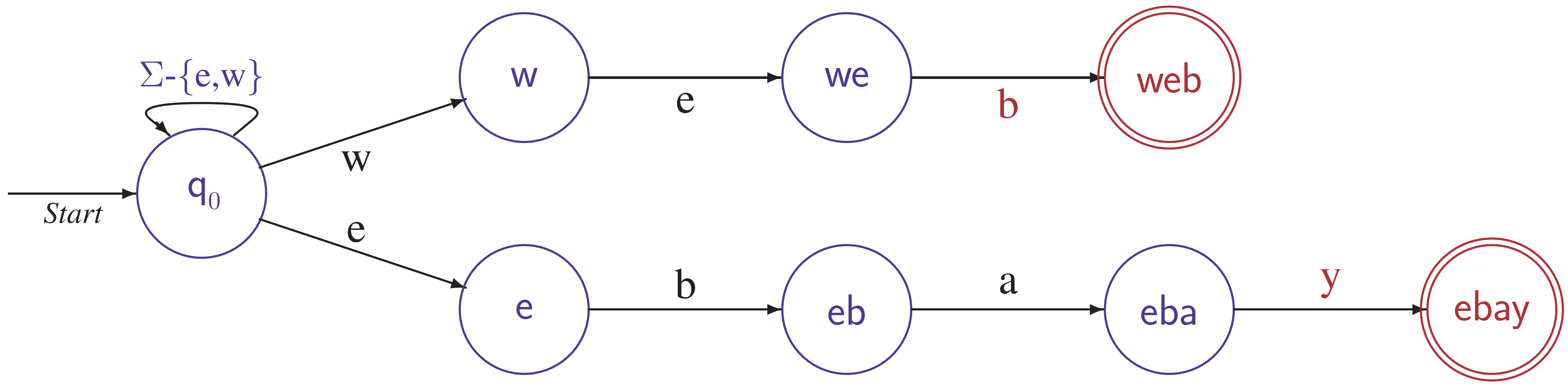
Nichtdeterminismus $\hat{=}$ verfolge alle Möglichkeiten simultan

Deterministische Variante ist viel komplizierter



Grundstruktur ist ähnlich

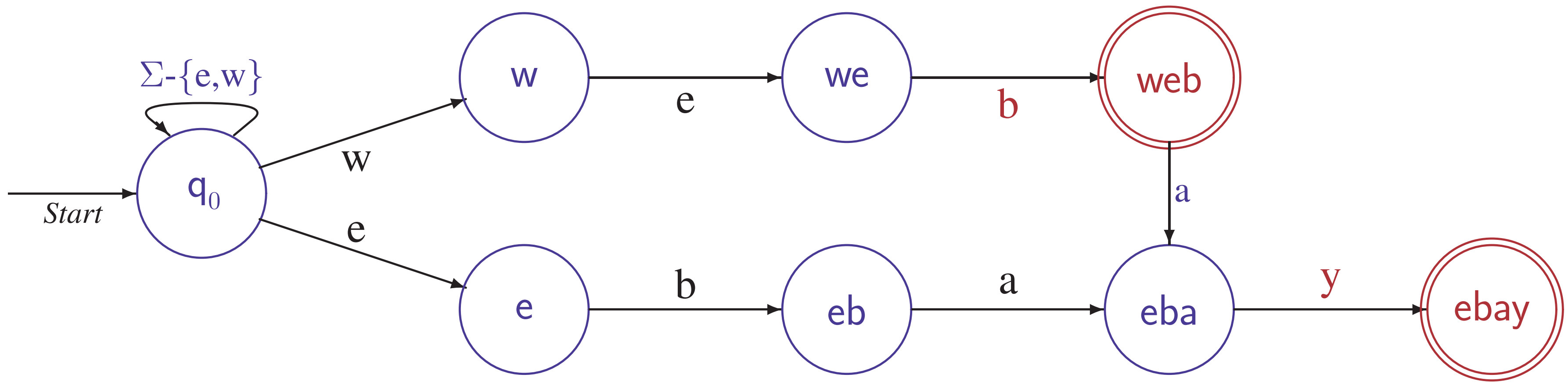
Deterministische Variante ist viel komplizierter



Grundstruktur ist ähnlich, ... aber

– Das Verhalten muß für alle möglichen Eingaben erklärt werden

Deterministische Variante ist viel komplizierter

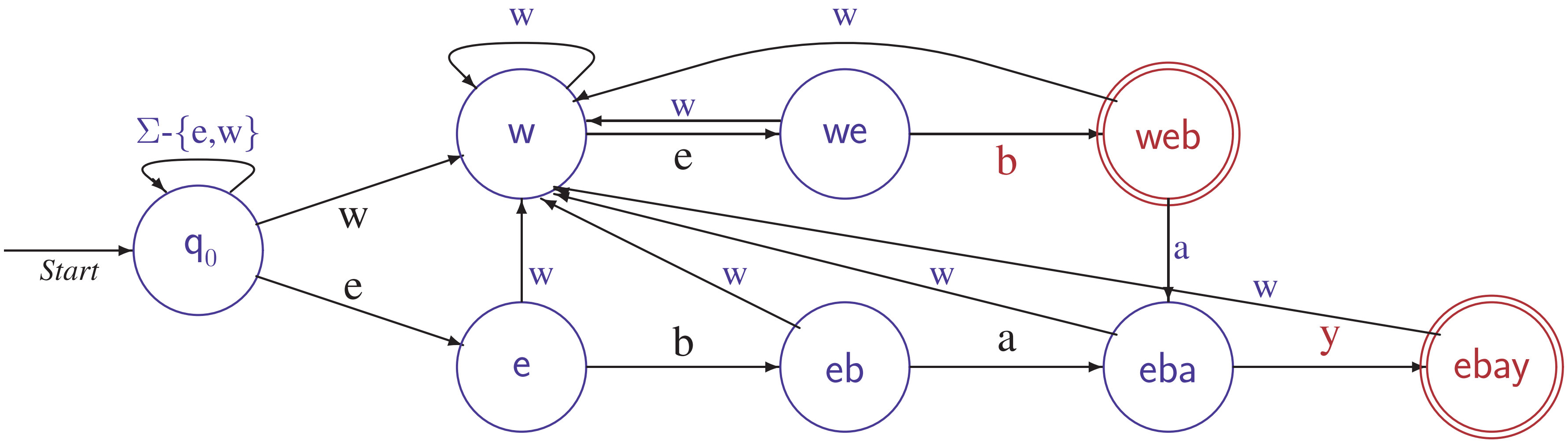


Grundstruktur ist ähnlich, ... aber

– Das Verhalten muß für alle möglichen Eingaben erklärt werden

– Endzustände müssen verlassen werden, wenn weitere Eingaben kommen

Deterministische Variante ist viel komplizierter



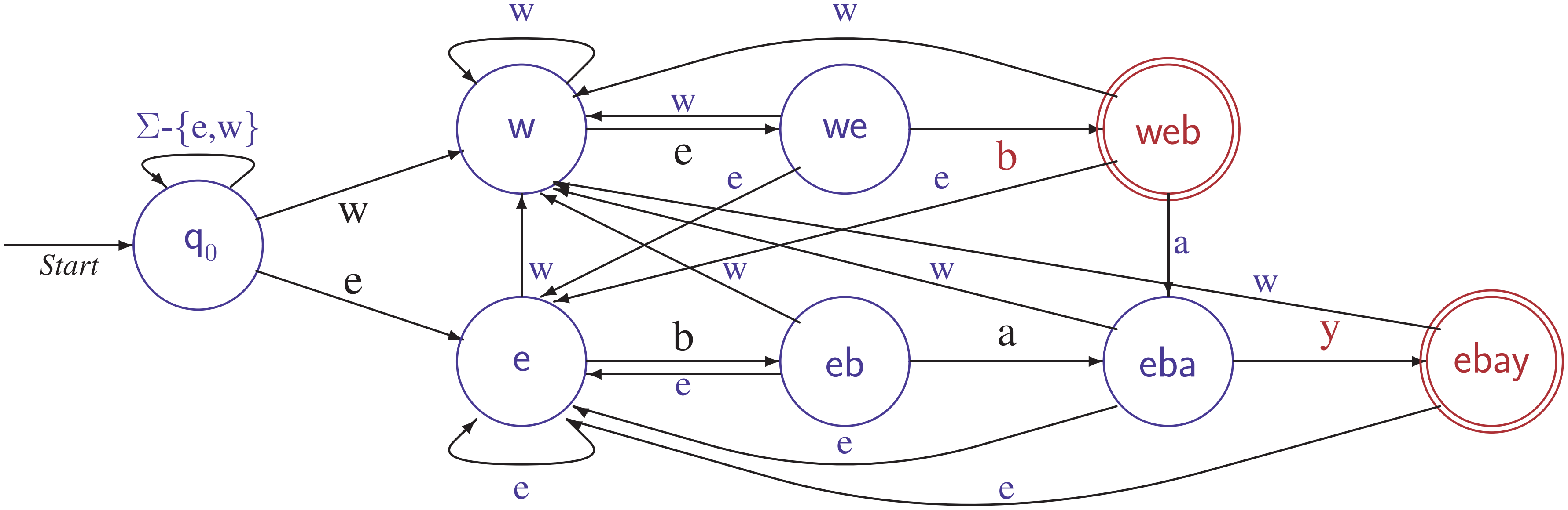
Grundstruktur ist ähnlich, ... aber

– Das Verhalten muß für alle möglichen Eingaben erklärt werden

– Endzustände müssen verlassen werden, wenn weitere Eingaben kommen

– Bei “unerwünschten” Eingaben muß man zurückspringen

Deterministische Variante ist viel komplizierter



Grundstruktur ist ähnlich, ... aber

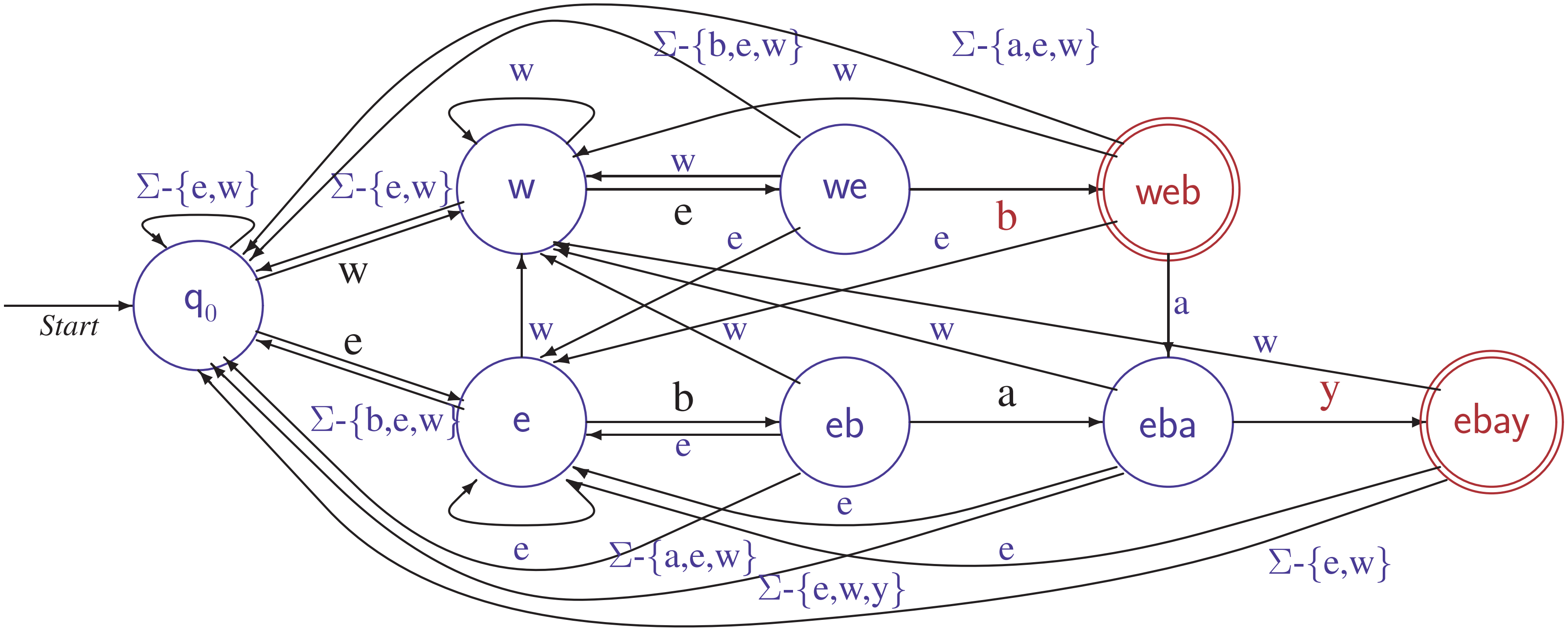
– Das Verhalten muß für alle möglichen Eingaben erklärt werden

– Endzustände müssen verlassen werden, wenn weitere Eingaben kommen

– Bei “unerwünschten” Eingaben muß man zurückspringen

– Man kann an viele Stellen zurückspringen

Deterministische Variante ist viel komplizierter



Grundstruktur ist ähnlich, ... aber

– Das Verhalten muß für alle möglichen Eingaben erklärt werden

– Endzustände müssen verlassen werden, wenn weitere Eingaben kommen

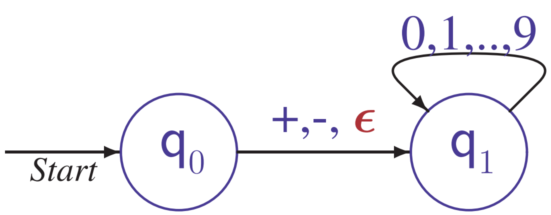
– Bei “unerwünschten” Eingaben muß man zurückspringen

– Man kann an viele Stellen zurückspringen

– und oft muß man wieder ganz von vorne anfangen

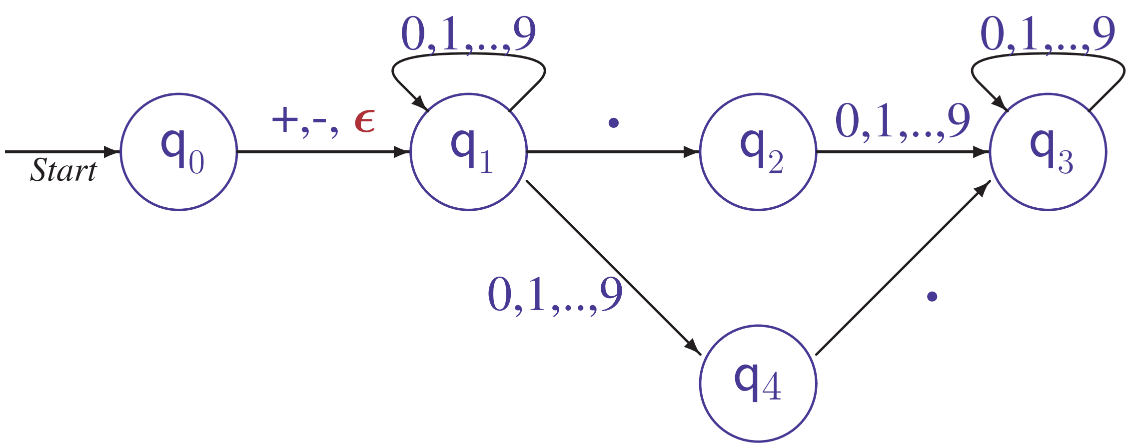
ϵ-Übergänge – Verarbeitung optionaler Eingaben

- **Erkenne Dezimalzahlen im Programmcode**
  - Zwei Zeichenreihen von Ziffern getrennt durch Dezimalpunkt
  - Eine der beiden Zeichenreihen darf leer sein, aber nicht beide
  - Optionales Vorzeichen + oder -



ϵ-Übergänge – Verarbeitung optionaler Eingaben

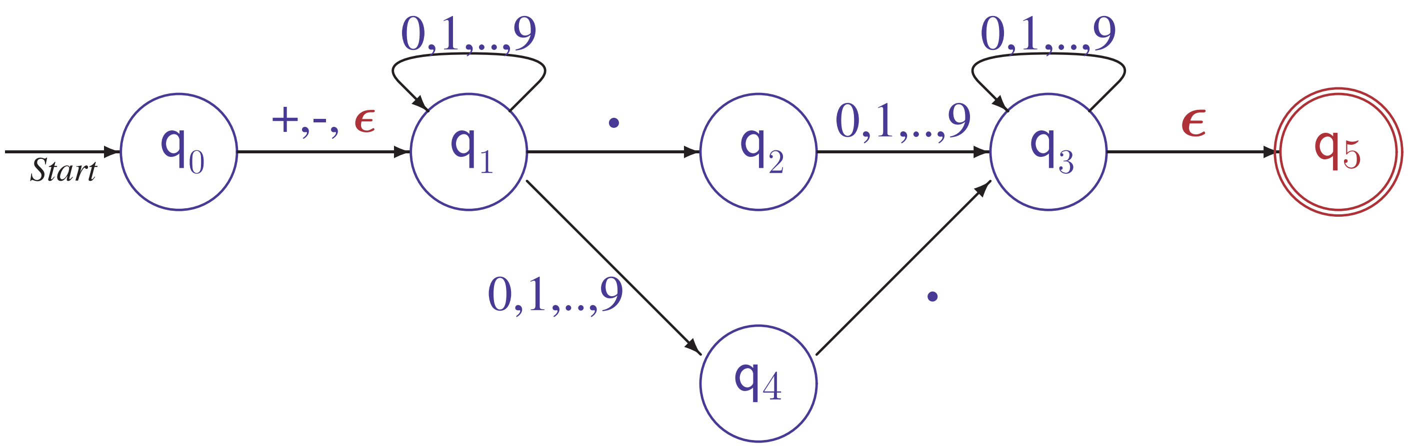
- **Erkenne Dezimalzahlen im Programmcode**
  - Zwei Zeichenreihen von Ziffern getrennt durch Dezimalpunkt
  - Eine der beiden Zeichenreihen darf leer sein, aber nicht beide
  - Optionales Vorzeichen + oder −



ϵ-Übergänge – Verarbeitung optionaler Eingaben

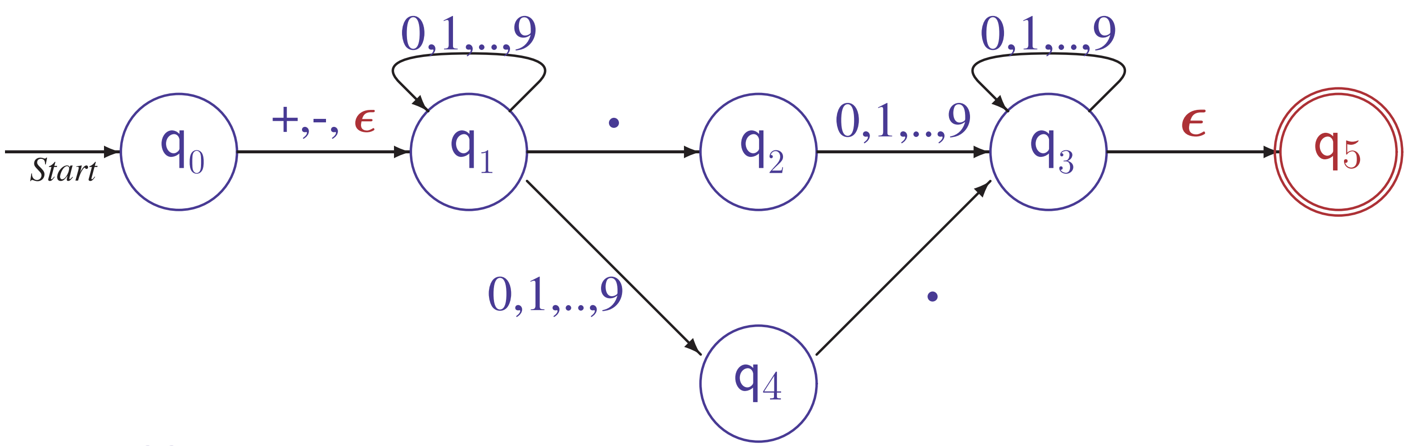
- **Erkenne Dezimalzahlen im Programmcode**
  - Zwei Zeichenreihen von Ziffern getrennt durch Dezimalpunkt
  - Eine der beiden Zeichenreihen darf leer sein, aber nicht beide
  - Optionales Vorzeichen + oder –

separater Endzustand

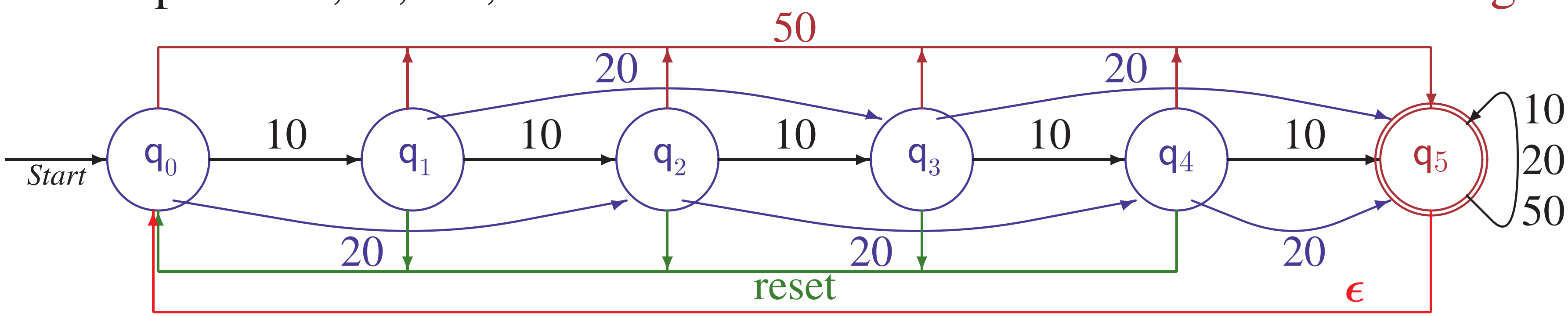


ϵ-Übergänge – Verarbeitung optionaler Eingaben

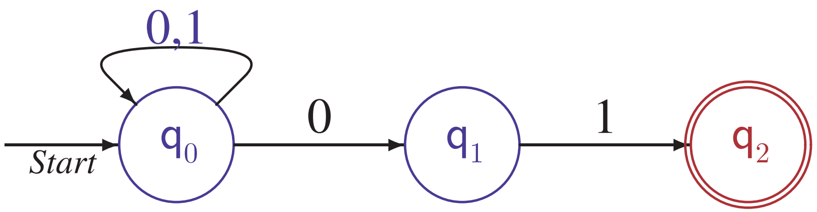
- **Erkenne Dezimalzahlen im Programmcode**
  - Zwei Zeichenreihen von Ziffern getrennt durch Dezimalpunkt
  - Eine der beiden Zeichenreihen darf leer sein, aber nicht beide
  - Optionales Vorzeichen + oder – separater Endzustand

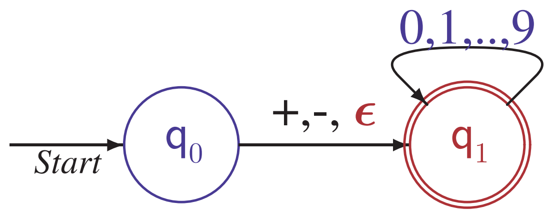


- **50c Kaffeeautomat**
  - Akzeptiert 10,20,50c, mit Reset-Taste und automatischer Rücksetzung



Nichtdeterministische Automaten – präzisiert





Ein **ϵ-NEA** (**nichtdeterministischer endlicher Automat mit ϵ-Übergängen**) ist ein 5-Tupel $A = (Q, \Sigma, \delta, q_0, F)$ mit

- Q nichtleere endliche **Zustandsmenge**
- Σ (endliches) **Eingabealphabet** mit $\epsilon \notin \Sigma$
- $\delta: Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q)$ **Zustandsüberführungsfunktion** *
- $q_0 \in Q$ **Startzustand**
- $F \subseteq Q$ Menge von **akzeptierenden** (End-) **Zuständen**

Ein **NEA** ist ein nichtdet. endlicher Automat ohne ϵ-Übergänge

Oft wird "NEA" als Oberbegriff für Automaten mit und ohne ϵ-Übergänge verwendet

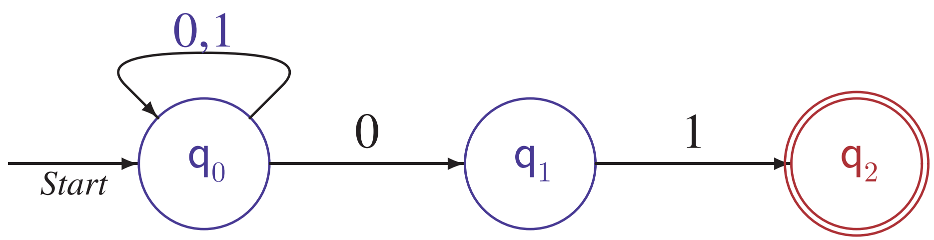
---

* $\mathcal{P}(Q) = \{S \mid S \subseteq Q\}$ (**Potenzmenge** von Q)

Bei ϵ-NEAs ist $\delta(q', a)$ ist eine (möglicherweise leere) Menge von Zuständen

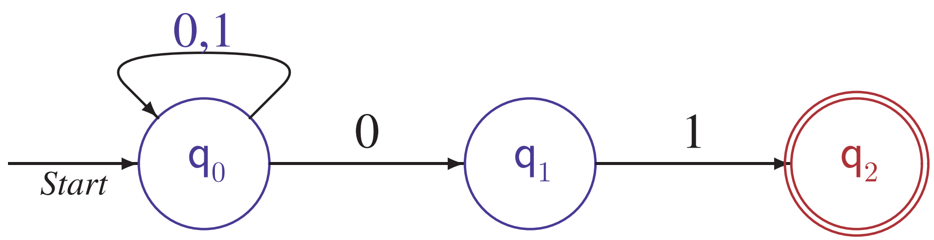
Arbeitsweise von NEAs

Erkenne Strings, die mit `01` enden



Arbeitsweise von NEAs

Erkenne Strings, die mit `01` enden



(1) Jedes Teilwort kann in q_0 bleiben

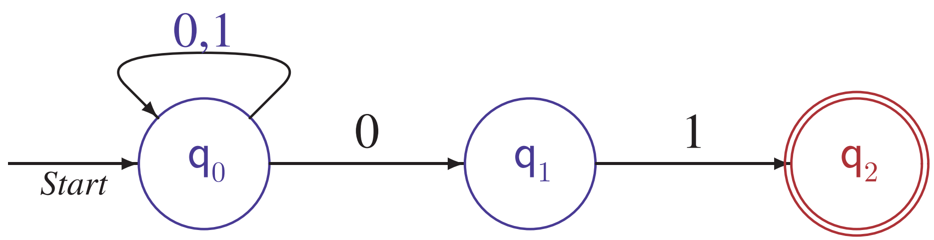
(2) Ein Teilwort muss mit `0` enden, um nach q_1 zu führen

(3) Ein Teilwort muss mit `01` enden, um nach q_2 zu führen

(4) In q_2 muss das Wort abgearbeitet sein

Arbeitsweise von NEAs

Erkenne Strings, die mit `01` enden



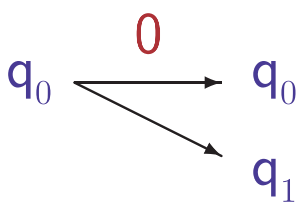
(1) Jedes Teilwort kann in q_0 bleiben

(2) Ein Teilwort muss mit `0` enden, um nach q_1 zu führen

(3) Ein Teilwort muss mit `01` enden, um nach q_2 zu führen

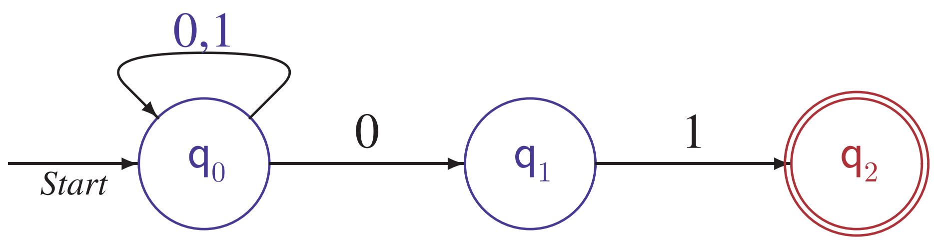
(4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von `00101`



Arbeitsweise von NEAs

Erkenne Strings, die mit `01` enden



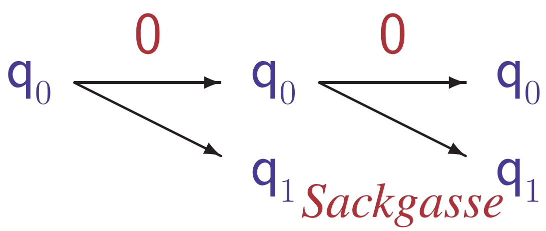
(1) Jedes Teilwort kann in q_0 bleiben

(2) Ein Teilwort muss mit `0` enden, um nach q_1 zu führen

(3) Ein Teilwort muss mit `01` enden, um nach q_2 zu führen

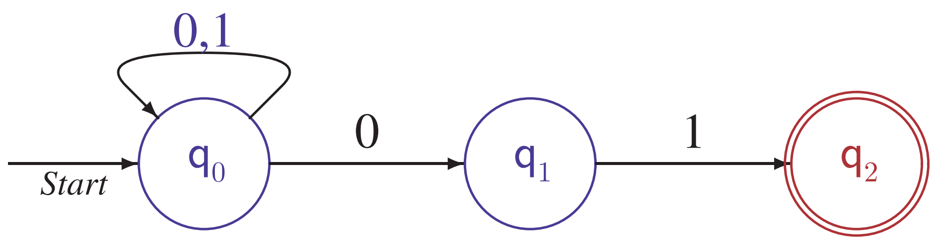
(4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von `00101`



Arbeitsweise von NEAs

Erkenne Strings, die mit `01` enden



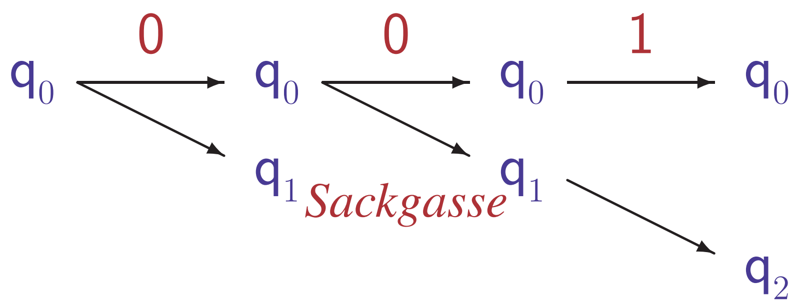
(1) Jedes Teilwort kann in q_0 bleiben

(2) Ein Teilwort muss mit `0` enden, um nach q_1 zu führen

(3) Ein Teilwort muss mit `01` enden, um nach q_2 zu führen

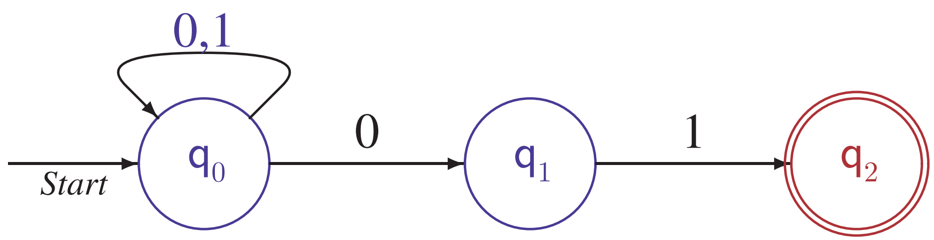
(4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von `00101`



Arbeitsweise von NEAs

Erkenne Strings, die mit 01 enden



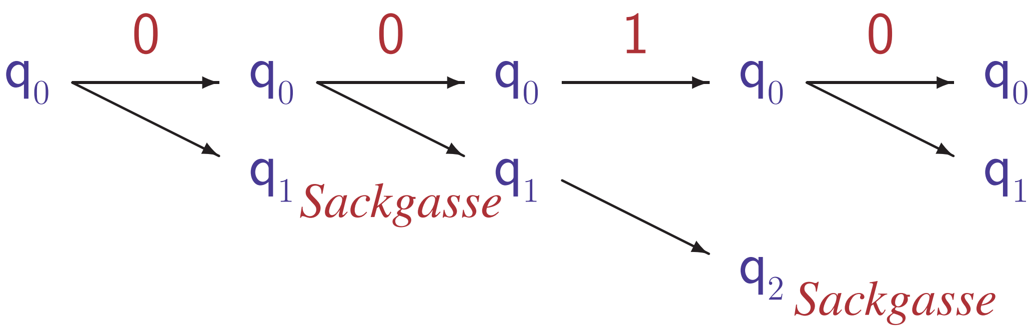
(1) Jedes Teilwort kann in q_0 bleiben

(2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen

(3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen

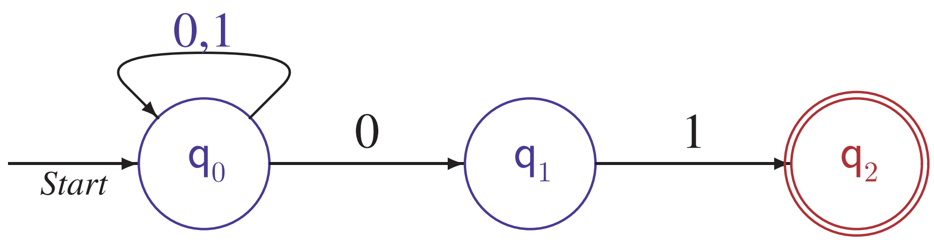
(4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101



Arbeitsweise von NEAs

Erkenne Strings, die mit 01 enden



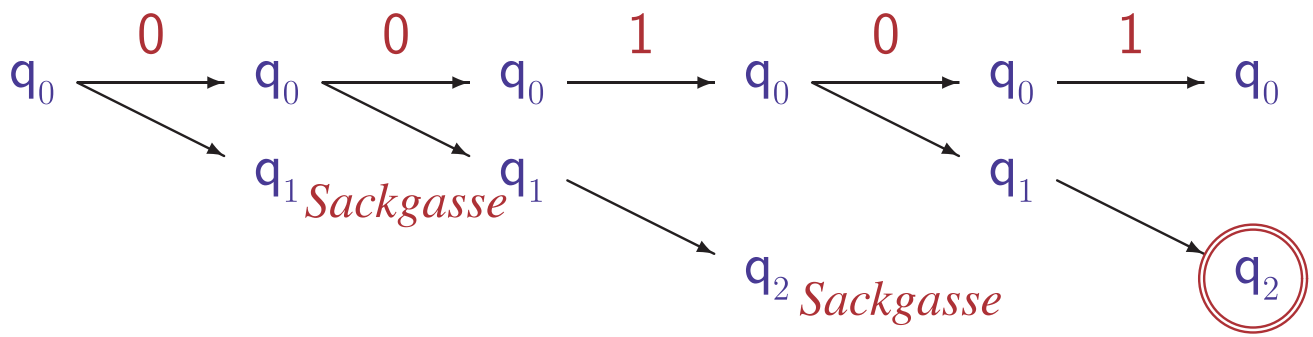
(1) Jedes Teilwort kann in q_0 bleiben

(2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen

(3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen

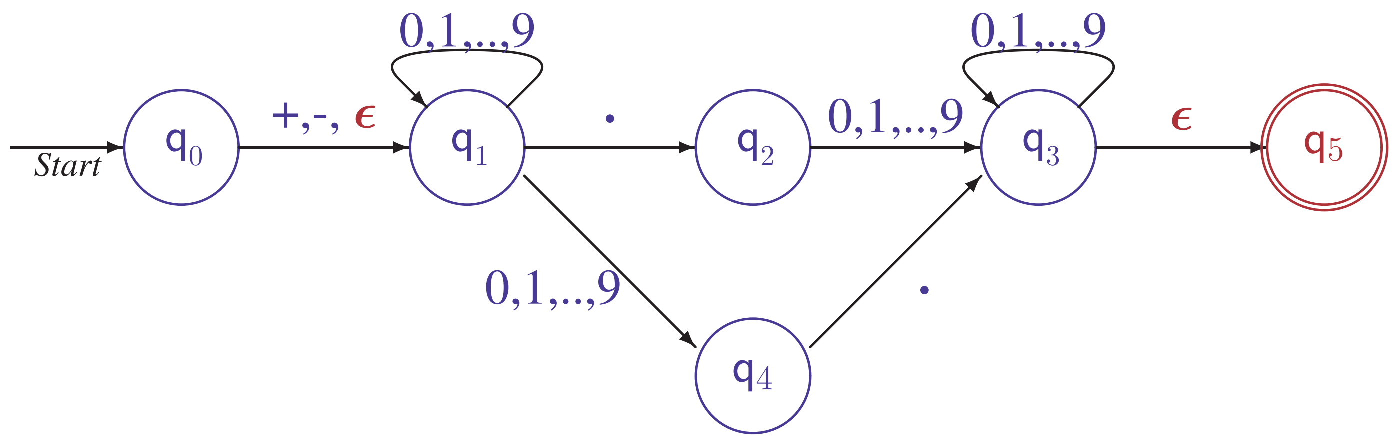
(4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101

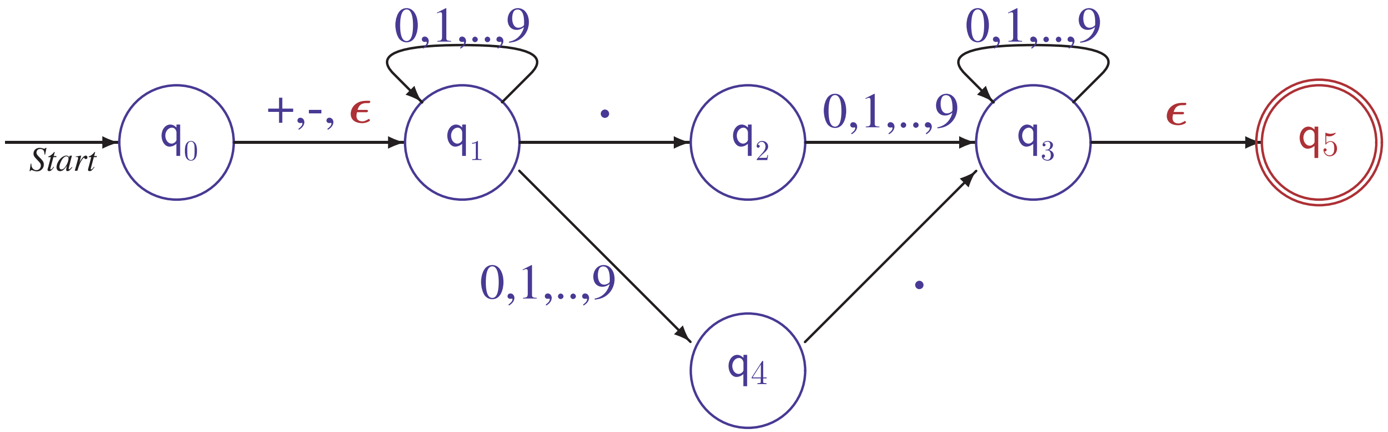


Ein Abarbeitungsweg führt zu einem akzeptierenden Zustand

Arbeitsweise von NEAs mit ϵ-Übergängen



Arbeitsweise von NEAs mit ϵ-Übergängen



(1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1

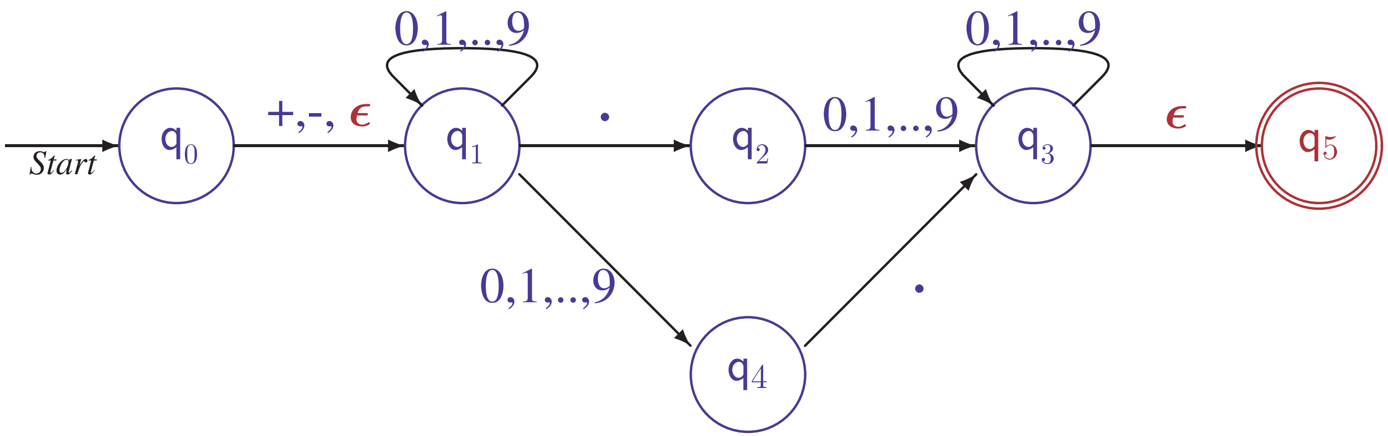
(2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4

(3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3

(4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3

(5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Arbeitsweise von NEAs mit ϵ-Übergängen



(1) Die Teilwörter +, −, und ϵ führen nach q_1

(2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4

(3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3

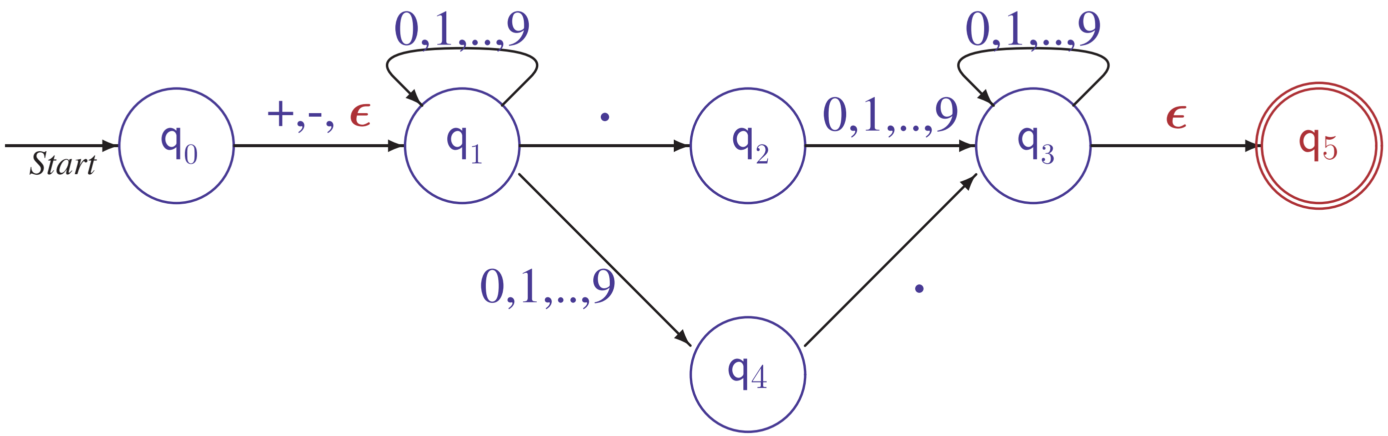
(4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3

(5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von `3.14159`

$q_0 \xrightarrow{\epsilon} q_1$

Arbeitsweise von NEAs mit ϵ-Übergängen



(1) Die Teilwörter +, −, und ϵ führen nach q_1

(2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4

(3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3

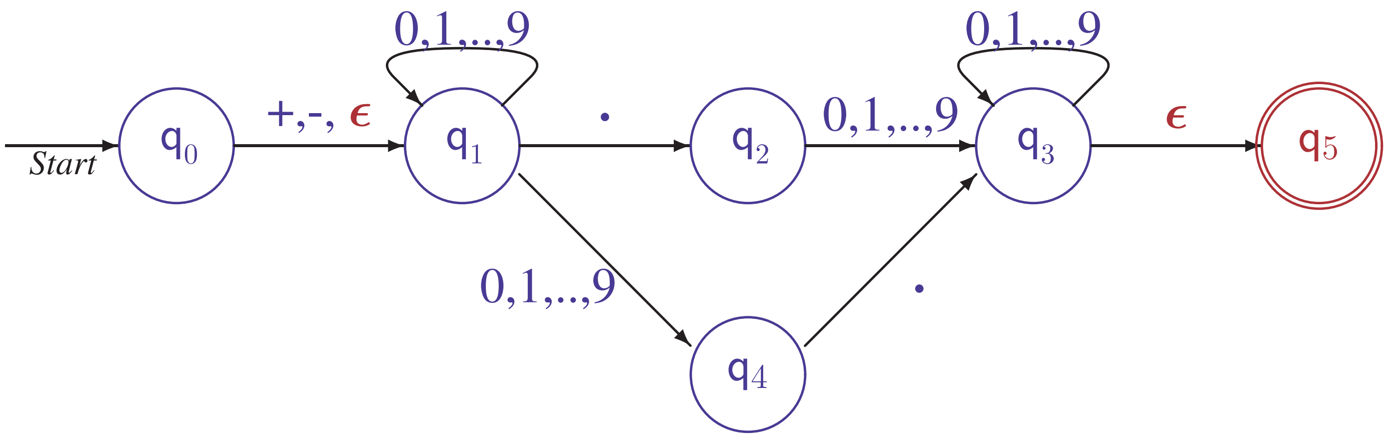
(4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3

(5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

$$\begin{tikzcd} q_0 \arrow[r, "\epsilon"] & q_1 \arrow[r, "3"] \arrow[dr] & q_1 \\ & & q_4 \end{tikzcd}$$

Arbeitsweise von NEAs mit ϵ-Übergängen



(1) Die Teilwörter +, −, und ϵ führen nach q_1

(2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4

(3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3

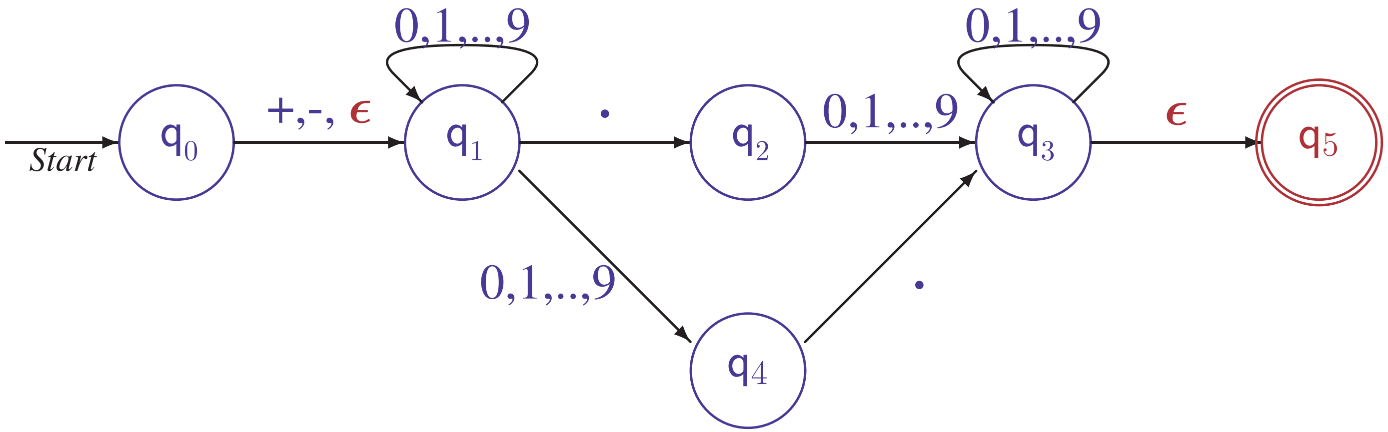
(4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3

(5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

$$\begin{tikzcd}
q_0 \arrow[r, "\epsilon"] & q_1 \arrow[r, "3"] \arrow[dr] & q_1 \arrow[r, "."] & q_2 \\
& & q_4 \arrow[r] & q_3
\end{tikzcd}$$

Arbeitsweise von NEAs mit ϵ-Übergängen



(1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1

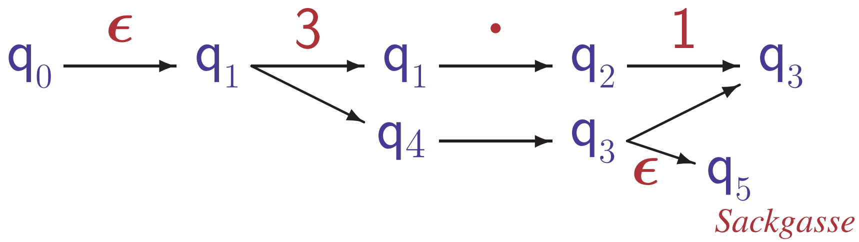
(2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4

(3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3

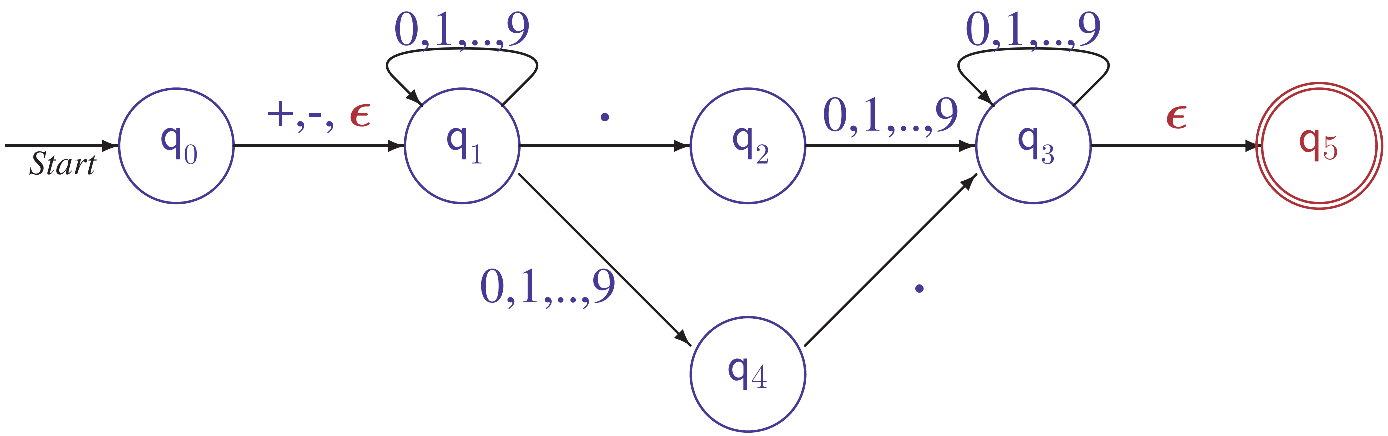
(4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3

(5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von `3.14159`



ARBEITSWEISE VON NEAS MIT ϵ-ÜBERGÄNGEN



(1) Die Teilwörter +, −, und ϵ führen nach q_1

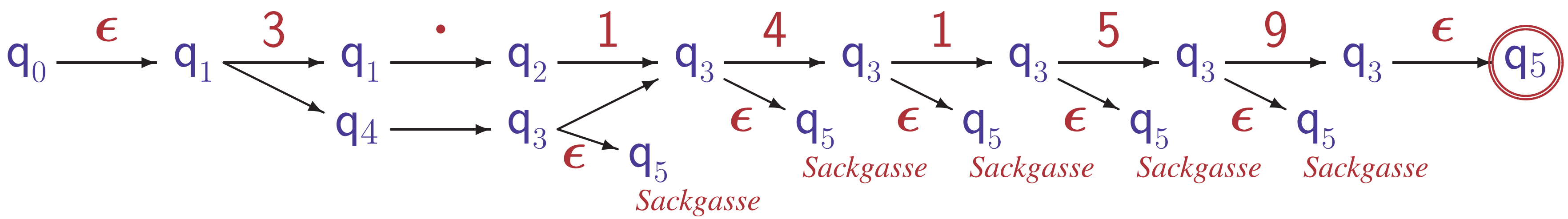
(2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4

(3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3

(4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3

(5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159



Ein Abarbeitungsweg mit ϵ-Übergängen führt zu einem Endzustand

Arbeitsweise von ϵ-NEAs – präzisiert

- **Beschreibe ϵ-Hülle eines Zustands q**
  - Die von q mit ϵ-Übergängen erreichbaren Zustände
  - Iterative Definition: Kleinste Menge mit der Eigenschaft

    $q \in \epsilon$-Hülle(q) und $p \in \epsilon$-Hülle$(q) \wedge r \in \delta(p, \epsilon) \Rightarrow r \in \epsilon$-Hülle$(q)$

ARBEITSWEISE VON ϵ-NEAs – PRÄZISIERT

- **Beschreibe ϵ-Hülle eines Zustands q**
  - Die von q mit ϵ-Übergängen erreichbaren Zustände
  - Iterative Definition: Kleinste Menge mit der Eigenschaft

    $q \in \epsilon$-Hülle(q) und $p \in \epsilon$-Hülle$(q) \wedge r \in \delta(p,\epsilon) \Rightarrow r \in \epsilon$-Hülle$(q)$

- **Erweiterte Überführungsfunktion $\hat{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)$**
  - Aufsammeln aller bei der Abarbeitung erreichbaren Zustände einschließlich derjenigen, die ohne Eingabe erreicht werden
  - Induktive Definition (kaskadisches Aufsammeln von Zuständen)

$$\hat{\delta}(q,w) = \begin{cases} \epsilon\text{-Hülle}(q) & \text{falls } w=\epsilon, \\ \bigcup_{q' \in \hat{\delta}(q,v)} \bigcup_{q'' \in \delta(q',a)} \epsilon\text{-Hülle}(q'') & \text{falls } w=v\,a\ (a \in \Sigma)\ ^* \end{cases}$$

* d.h. $p \in \hat{\delta}(q,w)$ gdw. es gibt ein $q' \in \hat{\delta}(q,v)$ und $q'' \in \delta(q',a)$ so dass $p \in \epsilon$-Hülle(q'')

Arbeitsweise von ϵ-NEAs – präzisiert

- **Beschreibe ϵ-Hülle eines Zustands q**
  - Die von q mit ϵ-Übergängen erreichbaren Zustände
  - Iterative Definition: Kleinste Menge mit der Eigenschaft
    $q \in \epsilon\text{-Hülle}(q)$ und $p \in \epsilon\text{-Hülle}(q) \wedge r \in \delta(p,\epsilon) \;\Rightarrow\; r \in \epsilon\text{-Hülle}(q)$
- **Erweiterte Überführungsfunktion $\hat{\delta} : Q \times \Sigma^* \to \mathcal{P}(Q)$**
  - Aufsammeln aller bei der Abarbeitung erreichbaren Zustände einschließlich derjenigen, die ohne Eingabe erreicht werden
  - Induktive Definition (kaskadisches Aufsammeln von Zuständen)

$$\hat{\delta}(q,w) = \begin{cases} \epsilon\text{-Hülle}(q) & \text{falls } w{=}\epsilon, \\ \bigcup_{q' \in \hat{\delta}(q,v)} \bigcup_{q'' \in \delta(q',a)} \epsilon\text{-Hülle}(q'') & \text{falls } w{=}v\,a \ (a \in \Sigma) \end{cases} \quad *$$

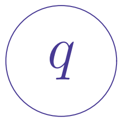
* d.h. $p \in \hat{\delta}(q,w)$ gdw. es gibt ein $q' \in \hat{\delta}(q,v)$ und $q'' \in \delta(q',a)$ so dass $p \in \epsilon\text{-Hülle}(q'')$

- **Von A akzeptierte Sprache**
  - Menge der Eingaben w, für die $\hat{\delta}(q_0,w)$ einen Endzustand enthält

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

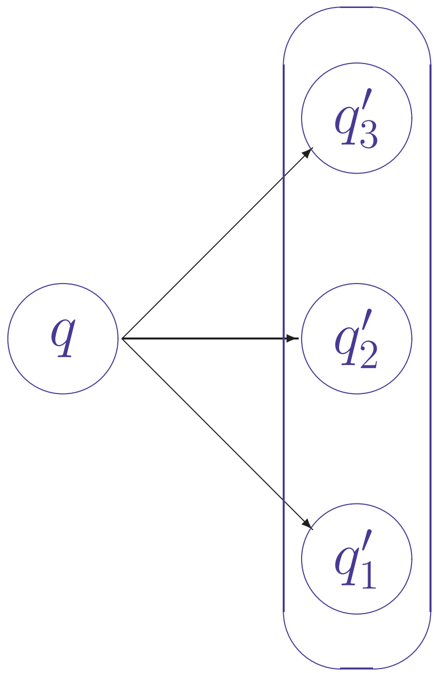
Erweiterte Überführungsfunktion

$$\hat{\delta}(q,w) = \begin{cases} \epsilon\text{-Hülle}(q) & w{=}\epsilon, \\ \bigcup_{q'\in\hat{\delta}(q,v)} \bigcup_{q''\in\delta(q',a)} \epsilon\text{-Hülle}(q'') & w{=}v\,a\ (a\in\Sigma) \end{cases} \quad *$$



Erweiterte Überführungsfunktion

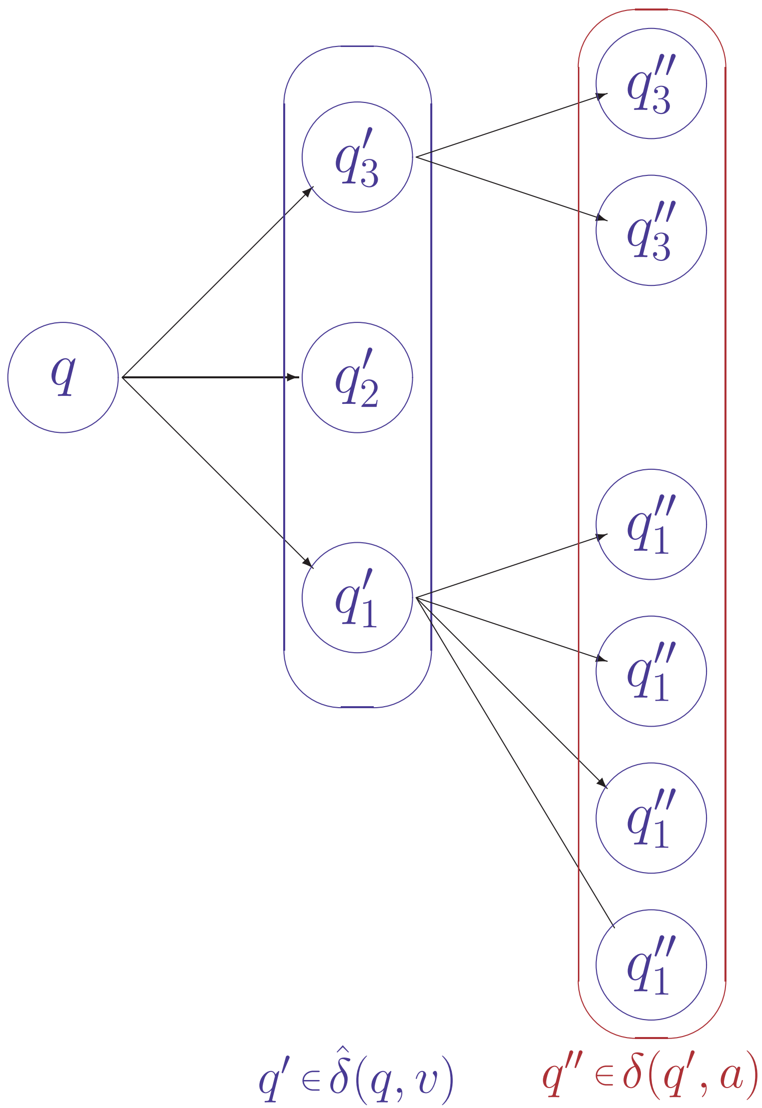
$$\hat{\delta}(q,w) = \begin{cases} \epsilon\text{-Hülle}(q) & w{=}\epsilon, \\ \bigcup_{q' \in \hat{\delta}(q,v)} \bigcup_{q'' \in \delta(q',a)} \epsilon\text{-Hülle}(q'') & w{=}v\,a\ (a \in \Sigma) \end{cases}$$ *



$q' \in \hat{\delta}(q,v)$

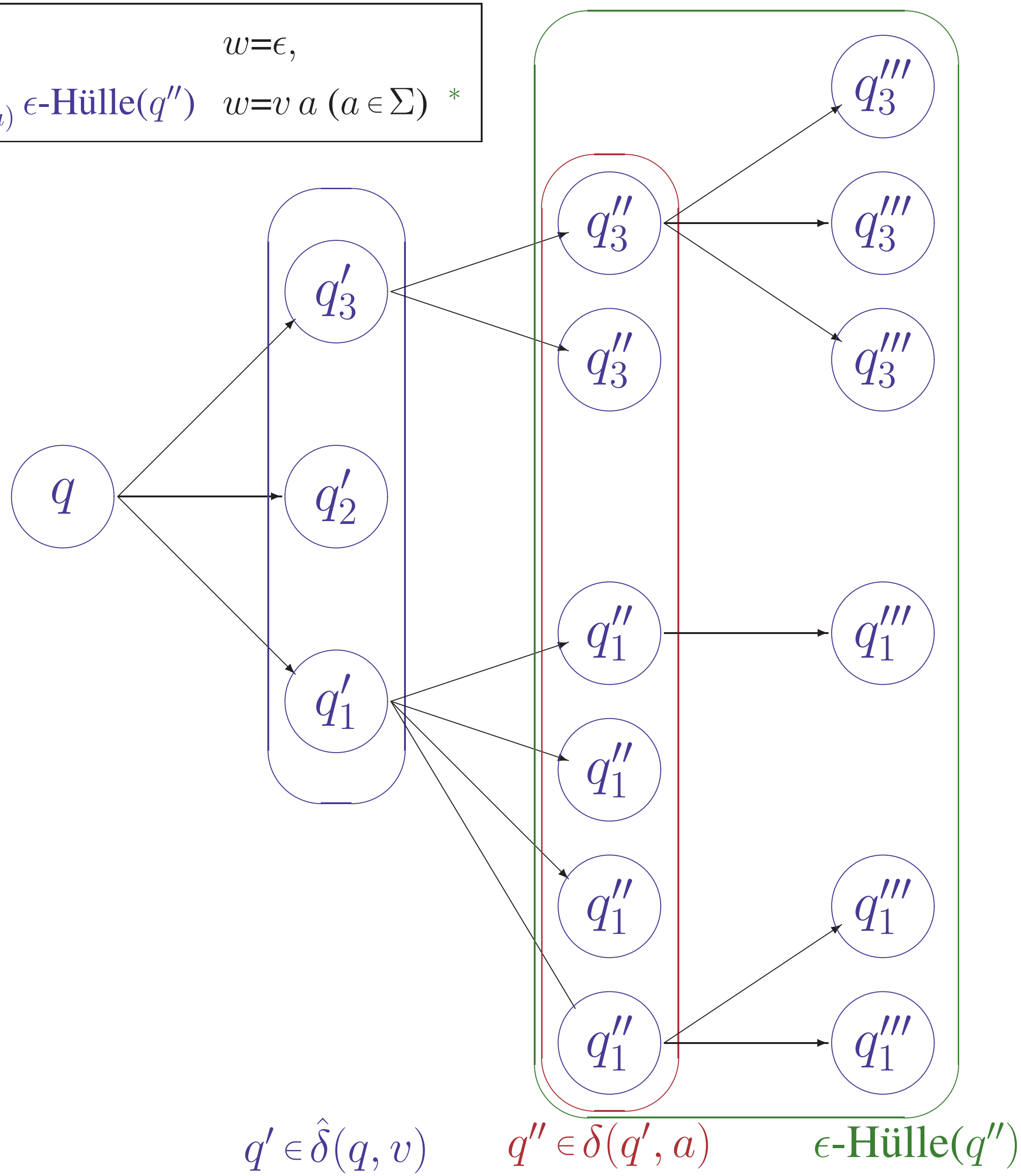
Erweiterte Überführungsfunktion

$$\hat{\delta}(q,w) = \begin{cases} \epsilon\text{-Hülle}(q) & w{=}\epsilon,\\ \bigcup_{q'\in\hat{\delta}(q,v)} \bigcup_{q''\in\delta(q',a)} \epsilon\text{-Hülle}(q'') & w{=}v\,a\ (a\in\Sigma) \end{cases} \quad *$$

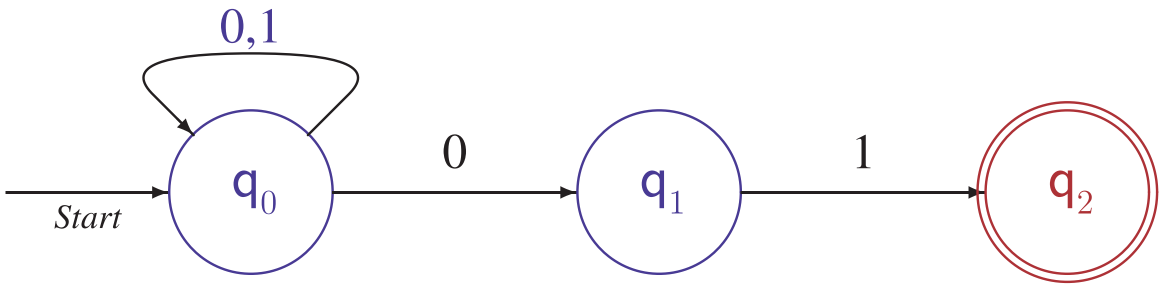


Erweiterte Überführungsfunktion

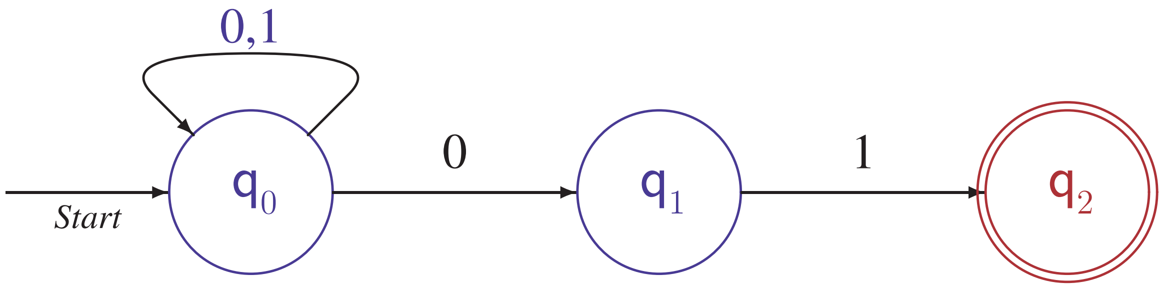
$$\hat{\delta}(q,w) = \begin{cases} \epsilon\text{-Hülle}(q) & w{=}\epsilon, \\ \bigcup_{q' \in \hat{\delta}(q,v)} \bigcup_{q'' \in \delta(q',a)} \epsilon\text{-Hülle}(q'') & w{=}v\,a\ (a \in \Sigma) \end{cases} \quad *$$



Überführungsfunktion $\hat{\delta}$ (ohne ϵ-Übergänge)

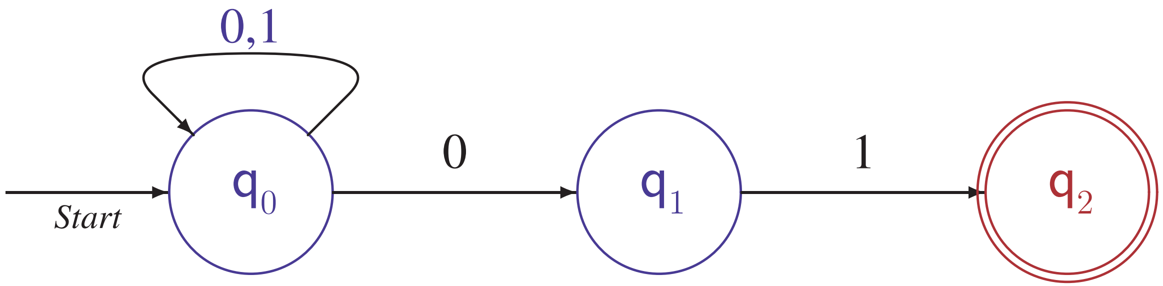


Überführungsfunktion $\hat{\delta}$ (ohne ϵ-Übergänge)



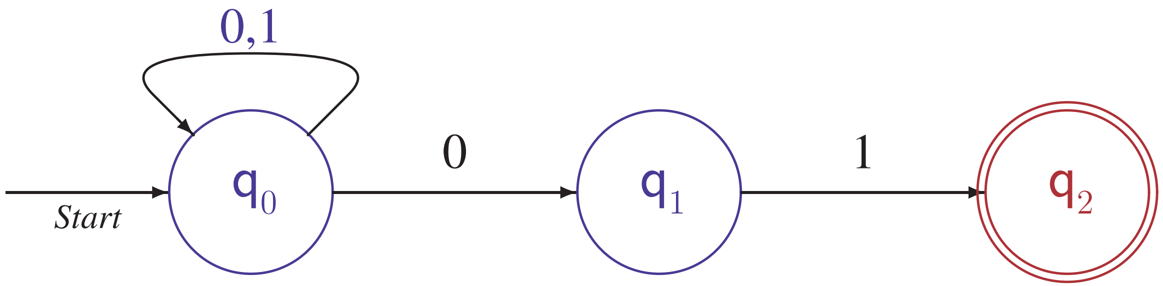
- **Abarbeitung von** `00101`
  - $\hat{\delta}(q_0, \epsilon) =$ $\{q_0\}$

ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (OHNE ϵ-ÜBERGÄNGE)



- **Abarbeitung von** `00101`
  - $\hat{\delta}(q_0, \epsilon)$ **=** $\{q_0\}$
  - $\hat{\delta}(q_0, 0) = \delta(q_0, 0)$ **=** $\{q_0, q_1\}$

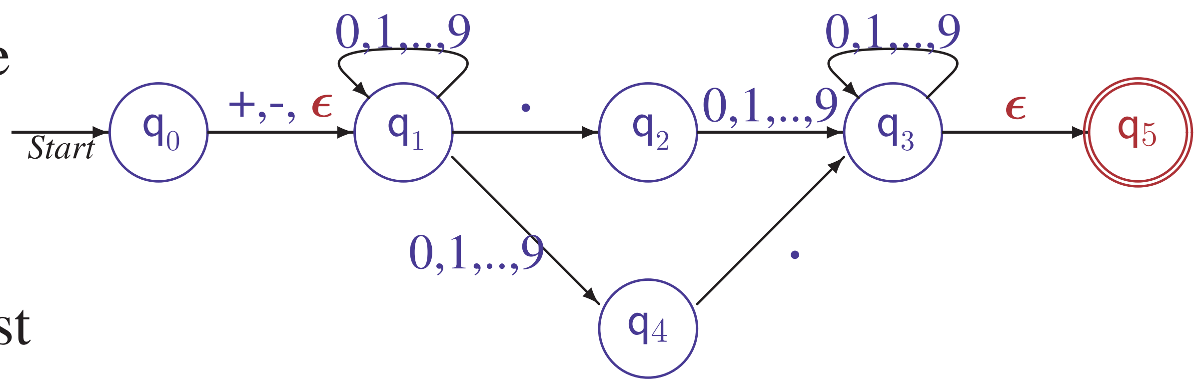
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (OHNE ϵ-ÜBERGÄNGE)



- **Abarbeitung von** `00101`
  - $\hat{\delta}(q_0, \epsilon) =$ $\{q_0\}$
  - $\hat{\delta}(q_0, 0) = \delta(q_0, 0) =$ $\{q_0, q_1\}$
  - $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset =$ $\{q_0, q_1\}$
  - $\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} =$ $\{q_0, q_2\}$
  - $\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \emptyset =$ $\{q_0, q_1\}$
  - $\hat{\delta}(q_0, 00101) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} =$ $\{q_0, q_2\}$
- `00101` **wird akzeptiert da** $\hat{\delta}(q_0, 00101) \cap F = \{q_2\}$

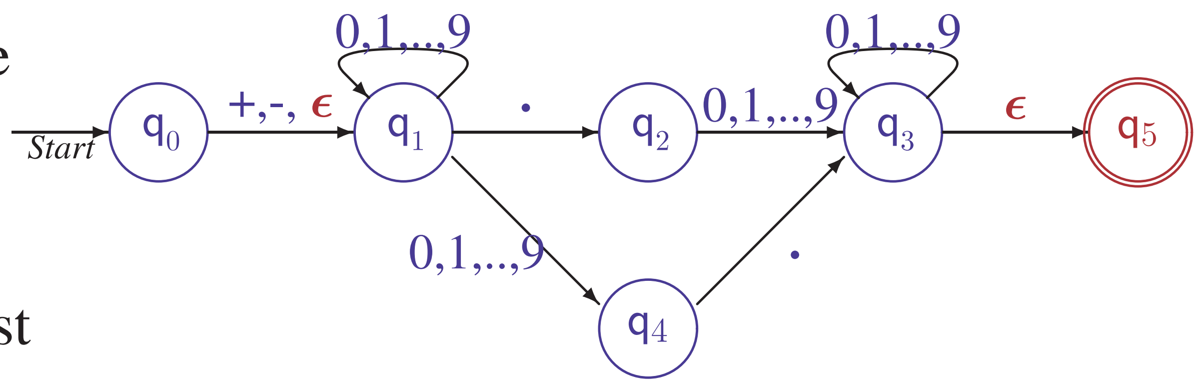
Bestimmung der ϵ-Hülle

- **Dezimalautomat**
  - Nur zwei ϵ-Übergänge
  - ϵ-Hülle$(q_0) = \{q_0, q_1\}$
  - ϵ-Hülle$(q_3) = \{q_3, q_5\}$
  - ϵ-Hülle$(q_i) = \{q_i\}$ sonst

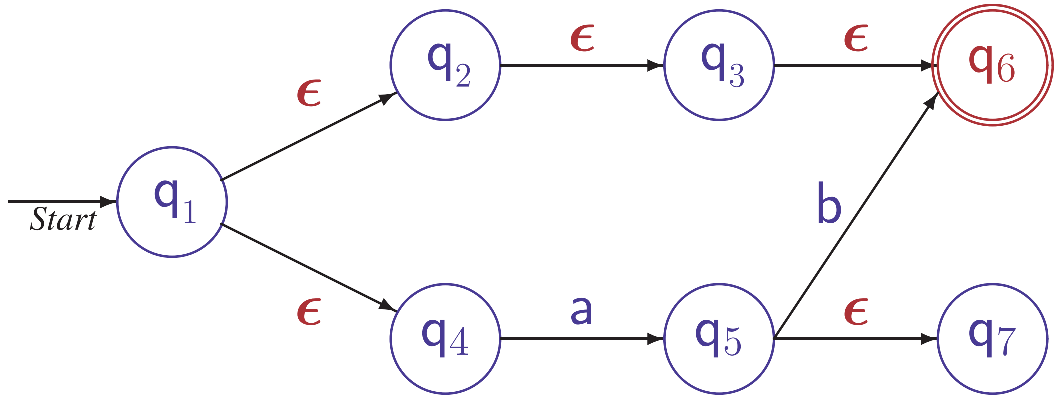


Bestimmung der ϵ-Hülle

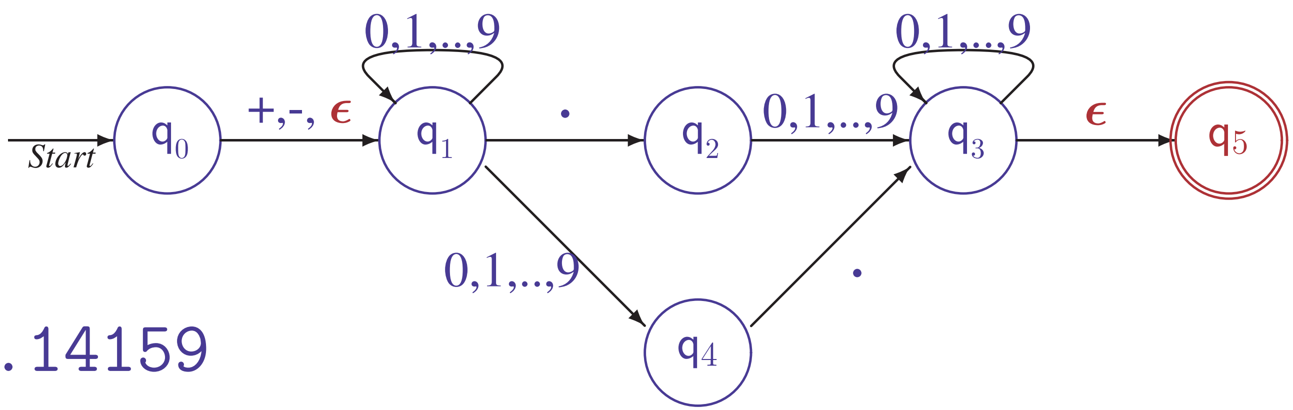
- **Dezimalautomat**
  - Nur zwei ϵ-Übergänge
  - ϵ-Hülle$(q_0) = \{q_0, q_1\}$
  - ϵ-Hülle$(q_3) = \{q_3, q_5\}$
  - ϵ-Hülle$(q_i) = \{q_i\}$ sonst



- **Viele ϵ-Übergänge**
  - ϵ-Hülle$(q_1) = \{q_1, q_2, q_3, q_4, q_6\}$
  - ϵ-Hülle$(q_2) = \{q_2, q_3, q_6\}$
  - ϵ-Hülle$(q_3) = \{q_3, q_6\}$
  - ϵ-Hülle$(q_4) = \{q_4\}$
  - ϵ-Hülle$(q_5) = \{q_5, q_7\}$
  - ϵ-Hülle$(q_6) = \{q_6\}$
  - ϵ-Hülle$(q_7) = \{q_7\}$



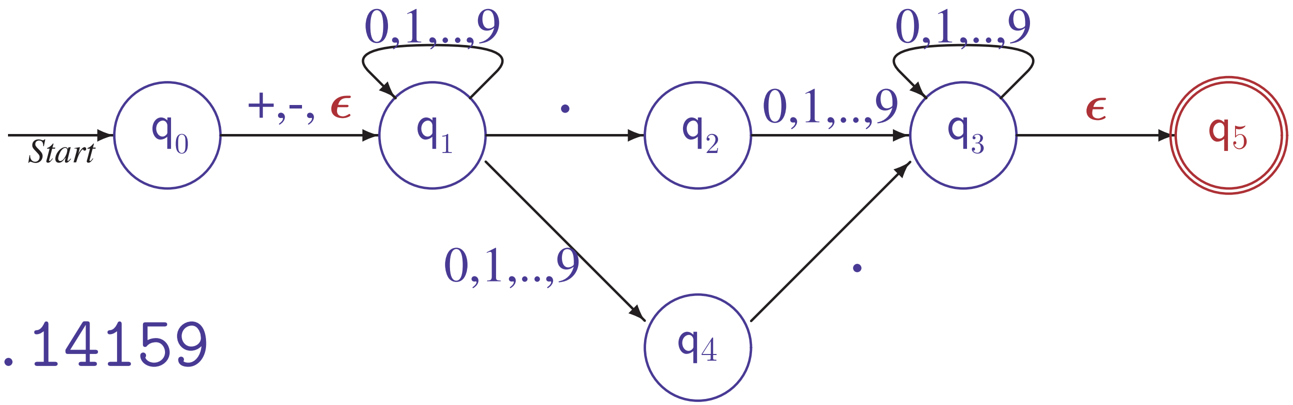
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (MIT ϵ-ÜBERGÄNGEN)



Abarbeitung von `3.14159`

– $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-Hülle}(q_0) =$ $\{q_0, q_1\}$

Überführungsfunktion $\hat{\delta}$ (mit ϵ-Übergängen)



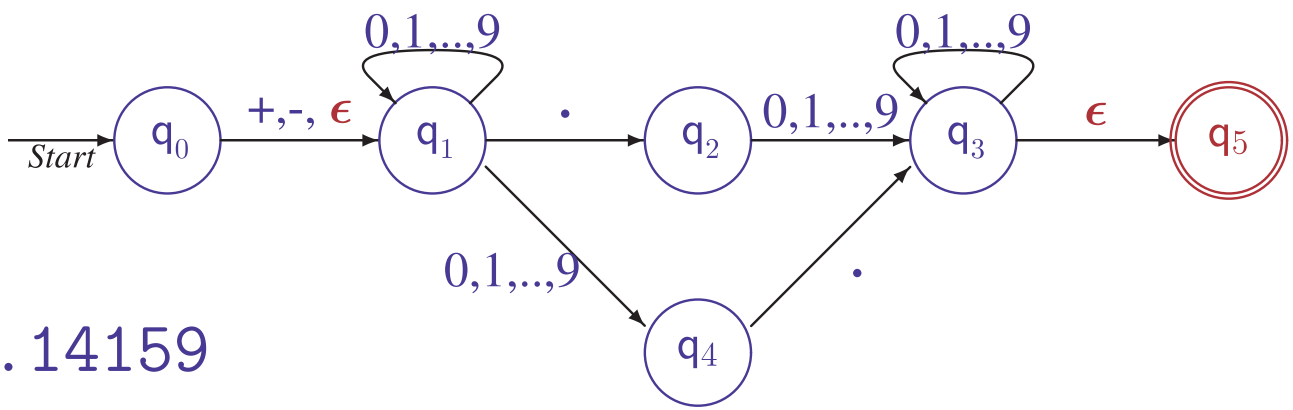
Abarbeitung von 3.14159

– $\hat{\boldsymbol{\delta}}(\boldsymbol{q_0}, \boldsymbol{\epsilon})$ = ϵ-Hülle(q_0) = $\quad\quad$ $\{\boldsymbol{q_0}, \boldsymbol{q_1}\}$

– $\hat{\delta}(q_0, 3)$: $\delta(q_0, 3) \cup \delta(q_1, 3)$ = $\emptyset \cup \{q_1, q_4\}$ = $\{q_1, q_4\}$

$\hat{\boldsymbol{\delta}}(\boldsymbol{q_0}, \boldsymbol{3})$ = ϵ-Hülle(q_1) $\cup$ ϵ-Hülle(q_4) = $\{q_1\} \cup \{q_4\}$ = $\quad\quad$ $\{\boldsymbol{q_1}, \boldsymbol{q_4}\}$

Überführungsfunktion $\hat{\delta}$ (mit ϵ-Übergängen)



Abarbeitung von `3.14159`

– $\hat{\boldsymbol{\delta}}(\boldsymbol{q_0}, \boldsymbol{\epsilon})$ = ϵ-Hülle(q_0) = $\quad \{\boldsymbol{q_0}, \boldsymbol{q_1}\}$

– $\hat{\delta}(q_0, 3)$: $\delta(q_0, 3) \cup \delta(q_1, 3) = \emptyset \cup \{q_1, q_4\} = \{q_1, q_4\}$

$\hat{\boldsymbol{\delta}}(\boldsymbol{q_0}, \mathbf{3})$ = ϵ-Hülle(q_1) $\cup$ ϵ-Hülle(q_4) = $\{q_1\} \cup \{q_4\}$ = $\quad \{\boldsymbol{q_1}, \boldsymbol{q_4}\}$

– $\hat{\delta}(q_0, 3.)$: $\delta(q_1, .) \cup \delta(q_4, .) = \{q_2\} \cup \{q_3\} = \{q_2, q_3\}$

$\hat{\boldsymbol{\delta}}(\boldsymbol{q_0}, \mathbf{3.})$ = ϵ-Hülle(q_2) $\cup$ ϵ-Hülle(q_3) = $\{q_2\} \cup \{q_3, q_5\}$ = $\quad \{\boldsymbol{q_2}, \boldsymbol{q_3}, \boldsymbol{q_5}\}$

– $\hat{\delta}(q_0, 3.1)$: $\delta(q_2, 1) \cup \delta(q_3, 1) \cup \delta(q_5, 1) = \{q_3\} \cup \{q_3\} \cup \emptyset = \{q_3\}$

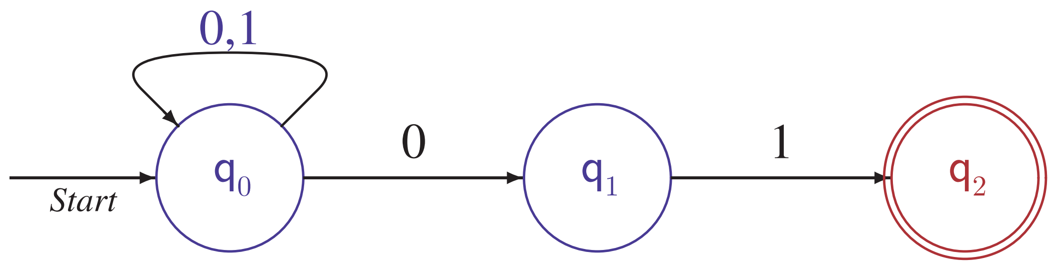
$\hat{\boldsymbol{\delta}}(\boldsymbol{q_0}, \mathbf{3.1})$ = ϵ-Hülle(q_3) = $\quad \{\boldsymbol{q_3}, \boldsymbol{q_5}\}$

– $\hat{\boldsymbol{\delta}}(\boldsymbol{q_0}, \mathbf{3.14})$ = ϵ-Hülle(q_3) = $\quad \{\boldsymbol{q_3}, \boldsymbol{q_5}\}$

$\vdots$

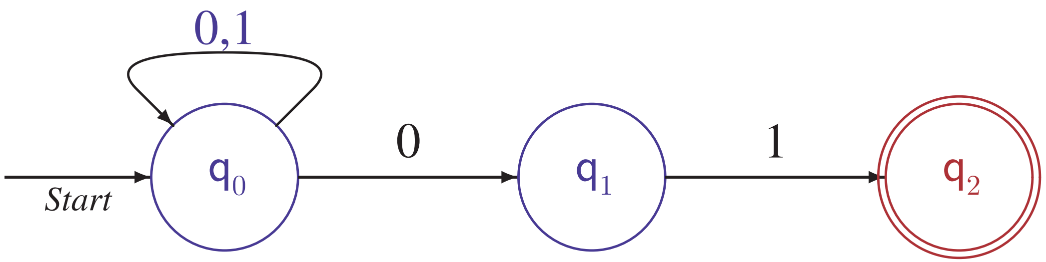
– $\hat{\boldsymbol{\delta}}(\boldsymbol{q_0}, \mathbf{3.14159})$ = ϵ-Hülle(q_3) = $\quad \{\boldsymbol{q_3}, \boldsymbol{q_5}\}$

Nachweis der erkannten Sprache (ohne ϵ-Übergänge)



$$L(A) = \{w \in \{0,1\}^* \mid w \text{ endet mit } 01\}$$

Nachweis der erkannten Sprache (ohne ϵ-Übergänge)



$$L(A) = \{w \in \{0,1\}^* \mid w \text{ endet mit } 01\}$$

- **Zeige durch simultane Induktion für alle $w \in \{0,1\}^*$**

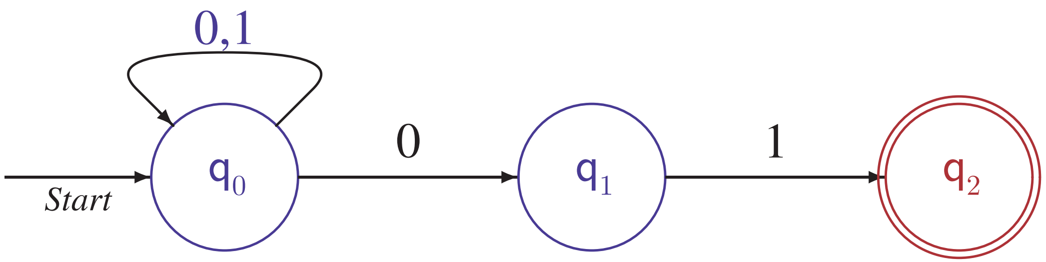
  $B_0(w)$: $q_0 \in \hat{\delta}(q_0, w)$ für alle w ($\hat{\delta}(q_0, w)$ muß nicht genau bestimmt werden!)

  $B_1(w)$: $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet

  $B_2(w)$: $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

  Dann folgt $w \in L(A) \Leftrightarrow \hat{\delta}(q_0, w) \cap \{q_2\} \neq \emptyset \Leftrightarrow$ **w endet mit 01**

NACHWEIS DER ERKANNTEN SPRACHE (OHNE ϵ-ÜBERGÄNGE)



$$L(A) = \{w \in \{0,1\}^* \mid w \text{ endet mit } 01\}$$

- **Zeige durch simultane Induktion für alle $w \in \{0,1\}^*$**

  $B_0(w)$: $q_0 \in \hat{\delta}(q_0, w)$ für alle w ($\hat{\delta}(q_0, w)$ muß nicht genau bestimmt werden!)

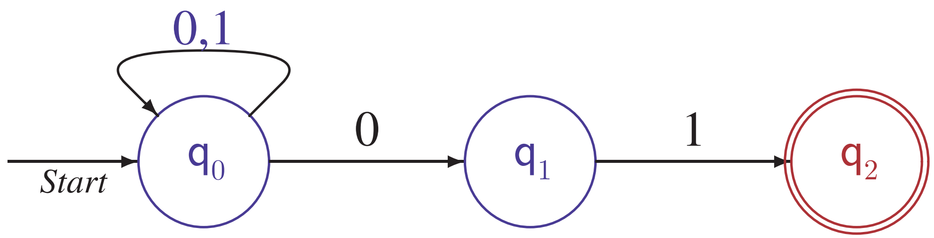
  $B_1(w)$: $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet

  $B_2(w)$: $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

  Dann folgt $w \in L(A) \Leftrightarrow \hat{\delta}(q_0, w) \cap \{q_2\} \neq \emptyset \Leftrightarrow$ **w endet mit 01**

- **Induktionsanfang:** $w = \epsilon$
  - Per Definition ist $\hat{\delta}(q_0, \epsilon) = \{q_0\}$. Also gilt Behauptung $B_0(w)$ ✓
  - w endet nicht mit 0 oder 01. $B_1(w)$ und $B_2(w)$ gelten trivialerweise ✓

Nachweis der erkannten Sprache II (semiformal)

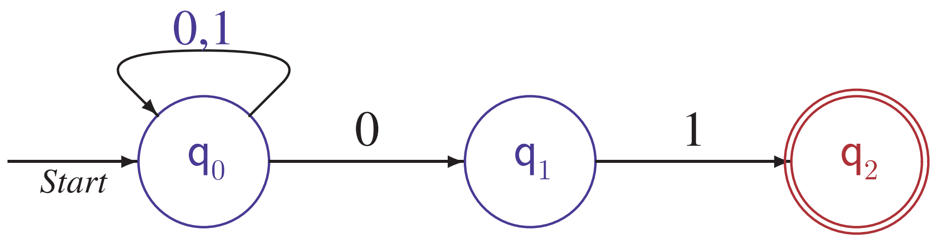


$B_0(w)$: $q_0 \in \hat\delta(q_0, w)$
$B_1(w)$: $q_1 \in \hat\delta(q_0, w)$ genau dann, wenn w mit 0 endet
$B_2(w)$: $q_2 \in \hat\delta(q_0, w)$ genau dann, wenn w mit 01 endet

Annahme: $B_0(w')$–$B_2(w')$ seien für $w' \in \{0, 1\}^*$ gezeigt

Induktionsschritt: Es sei $w = w'a$ für ein $a \in \{0, 1\}$

Nachweis der erkannten Sprache II (semiformal)



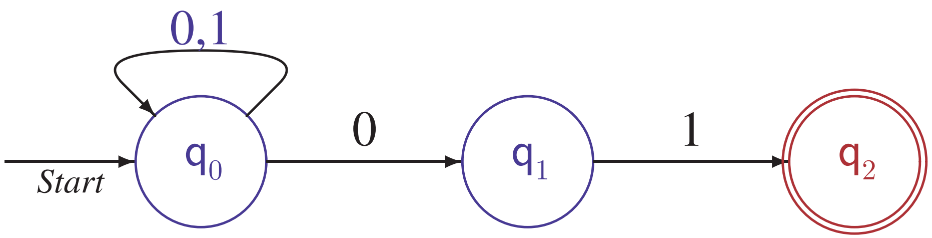
$B_0(w)$: $q_0 \in \hat\delta(q_0, w)$
$B_1(w)$: $q_1 \in \hat\delta(q_0, w)$ genau dann, wenn w mit 0 endet
$B_2(w)$: $q_2 \in \hat\delta(q_0, w)$ genau dann, wenn w mit 01 endet

Annahme: $B_0(w')$–$B_2(w')$ seien für $w' \in \{0, 1\}^*$ gezeigt

Induktionsschritt: Es sei $w = w'a$ für ein $a \in \{0, 1\}$

$\boldsymbol{B_0(w)}$: Wegen $q_0 \in \hat\delta(q_0, w')$ und $q_0 \in \delta(q_0, a)$ für $a \in \{0, 1\}$ folgt $q_0 \in \hat\delta(q_0, w)$ ✓

Nachweis der erkannten Sprache II (semiformal)



$B_0(w)$: $q_0 \in \hat\delta(q_0, w)$
$B_1(w)$: $q_1 \in \hat\delta(q_0, w)$ genau dann, wenn w mit 0 endet
$B_2(w)$: $q_2 \in \hat\delta(q_0, w)$ genau dann, wenn w mit 01 endet

Annahme: $B_0(w')$–$B_2(w')$ seien für $w' \in \{0, 1\}^*$ gezeigt

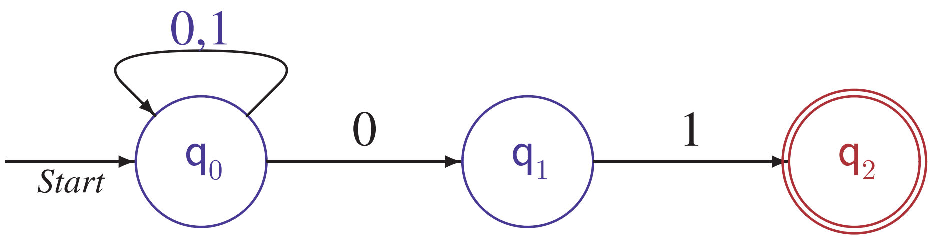
Induktionsschritt: Es sei $w = w'a$ für ein $a \in \{0, 1\}$

- $\boldsymbol{B_0(w)}$: Wegen $q_0 \in \hat\delta(q_0, w')$ und $q_0 \in \delta(q_0, a)$ für $a \in \{0, 1\}$ folgt $q_0 \in \hat\delta(q_0, w)$ ✓
- $\boldsymbol{B_1(w)}$: Sei $q_1 \in \hat\delta(q_0, w)$. Wegen $q_1 \in \delta(q, a) \Leftrightarrow q{=}q_0 \wedge a{=}0$ muss w mit 0 enden

  Wenn umgekehrt w mit 0 endet, dann ist $a{=}0$.

  Wegen $q_0 \in \hat\delta(q_0, w')$ und $q_1 \in \delta(q_0, a)$ folgt $q_1 \in \hat\delta(q_0, w)$ ✓

Nachweis der erkannten Sprache II (semiformal)



$B_0(w)$: $q_0 \in \hat\delta(q_0, w)$
$B_1(w)$: $q_1 \in \hat\delta(q_0, w)$ genau dann, wenn w mit 0 endet
$B_2(w)$: $q_2 \in \hat\delta(q_0, w)$ genau dann, wenn w mit 01 endet

Annahme: $B_0(w')$–$B_2(w')$ seien für $w' \in \{0,1\}^*$ gezeigt

Induktionsschritt: Es sei $w = w'a$ für ein $a \in \{0,1\}$

$B_0(w)$: Wegen $q_0 \in \hat\delta(q_0, w')$ und $q_0 \in \delta(q_0, a)$ für $a \in \{0,1\}$ folgt $q_0 \in \hat\delta(q_0, w)$ ✓

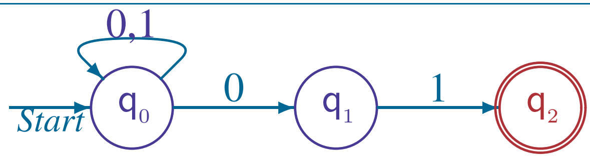
$B_1(w)$: Sei $q_1 \in \hat\delta(q_0, w)$. Wegen $q_1 \in \delta(q, a) \Leftrightarrow q{=}q_0 \wedge a{=}0$ muss w mit 0 enden
Wenn umgekehrt w mit 0 endet, dann ist $a{=}0$.
Wegen $q_0 \in \hat\delta(q_0, w')$ und $q_1 \in \delta(q_0, a)$ folgt $q_1 \in \hat\delta(q_0, w)$ ✓

$B_2(w)$: Sei $q_2 \in \hat\delta(q_0, w)$. Wegen $q_2 \in \delta(q, a) \Leftrightarrow q{=}q_1 \wedge a{=}1$ muss w mit 1 enden
und $q_1 \in \hat\delta(q_0, w')$ gelten. Wegen $B_1(w')$ endet w' mit 0, also w mit 01
Wenn umgekehrt w mit 01 endet, dann ist $a{=}1$ und w' endet mit 0.
Wegen $q_1 \in \hat\delta(q_0, w')$ nach $B_1(w')$ und $q_2 \in \delta(q_1, a)$ folgt $q_2 \in \hat\delta(q_0, w)$ ✓

Arbeitsweise von ϵ-NEAs – Alternative Beschreibung mit Konfigurationsübergängen

- **Definiere Konfigurationen**
  – Formal dargestellt als Tupel $\boldsymbol{K} = (\boldsymbol{q},\boldsymbol{w}) \in Q\times\Sigma^*$
- **Definiere Konfigurationsübergangsrelation** $\overset{*}{\vdash}$
  – Wechsel zwischen Konfigurationen durch Abarbeitung von Wörtern
  – $(\boldsymbol{q},\boldsymbol{aw}) \vdash (\boldsymbol{p},\boldsymbol{w})$, falls $p\in\delta(q,a)$
  – $(\boldsymbol{q},\boldsymbol{w}) \vdash (\boldsymbol{p},\boldsymbol{w})$, falls $p\in\delta(q,\epsilon)$
  – $\boldsymbol{K_1} \overset{*}{\vdash} \boldsymbol{K_2}$, falls $K_1 = K_2$ oder
  es gibt eine Konfiguration K mit $K_1 \overset{*}{\vdash} K$ und $K \vdash K_2$
- **Akzeptierte Sprache**
  – Menge der Eingaben, für die $\overset{*}{\vdash}$ zu akzeptierenden Zustand führt

$$\boldsymbol{L(A)} = \{\boldsymbol{w}\in\Sigma^* \mid \exists \boldsymbol{p}\in \boldsymbol{F}.\ (\boldsymbol{q_0},\boldsymbol{w}) \overset{*}{\vdash} (\boldsymbol{p},\boldsymbol{\epsilon})\}$$

Beweis für  mit Konfigurationen

(Dieses Mal etwas formaler)

$$L(A) = \{w \in \{0,1\}^* \mid \exists v \in \{0,1\}^*.\ w{=}v01\}$$

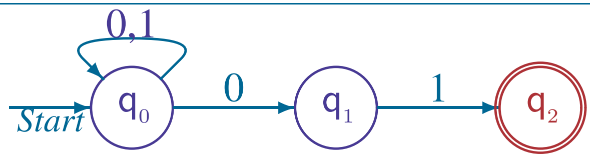
- **Zeige durch simultane Induktion für alle $w \in \{0,1\}^*$**

  $B_0(w)$: $(q_0, w) \vdash^* (q_0, \epsilon)$

  $B_1(w)$: $(q_0, w) \vdash^* (q_1, \epsilon) \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v0$

  $B_2(w)$: $(q_0, w) \vdash^* (q_2, \epsilon) \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v01$

  Formulierung ungeeignet für Induktionsbeweise

Beweis für  mit Konfigurationen

(Dieses Mal etwas formaler)

$$L(A) = \{w \in \{0,1\}^* \mid \exists v \in \{0,1\}^*.\ w{=}v01\}$$

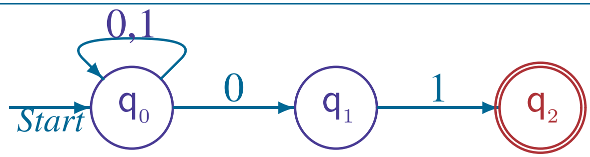
- **Zeige durch simultane Induktion für alle w, $v \in \{0,1\}^*$**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v)$

  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v0$

  $B_2(w)$: $(q_0, w) \vdash^* (q_2, \epsilon) \quad \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v01$

  Besser: erlaube, daß nach Abarbeitung von w ein Restwort v noch ungelesen ist

Beweis für  mit Konfigurationen

(Dieses Mal etwas formaler)

$$L(A) = \{w \in \{0,1\}^* \mid \exists v \in \{0,1\}^*.\ w{=}v01\}$$

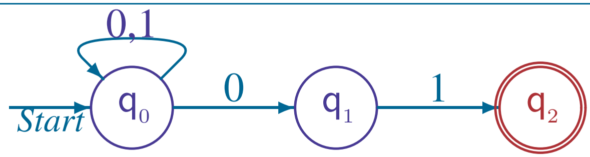
- **Zeige durch simultane Induktion für alle w, $v \in \{0,1\}^*$**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v)$

  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v0$

  $B_2(w)$: $(q_0, w) \vdash^* (q_2, \epsilon) \quad \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v01$

- **Induktionsanfang $w = \epsilon$**
  - Per Definition ist $(q_0, v) \vdash^* (q_0, v)$, $(q_0, v) \not\vdash^* (q_{1/2}, v)$ und $\epsilon{\neq}v0$, $\epsilon{\neq}v01$ $\surd$

Beweis für  mit Konfigurationen

(Dieses Mal etwas formaler)

$$L(A) = \{w \in \{0,1\}^* \mid \exists v \in \{0,1\}^*.\ w{=}v01\}$$

- **Zeige durch simultane Induktion für alle w, $v \in \{0,1\}^*$**

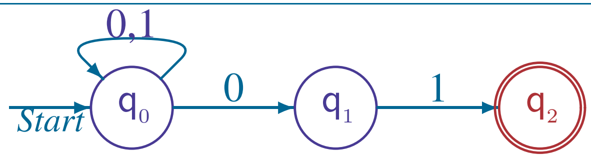
  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v)$

  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v0$

  $B_2(w)$: $(q_0, w) \vdash^* (q_2, \epsilon) \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v01$

- **Induktionsanfang $w = \epsilon$**
  - Per Definition ist $(q_0, v) \vdash^* (q_0, v)$, $(q_0, v) \not\vdash^* (q_{1/2}, v)$ und $\epsilon{\neq}v0$, $\epsilon{\neq}v01$ $\checkmark$

- **Induktionsannahme: Sei $B_0(w')..B_2(w')$ bewiesen für $w' \in \{0,1\}^*$**

Beweis für  mit Konfigurationen

(Dieses Mal etwas formaler)

$$L(A) = \{w \in \{0,1\}^* \mid \exists v \in \{0,1\}^*.\ w{=}v01\}$$

- **Zeige durch simultane Induktion für alle w, $v \in \{0,1\}^*$**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v)$

  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v0$

  $B_2(w)$: $(q_0, w) \vdash^* (q_2, \epsilon) \quad \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v01$

- **Induktionsanfang $w = \epsilon$**
  - Per Definition ist $(q_0, v) \vdash^* (q_0, v)$, $(q_0, v) \not\vdash^* (q_{1/2}, v)$ und $\epsilon{\neq}v0$, $\epsilon{\neq}v01$ $\quad\checkmark$

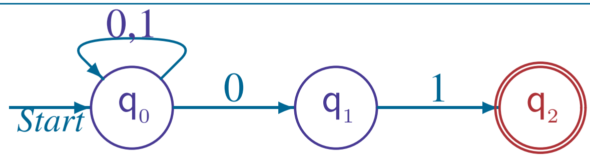
- **Induktionsannahme: Sei $B_0(w')..B_2(w')$ bewiesen für $w' \in \{0,1\}^*$**

- **Induktionsschritt: Sei $w = w'a$ für ein $a \in \{0,1\}$**

  $B_0(w)$: Wegen $B_0(w')$ gilt $(q_0, w\,v) = (q_0, w'av) \vdash^* (q_0, a\,v) \vdash (q_0, v)$ $\quad\checkmark$

  $B_1(w)$: Wegen $(p, av) \vdash (q_1, v) \Leftrightarrow p{=}q_0 \wedge a{=}0$ und $B_0(w')$ gilt
  $(q_0, wv) \vdash^* (q_1, v) \;\Leftrightarrow\; (q_0, w'av) \vdash^* (q_0, av) \vdash (q_1, v) \;\Leftrightarrow\; \exists v \in \{0,1\}^*. w{=}v0$ $\quad\checkmark$

  $B_2(w)$: Wegen $(p, av) \vdash (q_2, v) \Leftrightarrow p{=}q_1 \wedge a{=}1$ und $B_1(w')$ gilt
  $(q_0, w) \vdash^* (q_2, \epsilon) \;\Leftrightarrow\; (q_0, w'a) \vdash^* (q_1, a) \vdash (q_2, \epsilon) \;\Leftrightarrow\; \exists v \in \{0,1\}^*. w{=}w'1{=}v01$ $\quad\checkmark$

Beweis für  mit Konfigurationen

(Dieses Mal etwas formaler)

$$L(A) = \{w \in \{0,1\}^* \mid \exists v \in \{0,1\}^*.\ w{=}v01\}$$

- **Zeige durch simultane Induktion für alle w, $v \in \{0,1\}^*$**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v)$

  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v0$

  $B_2(w)$: $(q_0, w) \vdash^* (q_2, \epsilon) \quad \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v01$

- **Induktionsanfang $w = \epsilon$**
  – Per Definition ist $(q_0, v) \vdash^* (q_0, v)$, $(q_0, v) \not\vdash^* (q_{1/2}, v)$ und $\epsilon{\neq}v0$, $\epsilon{\neq}v01$ $\checkmark$

- **Induktionsannnahme: Sei $B_0(w')..B_2(w')$ bewiesen für $w' \in \{0,1\}^*$**

- **Induktionsschritt: Sei $w = w'a$ für ein $a \in \{0,1\}$**

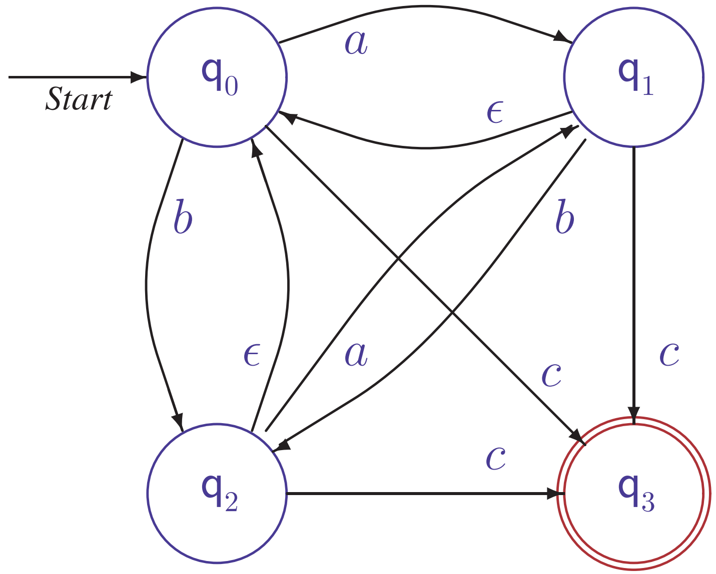
  $B_0(w)$: Wegen $B_0(w')$ gilt $(q_0, w\,v) = (q_0, w'av) \vdash^* (q_0, a\,v) \vdash (q_0, v)$ $\checkmark$

  $B_1(w)$: Wegen $(p, av) \vdash (q_1, v) \Leftrightarrow p{=}q_0 \wedge a{=}0$ und $B_0(w')$ gilt
  $(q_0, wv) \vdash^* (q_1, v) \Leftrightarrow (q_0, w'av) \vdash^* (q_0, av) \vdash (q_1, v) \Leftrightarrow \exists v \in \{0,1\}^*.w{=}v0$ $\checkmark$

  $B_2(w)$: Wegen $(p, av) \vdash (q_2, v) \Leftrightarrow p{=}q_1 \wedge a{=}1$ und $B_1(w')$ gilt
  $(q_0, w) \vdash^* (q_2, \epsilon) \Leftrightarrow (q_0, w'a) \vdash^* (q_1, a) \vdash (q_2, \epsilon) \Leftrightarrow \exists v \in \{0,1\}^*.w{=}w'1{=}v01$ $\checkmark$

- **Es folgt** $w \in L(A) \Leftrightarrow (q_0, w) \vdash^* (q_2, \epsilon) \Leftrightarrow \exists v \in \{0,1\}^*.\ w{=}v01$

Nachweis der erkannten Sprache (mit ϵ-Übergängen)



Nachweis der erkannten Sprache (mit ϵ-Übergängen)

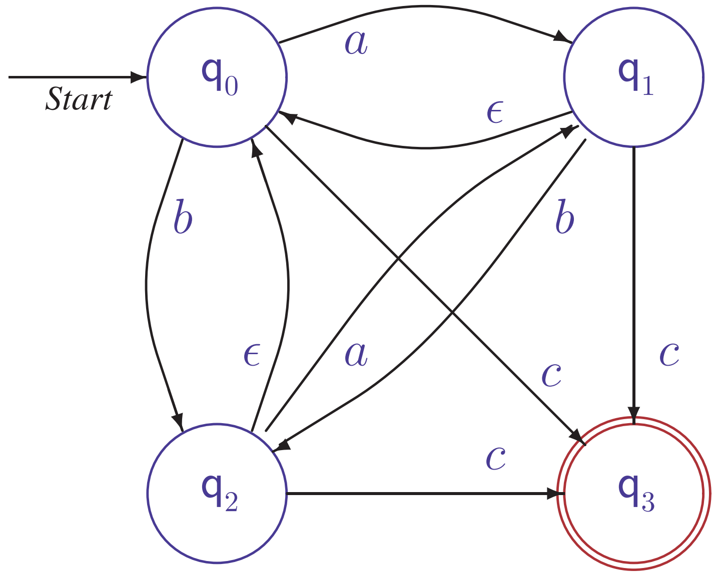
- **Mit welchen Wörtern werden die q_i erreicht?**

  q_0: Gleiche Wörter wie q_1 oder q_2

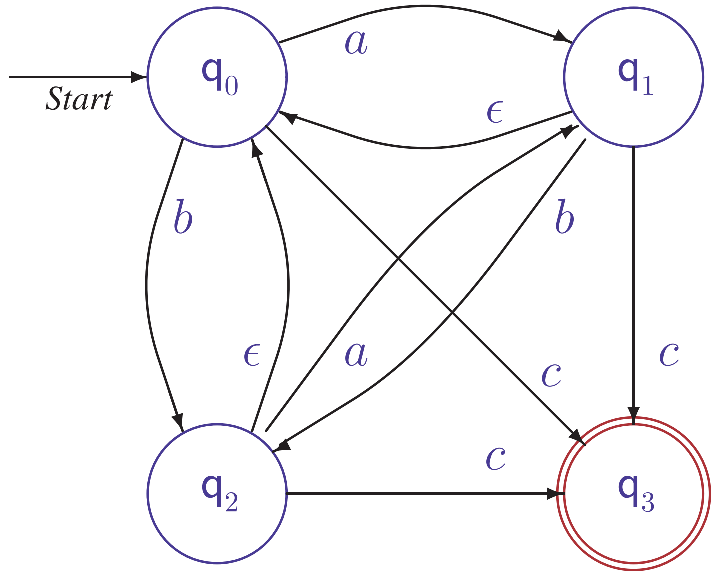
  q_1: Letztes Symbol ist ein a,

  q_2: Letztes Symbol ist ein b

  q_3: Letztes Symbol ist ein c



Nachweis der erkannten Sprache (mit ϵ-Übergängen)



- **Mit welchen Wörtern werden die q_i erreicht?**

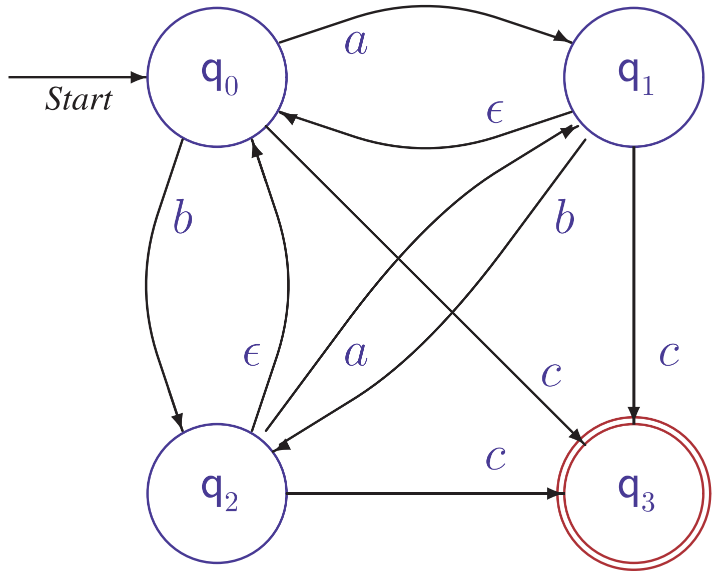
  q_0: Gleiche Wörter wie q_1 oder q_2

  q_1: Letztes Symbol ist ein a, davor b's oder a's

  q_2: Letztes Symbol ist ein b

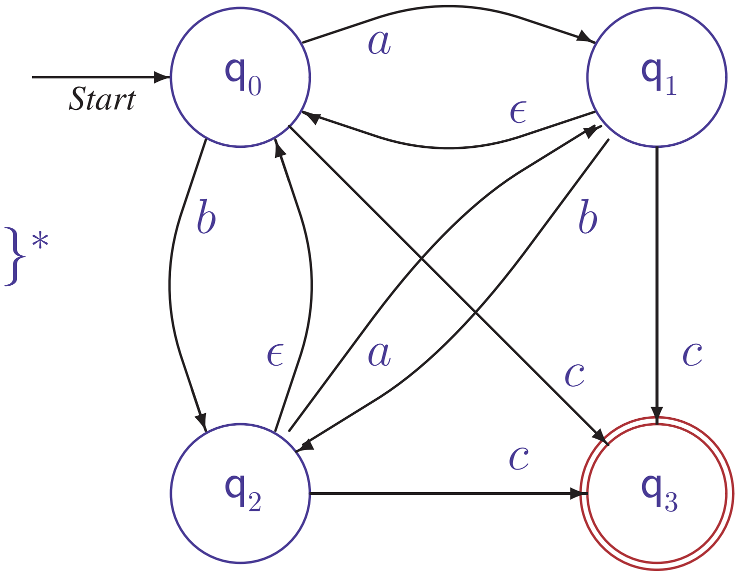
  q_3: Letztes Symbol ist ein c

Nachweis der erkannten Sprache (mit ϵ-Übergängen)



- **Mit welchen Wörtern werden die q_i erreicht?**
  q_0: Gleiche Wörter wie q_1 oder q_2
  q_1: Letztes Symbol ist ein a, davor b's oder a's
  q_2: Letztes Symbol ist ein b, davor a's oder b's
  q_3: Letztes Symbol ist ein c

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ-ÜBERGÄNGEN)



- **Mit welchen Wörtern werden die q_i erreicht?**

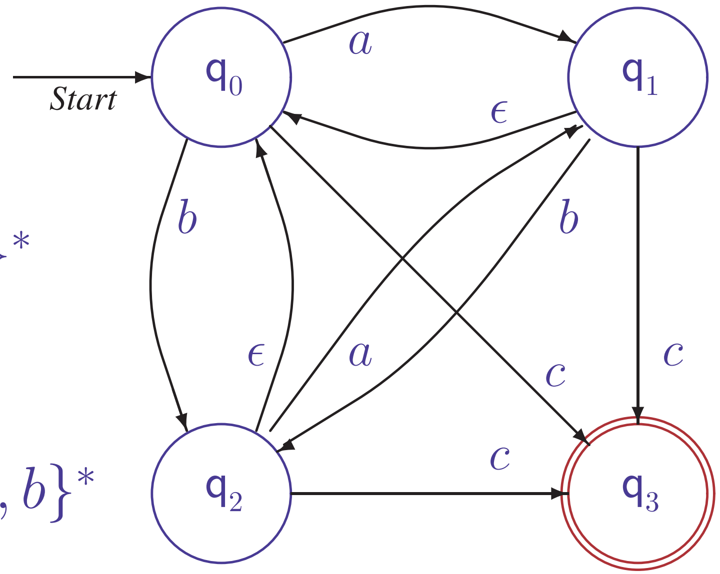
  q_0: Gleiche Wörter wie q_1 oder q_2 und ϵ, also $\{a, b\}^*$

  q_1: Letztes Symbol ist ein a, davor b's oder a's

  q_2: Letztes Symbol ist ein b, davor a's oder b's

  q_3: Letztes Symbol ist ein c

Nachweis der erkannten Sprache (mit ϵ-Übergängen)



- **Mit welchen Wörtern werden die q_i erreicht?**

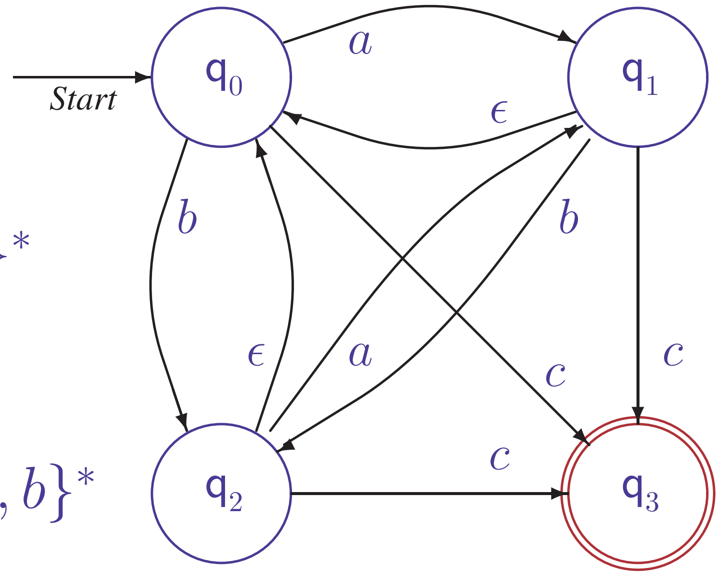
  q_0: Gleiche Wörter wie q_1 oder q_2 und ϵ, also $\{a, b\}^*$

  q_1: Letztes Symbol ist ein a, davor b's oder a's

  q_2: Letztes Symbol ist ein b, davor a's oder b's

  q_3: Letztes Symbol ist ein c, davor ein Wort aus $\{a, b\}^*$

Nachweis der erkannten Sprache (mit ϵ-Übergängen)



- **Mit welchen Wörtern werden die q_i erreicht?**

  q_0: Gleiche Wörter wie q_1 oder q_2 und ϵ, also $\{a,b\}^*$

  q_1: Letztes Symbol ist ein a, davor b's oder a's

  q_2: Letztes Symbol ist ein b, davor a's oder b's

  q_3: Letztes Symbol ist ein c, davor ein Wort aus $\{a,b\}^*$

- **Formuliere Aussage mit Konfigurationen**

  $B_0(w)$: $(q_0, w) \vdash^* (q_0, \epsilon) \;\Leftrightarrow\; w \in \{a,b\}^*$

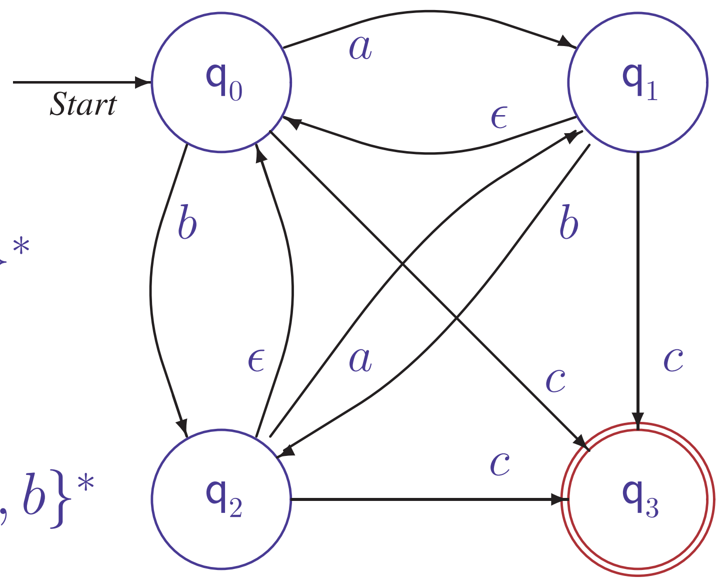
  $B_1(w)$: $(q_0, w) \vdash^* (q_1, \epsilon) \;\Leftrightarrow\; w = ua$ für ein $u \in \{a,b\}^*$

  $B_2(w)$: $(q_0, w) \vdash^* (q_2, \epsilon) \;\Leftrightarrow\; w = ub$ für ein $u \in \{a,b\}^*$

  $B_3(w)$: $(q_0, w) \vdash^* (q_3, \epsilon) \;\Leftrightarrow\; w = uc$ für ein $u \in \{a,b\}^*$

  Formulierung ungeeignet für Induktionsbeweise

Nachweis der erkannten Sprache (mit ϵ-Übergängen)



- **Mit welchen Wörtern werden die q_i erreicht?**

  q_0: Gleiche Wörter wie q_1 oder q_2 und ϵ, also $\{a, b\}^*$

  q_1: Letztes Symbol ist ein a, davor b's oder a's

  q_2: Letztes Symbol ist ein b, davor a's oder b's

  q_3: Letztes Symbol ist ein c, davor ein Wort aus $\{a, b\}^*$

- **Formuliere Aussage mit Konfigurationen**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \;\Leftrightarrow\; w \in \{a, b\}^*$

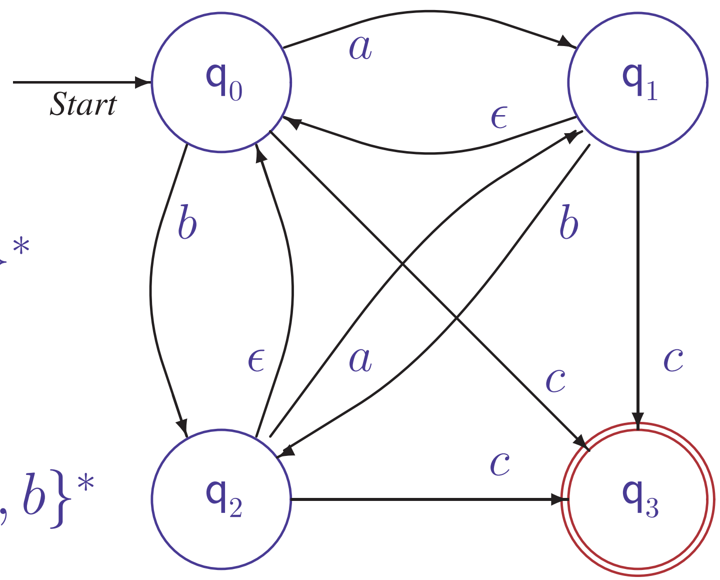
  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \;\Leftrightarrow\; w = ua$ für ein $u \in \{a, b\}^*$

  $B_2(w)$: $(q_0, w\,v) \vdash^* (q_2, v) \;\Leftrightarrow\; w = ub$ für ein $u \in \{a, b\}^*$

  $B_3(w)$: $(q_0, w) \vdash^* (q_3, \epsilon) \;\Leftrightarrow\; w = uc$ für ein $u \in \{a, b\}^*$

  Besser: erlaube, daß nach Abarbeitung von w ein Restwort v noch ungelesen ist

Nachweis der erkannten Sprache (mit ϵ-Übergängen)



- **Mit welchen Wörtern werden die q_i erreicht?**

  q_0: Gleiche Wörter wie q_1 oder q_2 und ϵ, also $\{a,b\}^*$

  q_1: Letztes Symbol ist ein a, davor b's oder a's

  q_2: Letztes Symbol ist ein b, davor a's oder b's

  q_3: Letztes Symbol ist ein c, davor ein Wort aus $\{a,b\}^*$

- **Formuliere Aussage mit Konfigurationen**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \;\Leftrightarrow\; w \in \{a,b\}^*$

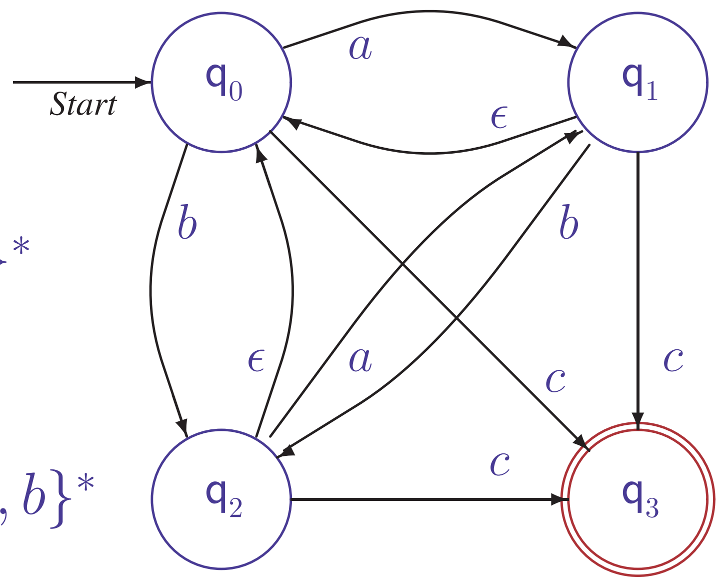
  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \;\Leftrightarrow\; w = ua$ für ein $u \in \{a,b\}^*$

  $B_2(w)$: $(q_0, w\,v) \vdash^* (q_2, v) \;\Leftrightarrow\; w = ub$ für ein $u \in \{a,b\}^*$

  $B_3(w)$: $(q_0, w) \vdash^* (q_3, \epsilon) \;\Leftrightarrow\; w = uc$ für ein $u \in \{a,b\}^*$

  Behauptung B_3 folgt unmittelbar aus B_0–B_2

Nachweis der erkannten Sprache (mit ϵ-Übergängen)



- **Mit welchen Wörtern werden die q_i erreicht?**
  q_0: Gleiche Wörter wie q_1 oder q_2 und ϵ, also $\{a,b\}^*$
  q_1: Letztes Symbol ist ein a, davor b's oder a's
  q_2: Letztes Symbol ist ein b, davor a's oder b's
  q_3: Letztes Symbol ist ein c, davor ein Wort aus $\{a,b\}^*$
- **Formuliere Aussage mit Konfigurationen**
  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \;\Leftrightarrow\; w \in \{a,b\}^*$
  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \;\Leftrightarrow\; w = ua$ für ein $u \in \{a,b\}^*$
  $B_2(w)$: $(q_0, w\,v) \vdash^* (q_2, v) \;\Leftrightarrow\; w = ub$ für ein $u \in \{a,b\}^*$
  $B_3(w)$: $(q_0, w) \vdash^* (q_3, \epsilon) \;\Leftrightarrow\; w = uc$ für ein $u \in \{a,b\}^*$
  Behauptung B_3 folgt unmittelbar aus B_0–B_2
- **Es folgt** $L(A) = \{w \mid (q_0, w) \vdash^* (q_3, \epsilon)\} = \{w \mid \exists u \in \{a,b\}^*. w = uc\}$

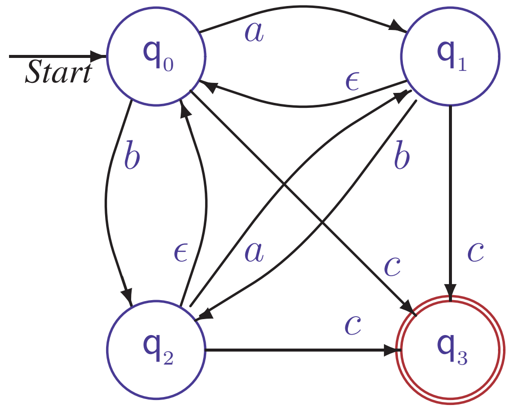
Induktiver Beweis für NEA's mit ϵ-Übergängen

- **Zeige simultan für alle Wörter $w, v \in \{a, b, c\}^*$:**

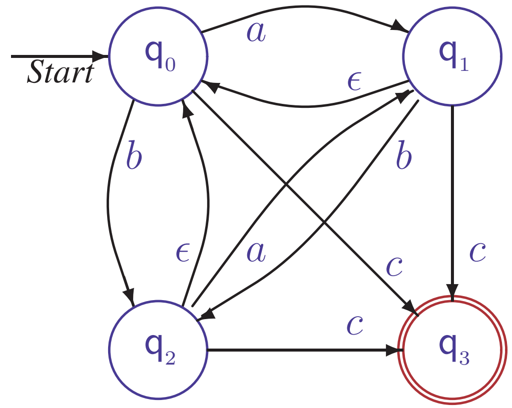
  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \;\Leftrightarrow\; w \in \{a, b\}^*$

  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \;\Leftrightarrow\; w = ua$ für ein $u \in \{a, b\}^*$

  $B_2(w)$: $(q_0, w\,v) \vdash^* (q_2, v) \;\Leftrightarrow\; w = ub$ für ein $u \in \{a, b\}^*$



Induktiver Beweis für NEA's mit ϵ-Übergängen



- **Zeige simultan für alle Wörter $w, v \in \{a, b, c\}^*$:**

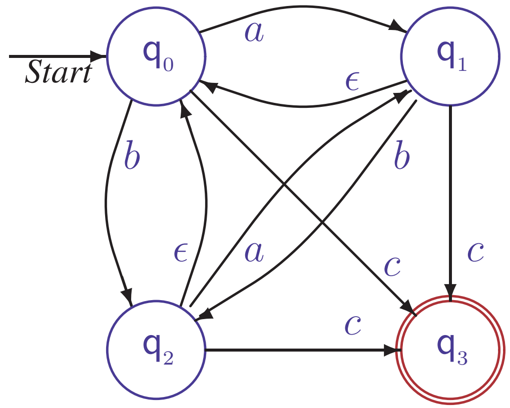
  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \Leftrightarrow w \in \{a, b\}^*$

  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \Leftrightarrow w = ua$ für ein $u \in \{a, b\}^*$

  $B_2(w)$: $(q_0, w\,v) \vdash^* (q_2, v) \Leftrightarrow w = ub$ für ein $u \in \{a, b\}^*$

- **Induktionsanfang $w = \epsilon$**
  - Per Definition gilt $(q_0, v) \vdash^* (q_0, v)$ und $w \in \{a, b\}^*$, und $w \neq ua$, $w \neq ub$ $\checkmark$

Induktiver Beweis für NEA's mit ϵ-Übergängen



- **Zeige simultan für alle Wörter $w, v \in \{a, b, c\}^*$:**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \;\Leftrightarrow\; w \in \{a, b\}^*$

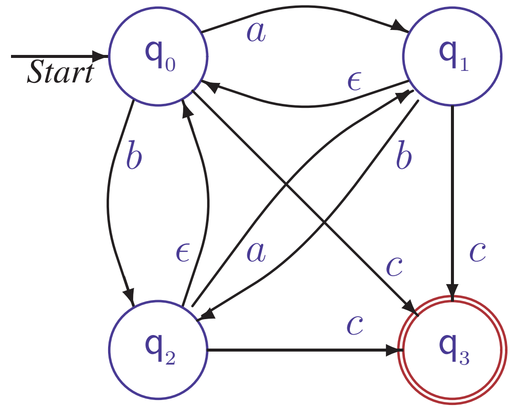
  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \;\Leftrightarrow\; w = ua$ für ein $u \in \{a, b\}^*$

  $B_2(w)$: $(q_0, w\,v) \vdash^* (q_2, v) \;\Leftrightarrow\; w = ub$ für ein $u \in \{a, b\}^*$

- **Induktionsanfang $w = \epsilon$**
  - Per Definition gilt $(q_0, v) \vdash^* (q_0, v)$ und $w \in \{a, b\}^*$, und $w \neq ua$, $w \neq ub$ $\checkmark$

- **Induktionsannahme: Sei $B_0(w')$–$B_2(w')$ gezeigt für $w' \in \{a, b, c\}^*$**

Induktiver Beweis für NEA's mit ϵ-Übergängen



- **Zeige simultan für alle Wörter $w, v \in \{a, b, c\}^*$:**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \Leftrightarrow w \in \{a, b\}^*$

  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \Leftrightarrow w = ua$ für ein $u \in \{a, b\}^*$

  $B_2(w)$: $(q_0, w\,v) \vdash^* (q_2, v) \Leftrightarrow w = ub$ für ein $u \in \{a, b\}^*$

- **Induktionsanfang $w = \epsilon$**
  - Per Definition gilt $(q_0, v) \vdash^* (q_0, v)$ und $w \in \{a, b\}^*$, und $w \neq ua$, $w \neq ub$ $\surd$

- **Induktionsannnahme: Sei $B_0(w')$–$B_2(w')$ gezeigt für $w' \in \{a, b, c\}^*$**

- **Induktionsschritt: Sei $w = w'x$ für ein $x \in \{a, b, c\}$**

  $B_1(w)$: $(q_0, w'x\,v) \vdash^* (q_1, v) \Leftrightarrow (q_0, w'x\,v) \vdash^* (q_{0/2}, xv) \vdash (q_1, v)$

  $\overset{(B_0(w'), B_2(w'))}{\Leftrightarrow} (w' \in \{a, b\}^* \vee \exists u \in \{a, b\}^*.w' = ub) \wedge x = a$ $\surd$

  $B_2(w)$: $(q_0, w'x\,v) \vdash^* (q_2, v) \Leftrightarrow (q_0, w'x\,v) \vdash^* (q_{0/1}, xv) \vdash (q_2, v)$

  $\overset{(B_0(w'), B_1(w'))}{\Leftrightarrow} (w' \in \{a, b\}^* \vee \exists u \in \{a, b\}^*.w' = ua) \wedge x = b$ $\surd$

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \Leftrightarrow (q_0, w\,v) \vdash^* (q_1, v) \vee (q_0, w\,v) \vdash^* (q_2, v)$

  $\overset{(B_1(w), B_2(w))}{\Leftrightarrow} \exists u \in \{a, b\}^*.w = ua \vee \exists u \in \{a, b\}^*.w = ub \Leftrightarrow w \in \{a, b\}^*$ $\surd$

Beziehung zu deterministischen Automaten

- **Nichtdeterministische Automaten sind flexibler**
  – Man muss sich nicht auf eine genaue Verarbeitungsfolge festlegen
  – Man kann optionale Eingaben elegant verarbeiten

Beziehung zu deterministischen Automaten

- **Nichtdeterministische Automaten sind flexibler**
  - Man muss sich nicht auf eine genaue Verarbeitungsfolge festlegen
  - Man kann optionale Eingaben elegant verarbeiten
- **DEAs sind genauso ausdrucksstark wie ϵ-NEAs**
  - Man kann Mengen von ϵ-NEA-Zuständen als DEA Zustände codieren
  - Man kann mengenwertige Zustandsüberführungsfunktionen codieren

Beziehung zu deterministischen Automaten

- **Nichtdeterministische Automaten sind flexibler**
  - Man muss sich nicht auf eine genaue Verarbeitungsfolge festlegen
  - Man kann optionale Eingaben elegant verarbeiten
- **DEAs sind genauso ausdrucksstark wie ϵ-NEAs**
  - Man kann Mengen von ϵ-NEA-Zuständen als DEA Zustände codieren
  - Man kann mengenwertige Zustandsüberführungsfunktionen codieren
- **(Potenzmengen- oder) Teilmengenkonstruktion**
  - Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
  - Konstruiere äquivalenten DEA $A_D = (Q_D, \Sigma, \delta_D, q_D, F_D)$ mit

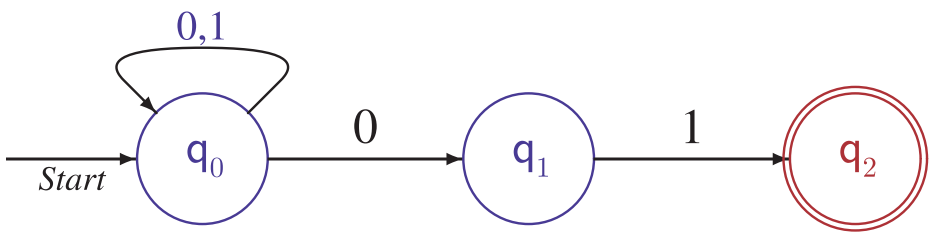
    $Q_D = \mathcal{P}(Q_N)$: (Jeder Zustand entspricht einer Menge von Zuständen)

    $q_D = \epsilon\text{-Hülle}(q_0)$ (Die anfangs erreichbaren Zustände von A_N, $\{q_0\}$ bei NEAs)

    $\delta_D(S,a) = \{p \mid \exists q \in S.\, p \in \hat{\delta}_N(q,a)\}$ (Die von allen $q \in S$ erreichbaren Zustände)

    $F_D = \{S \in Q_D \mid S \cap F_N \neq \emptyset\}$ ($S \in F_D \mathrel{\hat{=}}$ ein Endzustand von A_N ist erreichbar)
  - Naive Konstruktion benötigt **$2^{|Q_N|}$ Zustände**
  - Optimierte Teilmengenkonstruktion erzeugt nur “notwendige” Zustände

Optimierte Teilmengenkonstruktion für NEAs

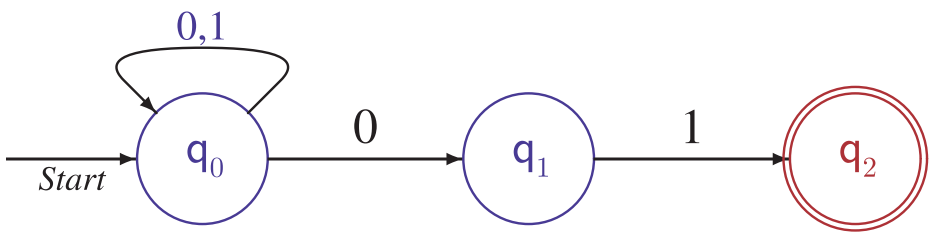


- **Konstruktion von Zustandsmengen und (reduzierter) Überführungsfunktion**

	0	1
$\rightarrow \{q_0\}$		

$$Q_0 := \{\{q_0\}\}$$

Optimierte Teilmengenkonstruktion für NEAs

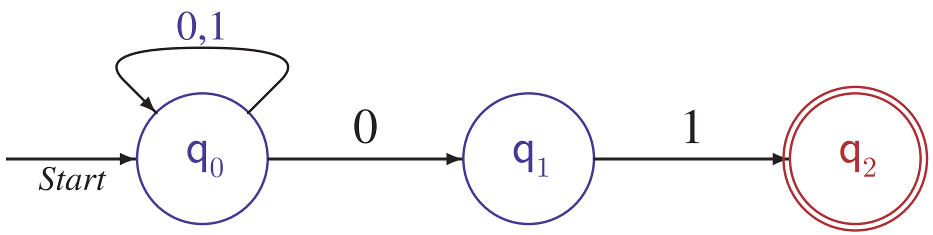


- **Konstruktion von Zustandsmengen und (reduzierter) Überführungsfunktion**

	0	1
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$

$$Q_0 := \{\{q_0\}\}$$

Optimierte Teilmengenkonstruktion für NEAs

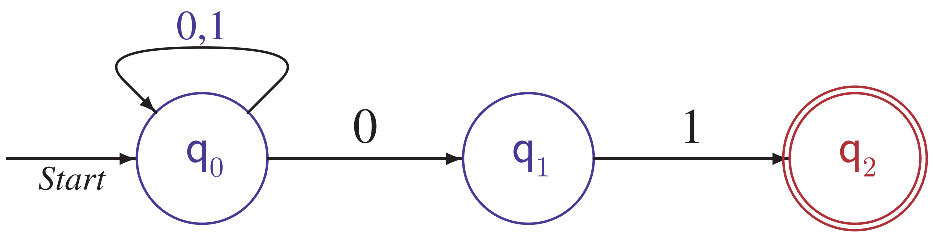


- **Konstruktion von Zustandsmengen und (reduzierter) Überführungsfunktion**

	0	1
$\rightarrow \{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\{q_0,q_1\}$		

$$Q_0 := \{\{q_0\}\}$$
$$Q_1 := \{\{q_0\}, \{q_0, q_1\}\}$$

Optimierte Teilmengenkonstruktion für NEAs

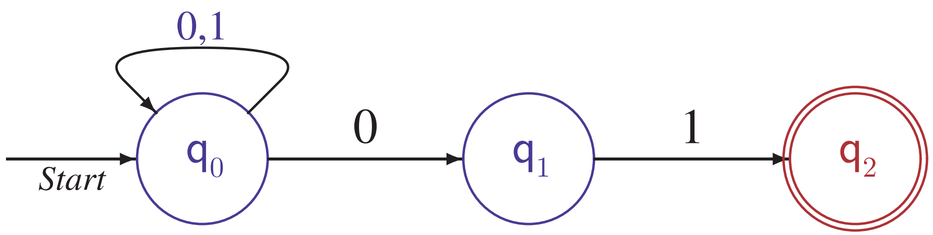


- **Konstruktion von Zustandsmengen und (reduzierter) Überführungsfunktion**

	0	1
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

$$Q_0 := \{\{q_0\}\}$$
$$Q_1 := \{\{q_0\}, \{q_0, q_1\}\}$$

Optimierte Teilmengenkonstruktion für NEAs

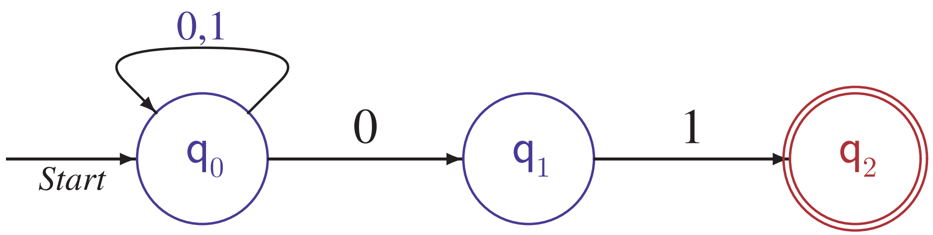


- **Konstruktion von Zustandsmengen und (reduzierter) Überführungsfunktion**

	0	1
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$* \{q_0, q_2\}$		

$$Q_0 := \{\{q_0\}\}$$
$$Q_1 := \{\{q_0\}, \{q_0, q_1\}\}$$
$$Q_2 := \{\{q_0\}, \{q_0, q_1\}, \{q_0, q_2\}\}$$

Optimierte Teilmengenkonstruktion für NEAs

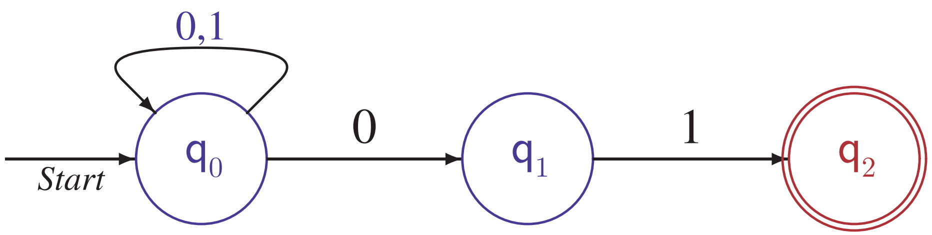


- **Konstruktion von Zustandsmengen und (reduzierter) Überführungsfunktion**

	0	1
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$* \{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$

$$Q_0 := \{\{q_0\}\}$$
$$Q_1 := \{\{q_0\}, \{q_0, q_1\}\}$$
$$Q_2 := \{\{q_0\}, \{q_0, q_1\}, \{q_0, q_2\}\}$$

Optimierte Teilmengenkonstruktion für NEAs



- **Konstruktion von Zustandsmengen und (reduzierter) Überführungsfunktion**

	0	1
$\rightarrow \{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\{q_0,q_1\}$	$\{q_0,q_1\}$	$\{q_0,q_2\}$
$*\ \{q_0,q_2\}$	$\{q_0,q_1\}$	$\{q_0\}$

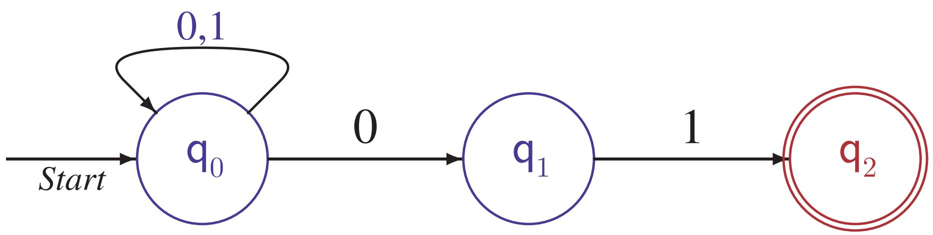
$Q_0 := \{\{q_0\}\}$

$Q_1 := \{\{q_0\}, \{q_0, q_1\}\}$

$Q_2 := \{\{q_0\}, \{q_0, q_1\}, \{q_0, q_2\}\}$

$Q_3 = Q_2$, also $Q_D = Q_2$,

Optimierte Teilmengenkonstruktion für NEAs



- **Konstruktion von Zustandsmengen und (reduzierter) Überführungsfunktion**

	0	1
→ {q0}	{q0,q1}	{q0}
{q0,q1}	{q0,q1}	{q0,q2}
* {q0,q2}	{q0,q1}	{q0}

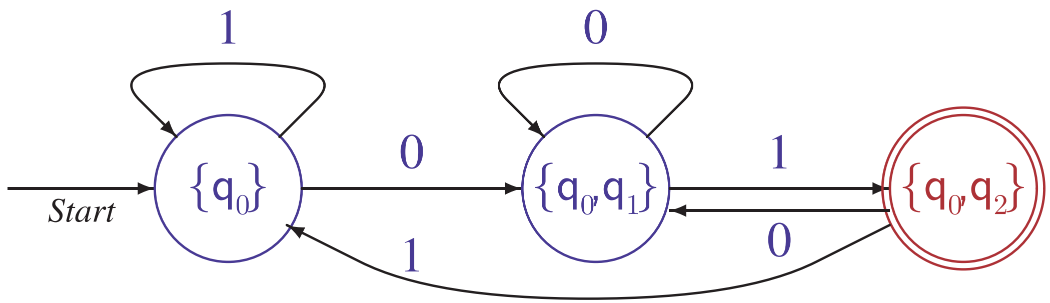
$Q_0 := \{\{q_0\}\}$

$Q_1 := \{\{q_0\}, \{q_0, q_1\}\}$

$Q_2 := \{\{q_0\}, \{q_0, q_1\}, \{q_0, q_2\}\}$

$Q_3 = Q_2$, also $Q_D = Q_2$,

- **Resultierender deterministischer Automat**



Optimierte Teilmengenkonstruktion HMU Satz 2.11/2.12

- **Optimierung:** $Q_D \hat{=}$ **erreichbare Zustände**
  - Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
  - Für $S \subseteq Q_N$ definiere $\boldsymbol{\delta_D}(S,a) = \bigcup_{q\in S} \hat{\delta}_N(q,a) = \{p \mid \exists q \in S.\, p \in \hat{\delta}_N(q,a)\}$

    (erfaßt ϵ-Hülle)

Optimierte Teilmengenkonstruktion HMU Satz 2.11/2.12

- **Optimierung: $Q_D \hat{=}$ erreichbare Zustände**
  - Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
  - Für $S \subseteq Q_N$ definiere $\boldsymbol{\delta_D}(S,a) = \bigcup_{q\in S} \hat{\delta_N}(q,a) = \{p \mid \exists q \in S.\, p \in \hat{\delta_N}(q,a)\}$

    (erfaßt ϵ-Hülle)
  - Konstruiere Zustandsmenge Q_D iterativ gleichzeitig mit δ_D

    Start: $Q_0 := \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\}$

    Schritt: $Q_{i+1} := Q_i \cup \{\delta_D(S,a) \mid S \in Q_i, a \in \Sigma\}$

    Abschluss: Wenn $Q_{i+1} = Q_i$, dann halte an und setze $\boldsymbol{Q_D} := Q_i$

Optimierte Teilmengenkonstruktion

HMU Satz 2.11/2.12

- **Optimierung: $Q_D \hat{=}$ erreichbare Zustände**
  - Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
  - Für $S \subseteq Q_N$ definiere $\boldsymbol{\delta_D}(S,a) = \bigcup_{q\in S} \hat{\delta}_N(q,a) = \{p \mid \exists q \in S.\, p \in \hat{\delta}_N(q,a)\}$

    (erfaßt ϵ-Hülle)
  - Konstruiere Zustandsmenge Q_D iterativ gleichzeitig mit δ_D

    Start: $Q_0 := \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\}$

    Schritt: $Q_{i+1} := Q_i \cup \{\delta_D(S,a) \mid S \in Q_i, a \in \Sigma\}$

    Abschluss: Wenn $Q_{i+1} = Q_i$, dann halte an und setze $\boldsymbol{Q_D} := Q_i$
  - Setze $\boldsymbol{q_D} = \epsilon\text{-Hülle}(q_0)$ und $\boldsymbol{F_D} = \{S \in Q_D \mid S \cap F_N \neq \emptyset\}$

Optimierte Teilmengenkonstruktion HMU Satz 2.11/2.12

- **Optimierung: $Q_D \hat{=}$ erreichbare Zustände**
  - Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
  - Für $S \subseteq Q_N$ definiere $\boldsymbol{\delta_D}(S,a) = \bigcup_{q\in S} \hat{\delta}_N(q,a) = \{p \mid \exists q \in S.\, p \in \hat{\delta}_N(q,a)\}$

    (erfaßt ϵ-Hülle)
  - Konstruiere Zustandsmenge Q_D iterativ gleichzeitig mit δ_D

    Start: $Q_0 := \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\}$

    Schritt: $Q_{i+1} := Q_i \cup \{\delta_D(S,a) \mid S \in Q_i, a \in \Sigma\}$

    Abschluss: Wenn $Q_{i+1} = Q_i$, dann halte an und setze $\boldsymbol{Q_D} := Q_i$
  - Setze $\boldsymbol{q_D} = \epsilon\text{-Hülle}(q_0)$ und $\boldsymbol{F_D} = \{S \in Q_D \mid S \cap F_N \neq \emptyset\}$

- **ϵ-NEAs und DEAs akzeptieren dieselben Sprachen**
  - Jeder DEA ist als “eindeutiger” ϵ-NEA beschreibbar

Optimierte Teilmengenkonstruktion: Korrektheit

Für den konstruierten DEA gilt $L(A_D) = L(A_N)$

Zeige: $\hat{\delta}_D(q_D, w) = \hat{\delta}_N(q_0, w)$ **für alle $w \in \Sigma^*$**

Beweis durch strukturelle Induktion über den Aufbau der Wörter aus Σ^*

– Basisfall: Sei $w = \epsilon$:

$\hat{\delta}_D(q_D, \epsilon) = q_D = \epsilon\text{-Hülle}(q_0) = \hat{\delta}_N(q_0, \epsilon)$

– Induktionsannahme: Es gelte $\hat{\delta}_D(q_D, w') = \hat{\delta}_N(q_0, w')$ für ein $w' \in \Sigma^*$

– Induktionsschritt: Sei $w = w'a$ für ein $a \in \Sigma$:

Dann gilt $\hat{\delta}_D(q_D, w)$

$= \delta_D(\hat{\delta}_D(q_D, w'), a)$ (Definition $\hat{\delta}$ deterministisch)

$= \delta_D(\hat{\delta}_N(q_0, w'), a)$ (Induktionsannahme)

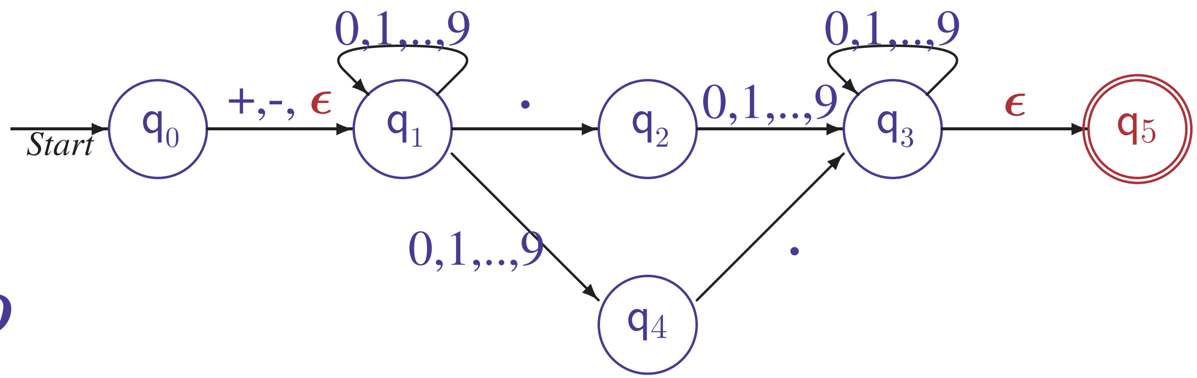
$= \bigcup_{q' \in \hat{\delta}_N(q_0,w')} \hat{\delta}_N(q', a)$ ($\delta_D(S, a) = \bigcup_{q \in S} \hat{\delta}_N(q, a)$)

$= \bigcup_{q' \in \hat{\delta}_N(q_0,w')} \bigcup_{q'' \in \delta_N(q',a)} \epsilon\text{-Hülle}(q'')$ (Definition $\hat{\delta}_N(q', a)$)

$= \hat{\delta}_N(q_0, w)$ (Definition $\hat{\delta}_N(q', w'\, a)$)

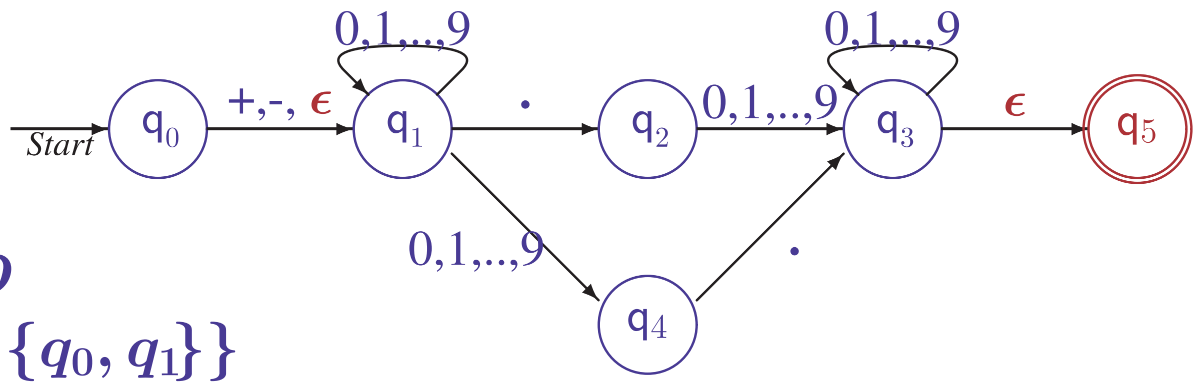
Es folgt $L(A_D) = \{w \mid \hat{\delta}_D(q_D, w) \in F_D\} = \{w \mid \hat{\delta}_N(q_0, w) \cap F_N \neq \emptyset\} = L(A_N)$

Optimierte Teilmengenkonstruktion für ϵ-NEAs



- **Konstruiere Q_D und δ_D**

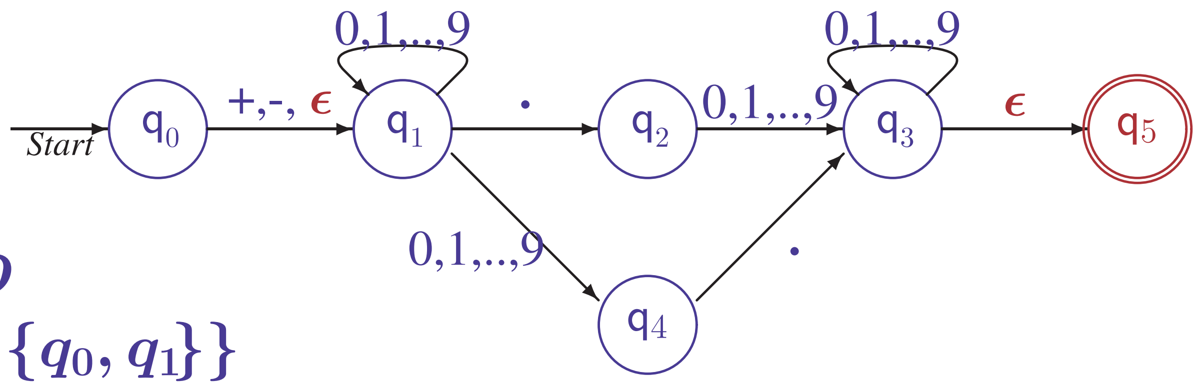
Optimierte Teilmengenkonstruktion für ϵ-NEAs



- **Konstruiere Q_D und δ_D**

  $Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

Optimierte Teilmengenkonstruktion für ϵ-NEAs

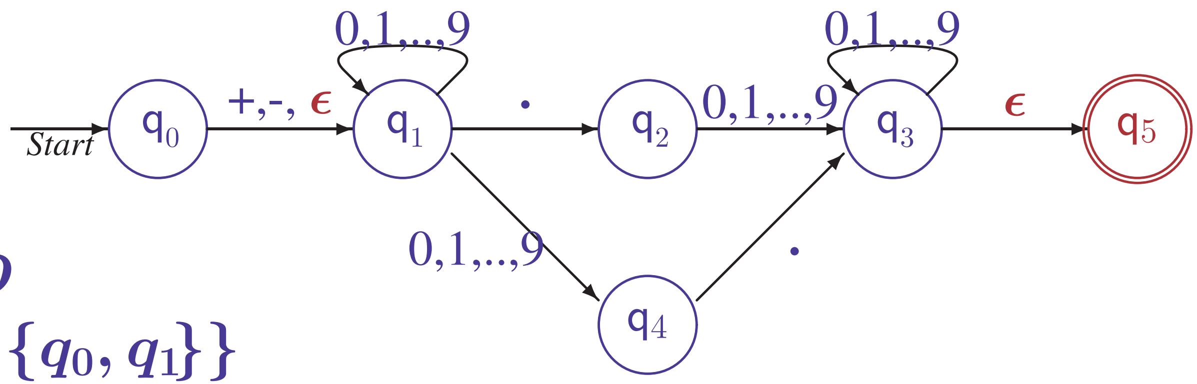


- **Konstruiere Q_D und δ_D**

  $Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

  – $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\delta_D(\{q_0, q_1\}, -) = \{q_1\}$

Optimierte Teilmengenkonstruktion für ϵ-NEAs



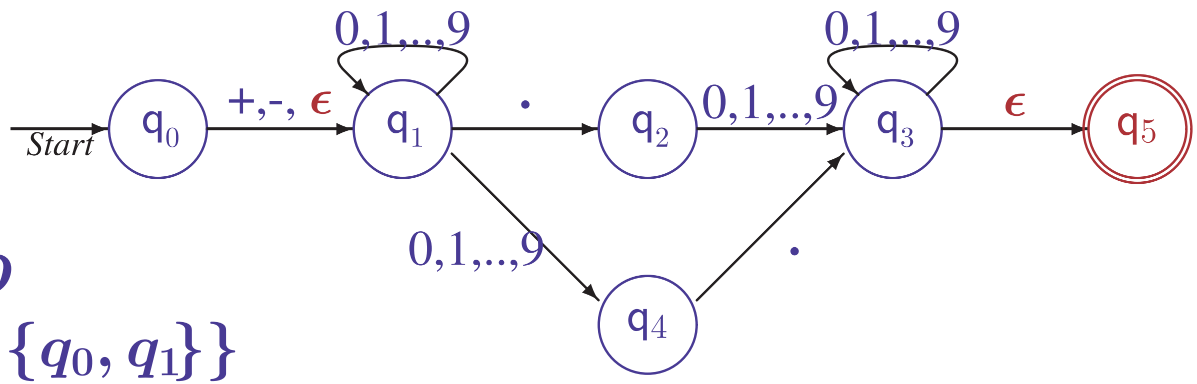
- **Konstruiere Q_D und δ_D**

  $Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

  – $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\delta_D(\{q_0, q_1\}, -) = \{q_1\}$

  – $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \ldots \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$,

Optimierte Teilmengenkonstruktion für ϵ-NEAs



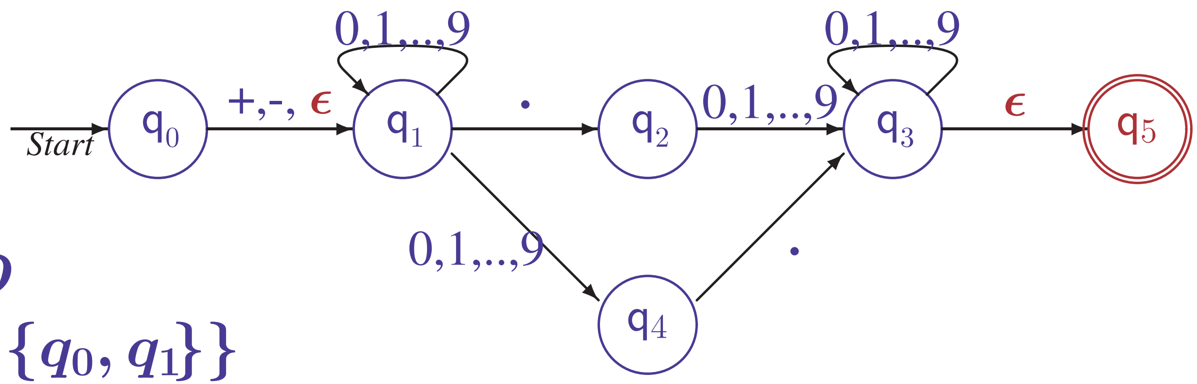
- **Konstruiere Q_D und δ_D**

  $Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

  – $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\quad\delta_D(\{q_0, q_1\}, -) = \{q_1\}$

  – $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \ldots \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$, $\quad\delta_D(\{q_0, q_1\}, .) = \{q_2\}$

Optimierte Teilmengenkonstruktion für ϵ-NEAs



- **Konstruiere Q_D und δ_D**

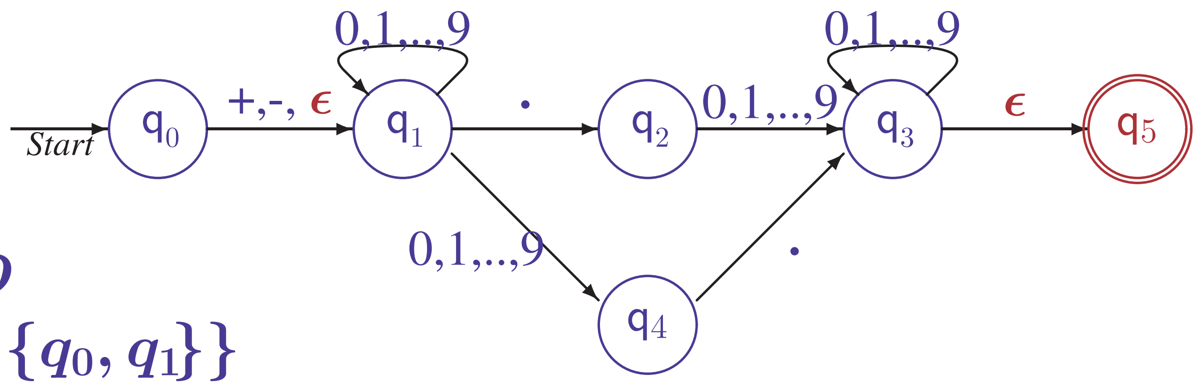
  $Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

  – $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\quad\delta_D(\{q_0, q_1\}, -) = \{q_1\}$

  – $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \ldots \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$, $\quad\delta_D(\{q_0, q_1\}, .) = \{q_2\}$

  $Q_1 = \{\ \{q_0, q_1\}\ \{q_1\}, \{q_1, q_4\}, \{q_2\}\ \}$

Optimierte Teilmengenkonstruktion für ϵ-NEAs



- **Konstruiere Q_D und δ_D**

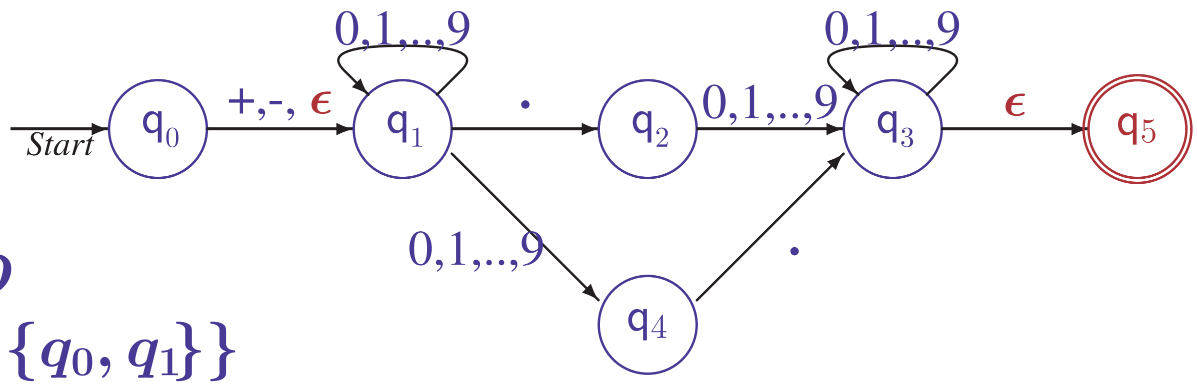
  $Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

  – $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\quad\delta_D(\{q_0, q_1\}, -) = \{q_1\}$

  – $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \ldots \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$, $\quad\delta_D(\{q_0, q_1\}, .) = \{q_2\}$

  $Q_1 = \{\ \{q_0, q_1\}\ \{q_1\}, \{q_1, q_4\}, \{q_2\}\ \}$

Optimierte Teilmengenkonstruktion für ϵ-NEAs



- **Konstruiere Q_D und δ_D**

  $Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

  – $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\quad\delta_D(\{q_0, q_1\}, -) = \{q_1\}$

  – $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \ldots \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$, $\quad\delta_D(\{q_0, q_1\}, .) = \{q_2\}$

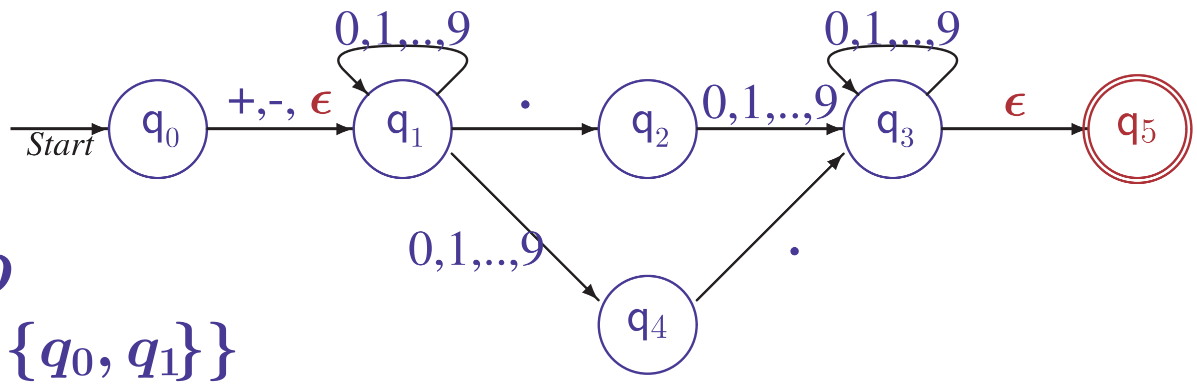
  $Q_1 = \{\ \{q_0, q_1\}\ \{q_1\}, \{q_1, q_4\}, \{q_2\}\ \}$

  – $\delta_D(\{q_1\}, +) = \delta_D(\{q_2\}, +) = \delta_D(\{q_1, q_4\}, +) = \emptyset, \ldots$

  – $\delta_D(\{q_1\}, 0) = \delta_D(\{q_1, q_4\}, 0) = \{q_1, q_4\}$ $\quad\delta_D(\{q_2\}, 0) = \{q_3, q_5\}, \ldots$

  – $\delta_D(\{q_1\}, .) = \{q_2\}$, $\quad\delta_D(\{q_2\}, .) = \emptyset$ $\quad\delta_D(\{q_1, q_4\}, .) = \{q_2, q_3, q_5\}$

Optimierte Teilmengenkonstruktion für ϵ-NEAs



- **Konstruiere Q_D und δ_D**

  $Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

  – $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\quad\delta_D(\{q_0, q_1\}, -) = \{q_1\}$

  – $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \ldots \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$, $\quad\delta_D(\{q_0, q_1\}, .) = \{q_2\}$

  $Q_1 = \{\ \{q_0, q_1\}\ \{q_1\}, \{q_1, q_4\}, \{q_2\}\ \}$

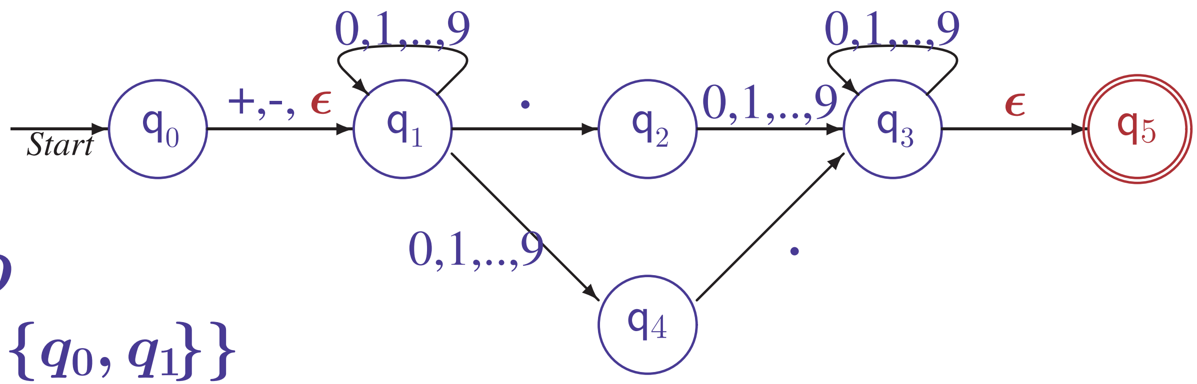
  – $\delta_D(\{q_1\}, +) = \delta_D(\{q_2\}, +) = \delta_D(\{q_1, q_4\}, +) = \emptyset, \ldots$

  – $\delta_D(\{q_1\}, 0) = \delta_D(\{q_1, q_4\}, 0) = \{q_1, q_4\}$ $\quad\delta_D(\{q_2\}, 0) = \{q_3, q_5\}, \ldots$

  – $\delta_D(\{q_1\}, .) = \{q_2\}$, $\quad\delta_D(\{q_2\}, .) = \emptyset$ $\quad\delta_D(\{q_1, q_4\}, .) = \{q_2, q_3, q_5\}$

  $Q_2 = \{\ \{q_0, q_1\}\ \{q_1\}, \{q_1, q_4\}, \{q_2\}, \emptyset, \{q_3, q_5\}, \{q_2, q_3, q_5\}\ \}$

Optimierte Teilmengenkonstruktion für ϵ-NEAs



- **Konstruiere Q_D und δ_D**

  $Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

  – $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\delta_D(\{q_0, q_1\}, -) = \{q_1\}$

  – $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \ldots \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$, $\delta_D(\{q_0, q_1\}, .) = \{q_2\}$

  $Q_1 = \{\ \{q_0, q_1\}\ \{q_1\}, \{q_1, q_4\}, \{q_2\}\ \}$

  – $\delta_D(\{q_1\}, +) = \delta_D(\{q_2\}, +) = \delta_D(\{q_1, q_4\}, +) = \emptyset, \ldots$

  – $\delta_D(\{q_1\}, 0) = \delta_D(\{q_1, q_4\}, 0) = \{q_1, q_4\}$ $\delta_D(\{q_2\}, 0) = \{q_3, q_5\}, \ldots$

  – $\delta_D(\{q_1\}, .) = \{q_2\}$, $\delta_D(\{q_2\}, .) = \emptyset$ $\delta_D(\{q_1, q_4\}, .) = \{q_2, q_3, q_5\}$

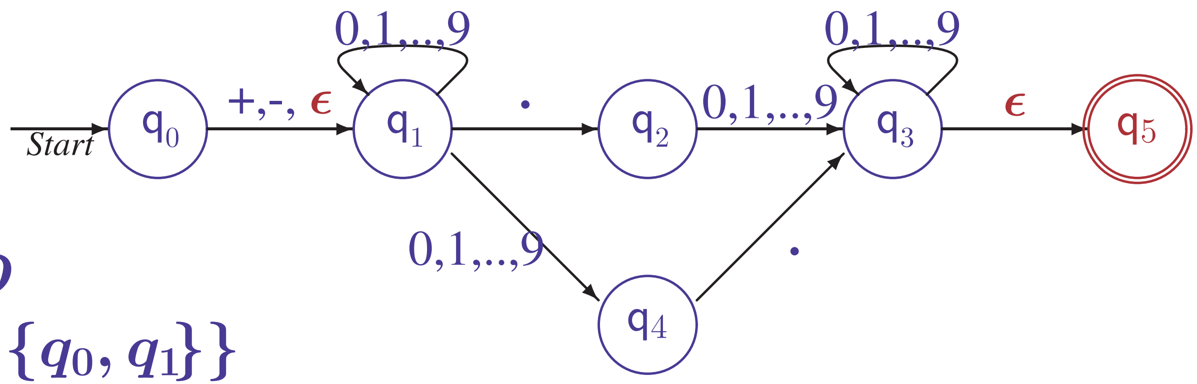
  $Q_2 = \{\ \{q_0, q_1\}\ \{q_1\}, \{q_1, q_4\}, \{q_2\}, \emptyset, \{q_3, q_5\}, \{q_2, q_3, q_5\}\ \}$

  – $\delta_D(\emptyset, +) = \delta_D(\{q_2, q_3, q_5\}, +) = \delta_D(\{q_3, q_5\}, +) = \emptyset, \ldots$

  – $\delta_D(\emptyset, 0) = \emptyset$, $\delta_D(\{q_2, q_3, q_5\}, 0) = \delta_D(\{q_3, q_5\}, 0) = \{q_3, q_5\}, \ldots$

  – $\delta_D(\emptyset, .) = \delta_D(\{q_2, q_3, q_5\}, .) = \delta_D(\{q_3, q_5\}, .) = \emptyset$

Optimierte Teilmengenkonstruktion für ϵ-NEAs



- **Konstruiere Q_D und δ_D**

$Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$

– $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\delta_D(\{q_0, q_1\}, -) = \{q_1\}$

– $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \ldots \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$, $\delta_D(\{q_0, q_1\}, .) = \{q_2\}$

$Q_1 = \{\ \{q_0, q_1\}\ \{q_1\}, \{q_1, q_4\}, \{q_2\}\ \}$

– $\delta_D(\{q_1\}, +) = \delta_D(\{q_2\}, +) = \delta_D(\{q_1, q_4\}, +) = \emptyset, \ldots$

– $\delta_D(\{q_1\}, 0) = \delta_D(\{q_1, q_4\}, 0) = \{q_1, q_4\}$ $\delta_D(\{q_2\}, 0) = \{q_3, q_5\}, \ldots$

– $\delta_D(\{q_1\}, .) = \{q_2\}$, $\delta_D(\{q_2\}, .) = \emptyset$ $\delta_D(\{q_1, q_4\}, .) = \{q_2, q_3, q_5\}$

$Q_2 = \{\ \{q_0, q_1\}\ \{q_1\}, \{q_1, q_4\}, \{q_2\}, \emptyset, \{q_3, q_5\}, \{q_2, q_3, q_5\}\ \}$

– $\delta_D(\emptyset, +) = \delta_D(\{q_2, q_3, q_5\}, +) = \delta_D(\{q_3, q_5\}, +) = \emptyset, \ldots$

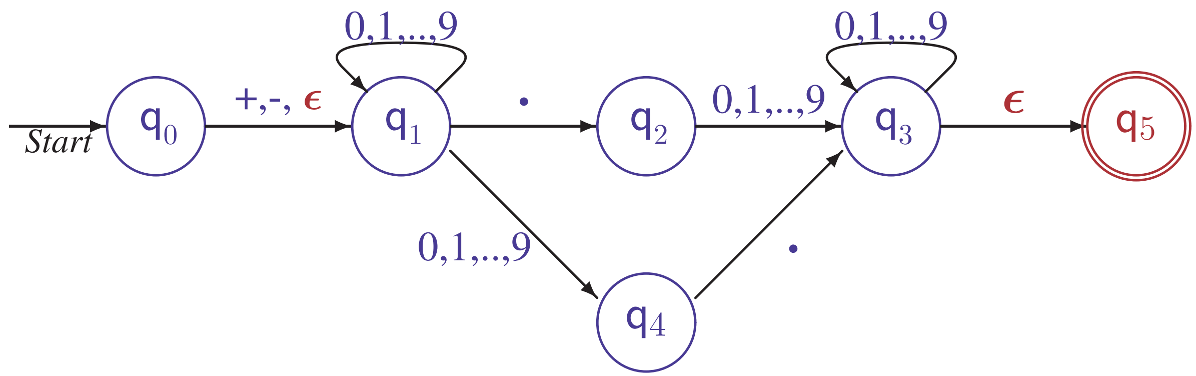
– $\delta_D(\emptyset, 0) = \emptyset$, $\delta_D(\{q_2, q_3, q_5\}, 0) = \delta_D(\{q_3, q_5\}, 0) = \{q_3, q_5\}, \ldots$

– $\delta_D(\emptyset, .) = \delta_D(\{q_2, q_3, q_5\}, .) = \delta_D(\{q_3, q_5\}, .) = \emptyset$

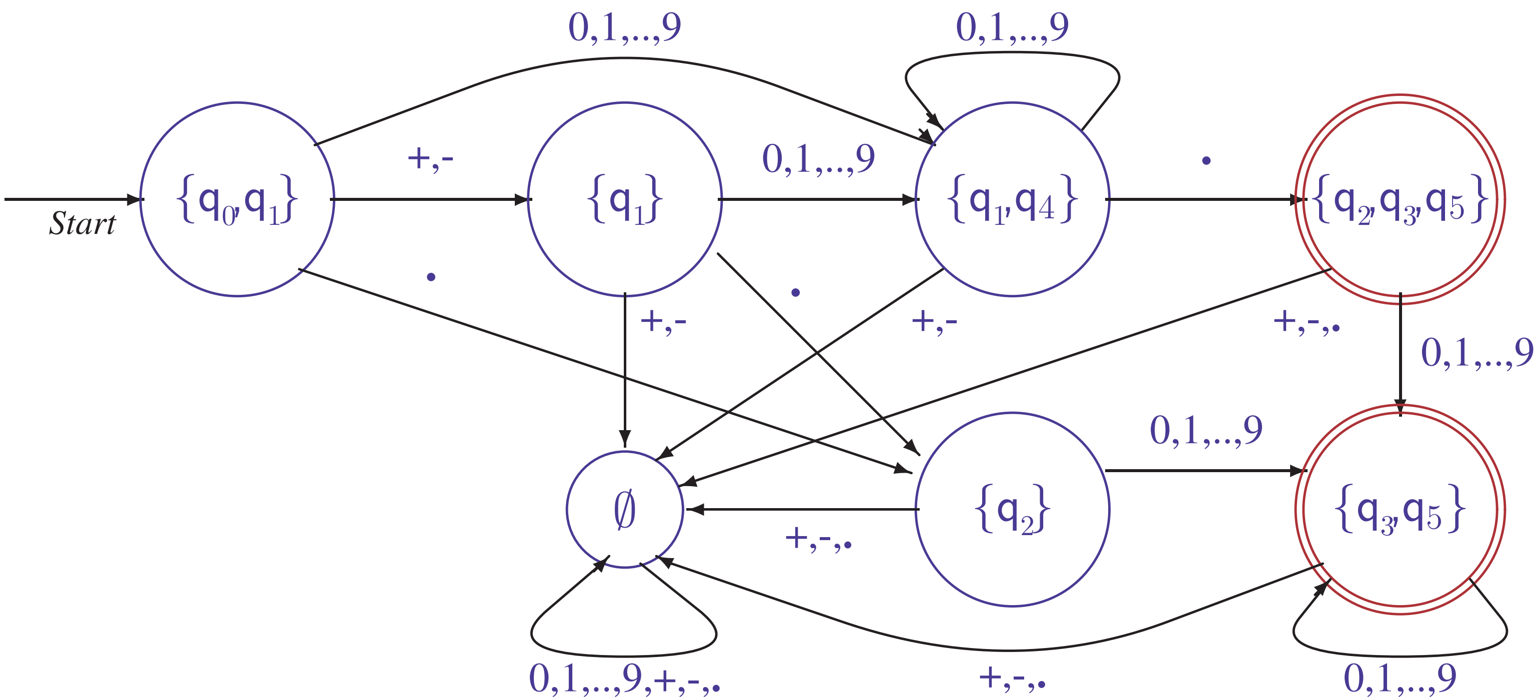
$Q_3 = \{\ \{q_0, q_1\}, \{q_1\}, \{q_1, q_4\}, \{q_2\}, \emptyset, \{q_3, q_5\}, \{q_2, q_3, q_5\}\ \} = Q_2 =: Q_D$

Erzeugter DEA für Dezimalzahlerkennung

Ursprünglicher ϵ-NEA

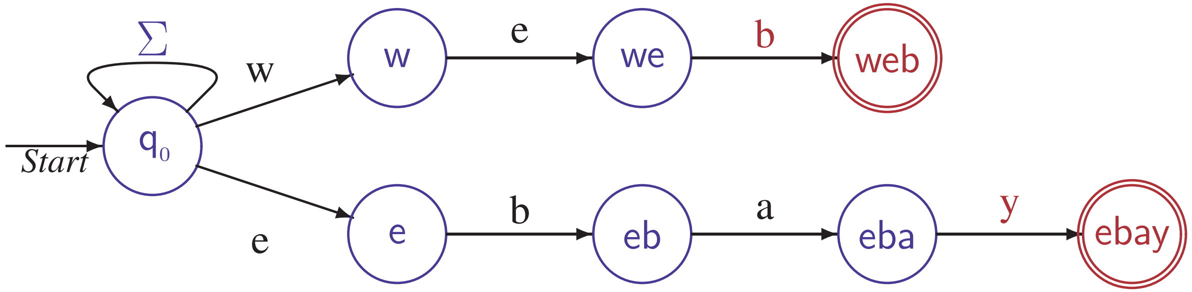


Generierter DEA



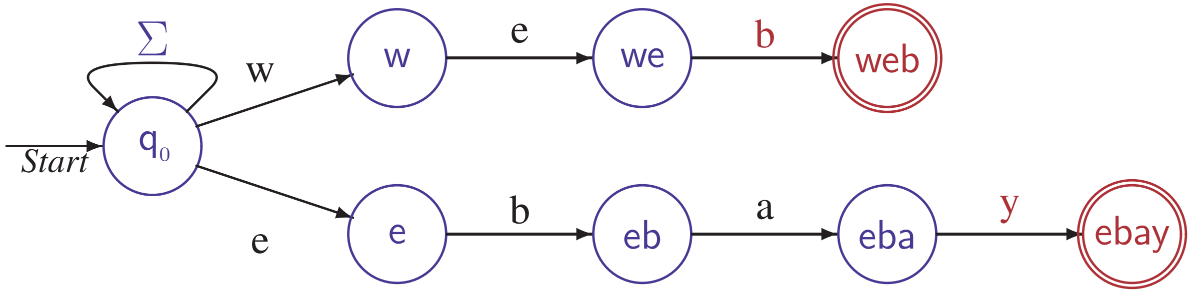
Deterministische Automaten für Textanalyse

Ursprünglicher ϵ-NEA

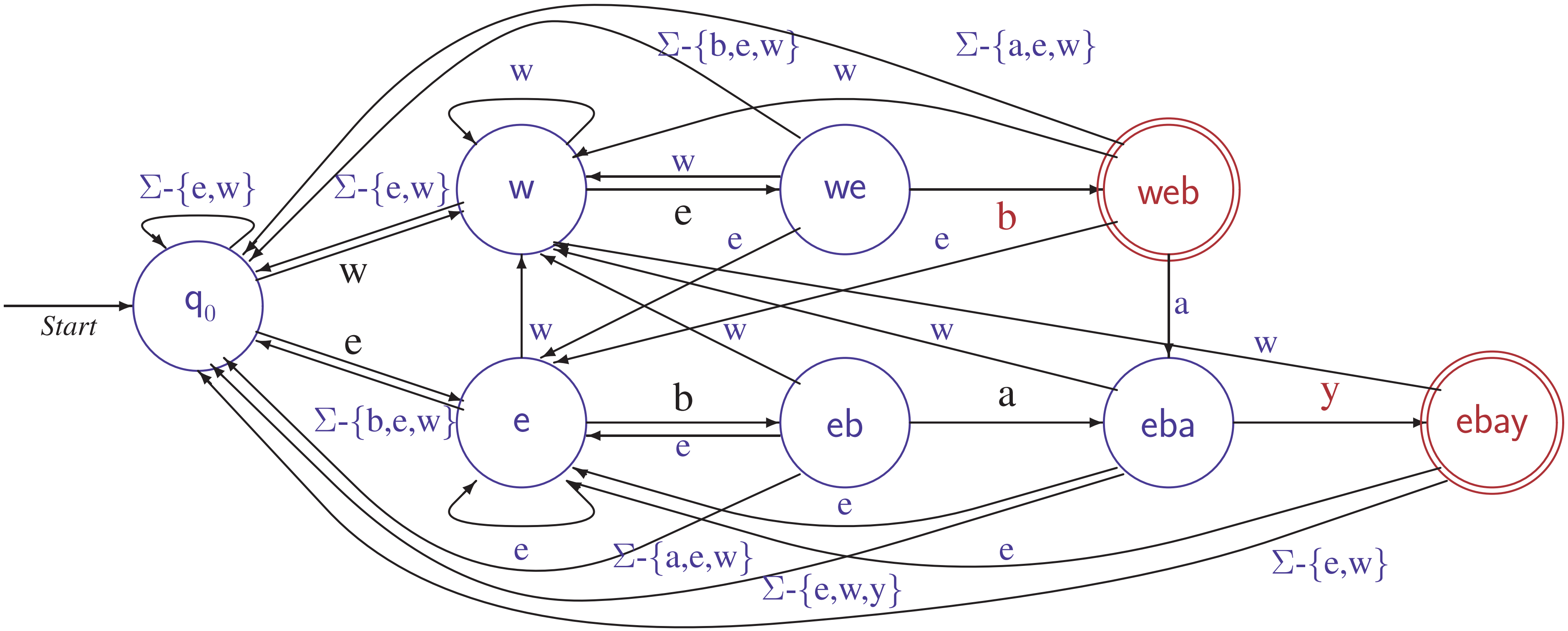


Deterministische Automaten für Textanalyse

Ursprünglicher ϵ-NEA



Generierter DEA



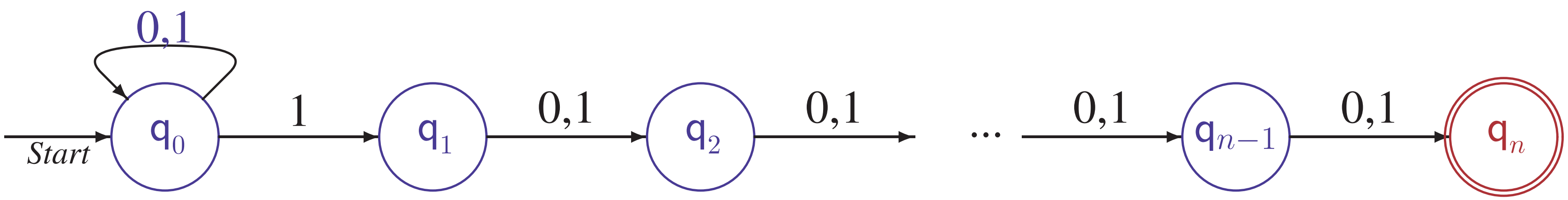
Teilmengenkonstruktion: Grösse des DEA

- **A_D kann so klein sein wie A_N**
  – Nur wenige Teilmengen von Q_N werden wirklich erreicht

Teilmengenkonstruktion: Grösse des DEA

- **A_D kann so klein sein wie A_N**
  - Nur wenige Teilmengen von Q_N werden wirklich erreicht

- **A_D kann exponentiell größer werden** HMU §2.3.6



  - $L(A_N) = \{w \in \{0,1\}^* \mid$ das n-te Zeichen vor dem Ende ist eine $1\}$
  - **Jeder DEA A für $L(A_N)$ benötigt mindestens 2^n Zustände**
  - **Beweis:** Es gibt 2^n Wörter der Länge n in $\{0,1\}^*$

    Hat A weniger als 2^n Zustände, so gibt es $w = a_1..a_n$ und $v = b_1..b_n$ mit $w \neq v$ und $\hat{\delta}_A(q_0, w) = \hat{\delta}_A(q_0, v)$ **(Schubfachprinzip)**

    Sei $a_i \neq b_i$. Für $q = \hat{\delta}_A(q_0, w0^{i-1}) = \hat{\delta}_A(q_0, v0^{i-1})$ folgt $q \in F$ und $q \notin F$

Endliche Automaten – Zusammenfassung

- **Deterministische Endliche Automaten (DEA)**
  - Endliche Menge von Zuständen, endliche Menge von Eingabesymbolen
  - Ein fester Startzustand, null oder mehr akzeptierende Zustände
  - Überführungsfunktion bestimmt Änderung des Zustands bei Abarbeitung der Eingabe
  - Erkannte Sprache: Eingaben, deren Abarbeitung in einem akzeptierenden Zustand endet
- **Automaten mit Ausgabe (Mealy/Moore-Automat)**
  - Wie DEA, mit zusätzlicher Ausgabefunktion
  - Gegenseitige Simulation möglich
- **Nichtdeterministische Automaten (ϵ-NEA / NEA)**
  - Wie DEA, aber mit mengenwertiger Überführungsfunktion und Zustandsüberführung bei leerer Eingabe
  - Durch Teilmengenkonstruktion in äquivalenten DEA transformierbar