

Theoretische Informatik I



Einheit 2.2

Nichtdeterministische Automaten



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

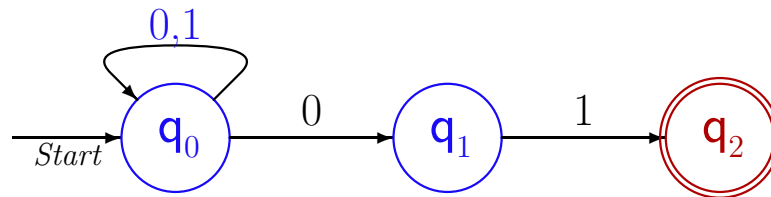
WAS IST NICHTDETERMISMUS?

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

WAS IST NICHTDETERMISMUS?

- **Verhalten nicht eindeutig bestimmt**

- Automat wählt Folgezustand aus mehreren Möglichkeiten

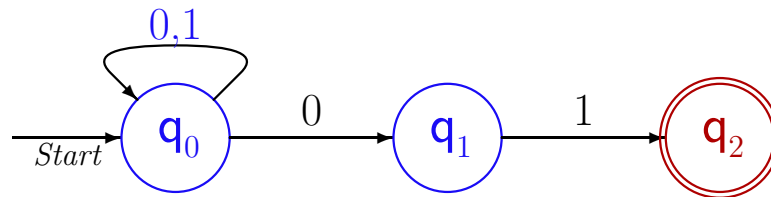


- Automat erkennt Strings, die mit **01** enden
- Eine 0 kann das erste Symbol des Endes **01** sein ... oder auch nicht

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

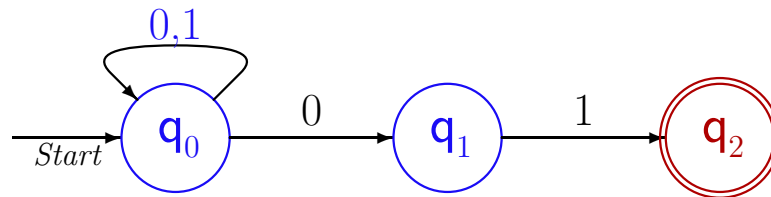
- **Spontane Übergänge zwischen Zuständen**

- Automat geht ohne Eingabe in anderen Zustand über (**ϵ -Übergang**)

WAS IST NICHTDETERMISMUS?

- **Verhalten nicht eindeutig bestimmt**

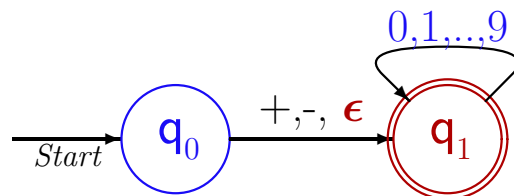
- Automat wählt Folgezustand aus mehreren Möglichkeiten



- 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**)

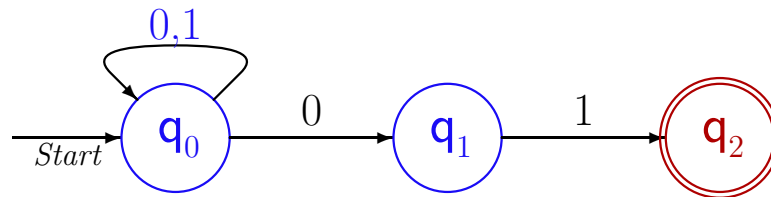


- Automat erkennt ganze Zahlen mit und ohne Vorzeichen

WAS IST NICHTDETERMISMUS?

- **Verhalten nicht eindeutig bestimmt**

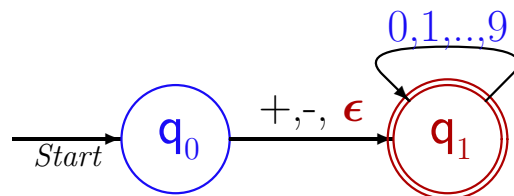
- Automat wählt Folgezustand aus mehreren Möglichkeiten



- 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)

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

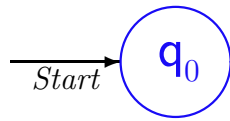
- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage
 - Deterministisches Erkennungsverfahren mühsam zu beschreiben

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**

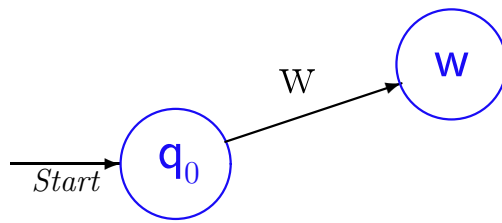
NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

- **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**



NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

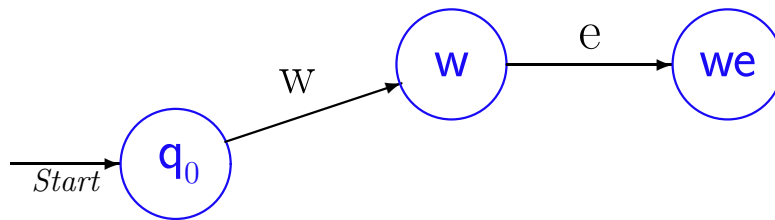
- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage
 - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
 - z.B. Suche nach den Wörtern **web**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



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

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

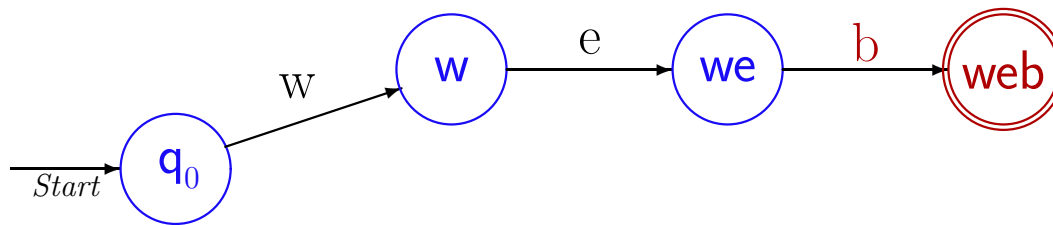
- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage
 - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
 - z.B. Suche nach den Wörtern **web**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



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

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

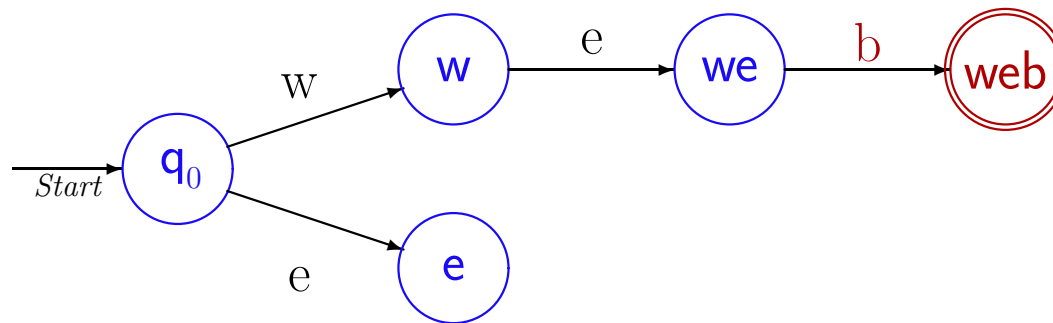
- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage
 - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
 - z.B. Suche nach den Wörtern **web**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



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

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

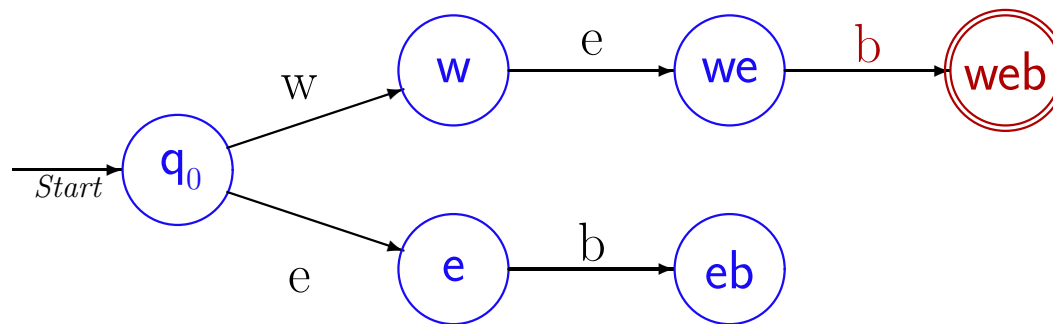
- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage
 - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
 - z.B. Suche nach den Wörtern **web**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



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

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

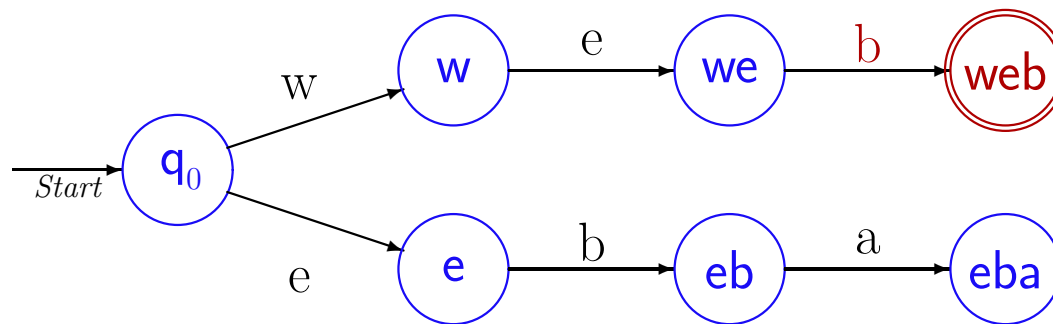
- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage
 - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
 - z.B. Suche nach den Wörtern **web**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



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

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

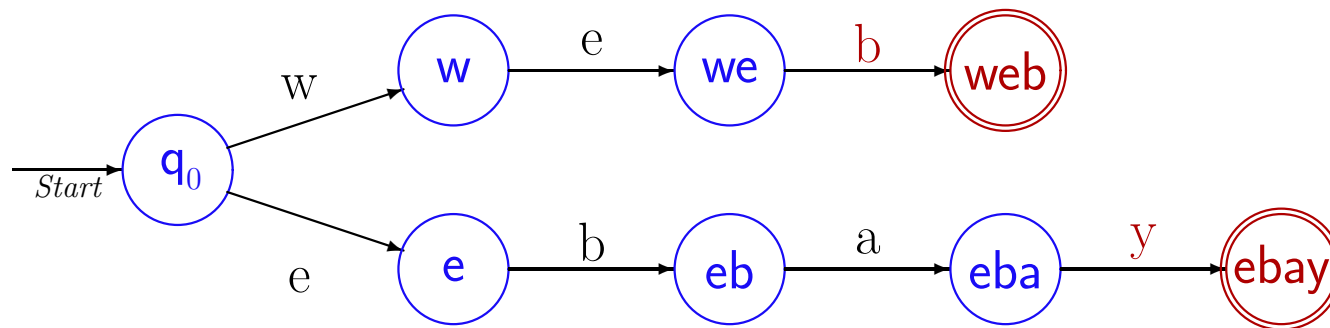
- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage
 - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
 - z.B. Suche nach den Wörtern **web**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



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

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

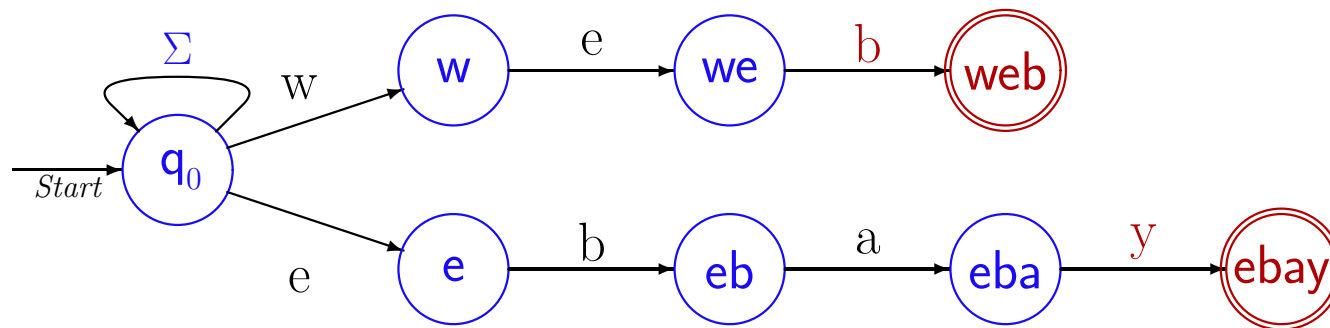
- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage
 - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
 - z.B. Suche nach den Wörtern **web**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



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

NICHTDETERMINISTISCHE AUTOMATEN – WOZU?

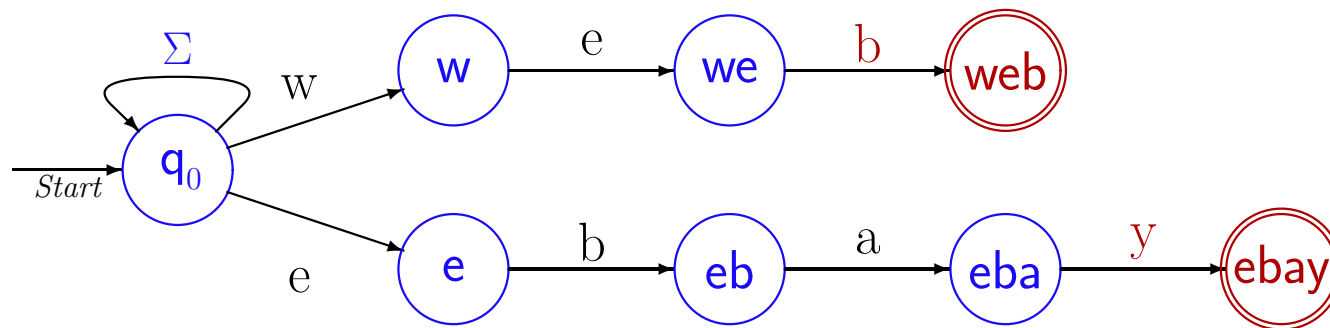
- **Elegante Form der Textsuche in Dokumenten**
 - Viele verschiedene Wörter in großen Textsammlungen (Internet)
 - Leichte Beschreibung der Suchanfrage
 - Deterministisches Erkennungsverfahren mühsam zu beschreiben
- **Idee: Simultane Verarbeitung von Alternativen**
 - z.B. Suche nach den Wörtern **web**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



- 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

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**



- 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

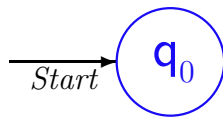
Nichtdeterminismus $\hat{=}$ verfolge alle Möglichkeiten simultan

- **Erkenne Dezimalzahlen im Programmcode**

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

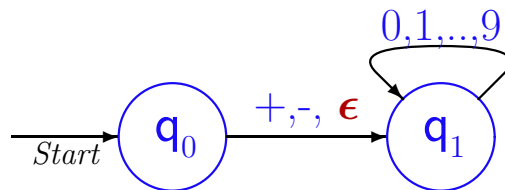
- **Erkenne Dezimalzahlen im Programmcode**

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



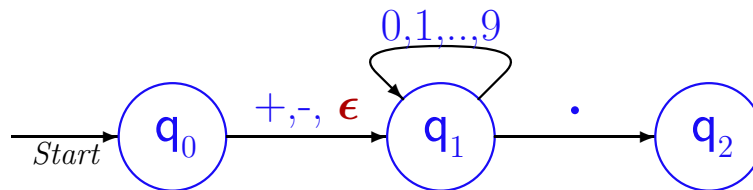
● **Erkenne Dezimalzahlen im Programmcode**

- Zwei Zeichenreihen von Ziffern getrennt durch Dezimalpunkt
- Eine der beiden Zeichenreihen darf leer sein, aber nicht beide
- Optionales Vorzeichen $+$ oder $-$



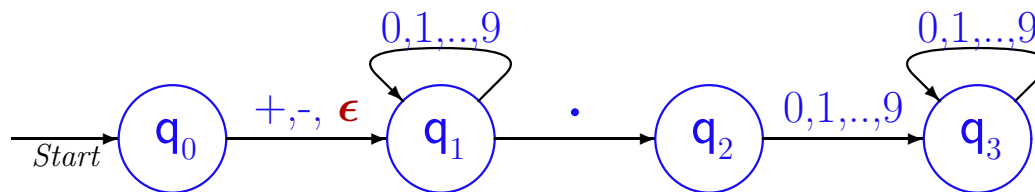
● **Erkenne Dezimalzahlen** im Programmcode

- Zwei Zeichenreihen von Ziffern getrennt durch Dezimalpunkt
- Eine der beiden Zeichenreihen darf leer sein, aber nicht beide
- Optionales Vorzeichen $+$ oder $-$



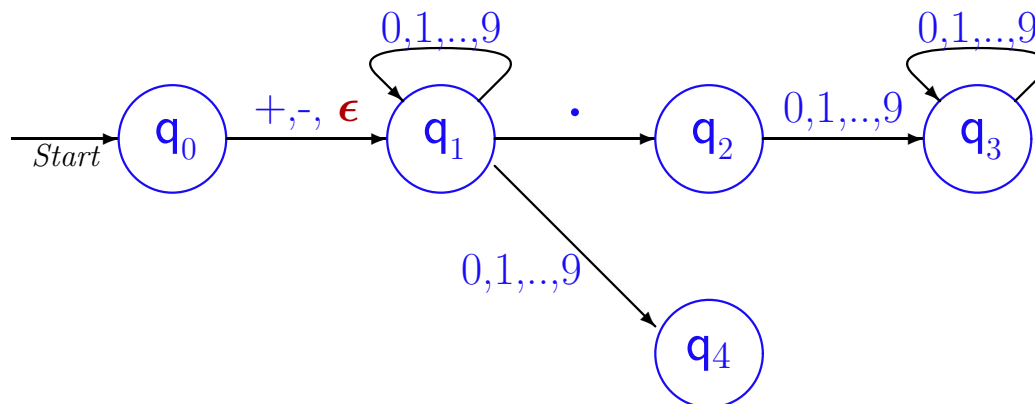
● **Erkenne Dezimalzahlen** im Programmcode

- Zwei Zeichenreihen von Ziffern getrennt durch Dezimalpunkt
- Eine der beiden Zeichenreihen darf leer sein, aber nicht beide
- Optionales Vorzeichen $+$ oder $-$



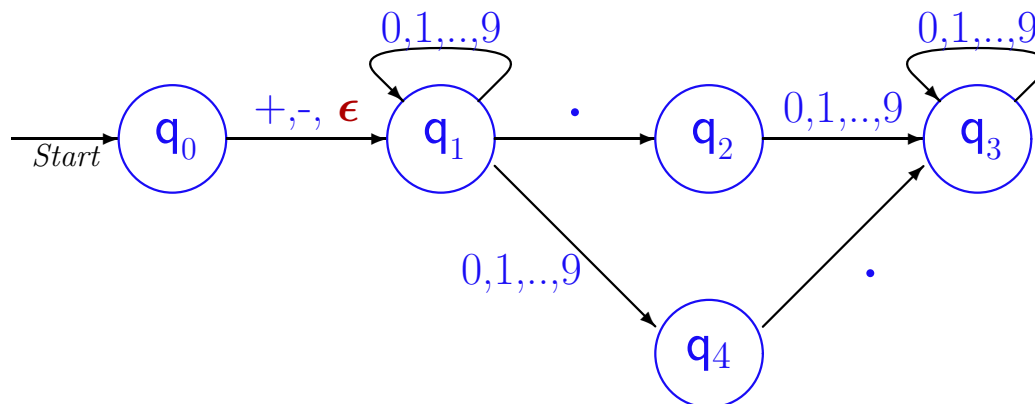
● **Erkenne Dezimalzahlen** im Programmcode

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



● **Erkenne Dezimalzahlen im Programmcode**

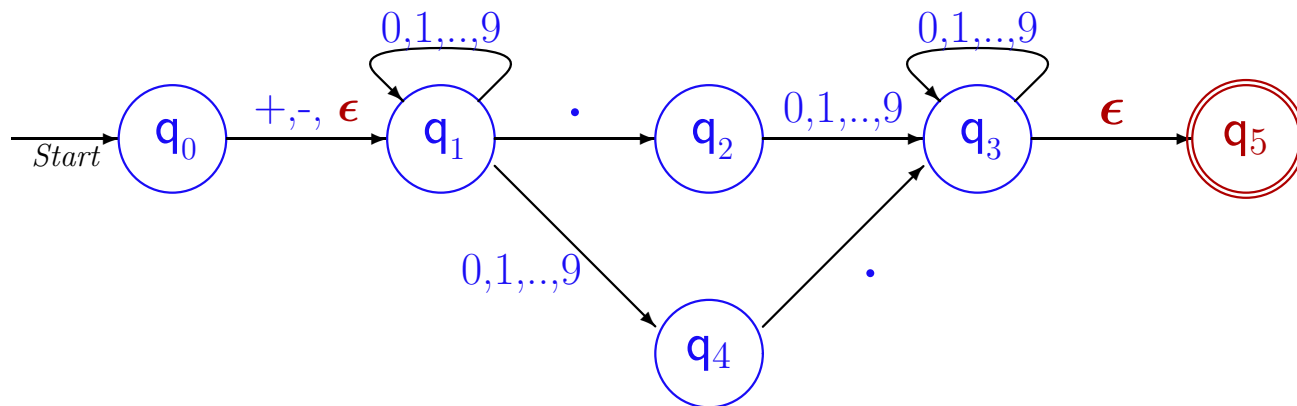
- Zwei Zeichenreihen von Ziffern getrennt durch Dezimalpunkt
- Eine der beiden Zeichenreihen darf leer sein, aber nicht beide
- Optionales Vorzeichen $+$ oder $-$



ϵ -ÜBERGÄNGE – VERARBEITUNG OPTIONALER EINGABEN

● **Erkenne Dezimalzahlen im Programmcode**

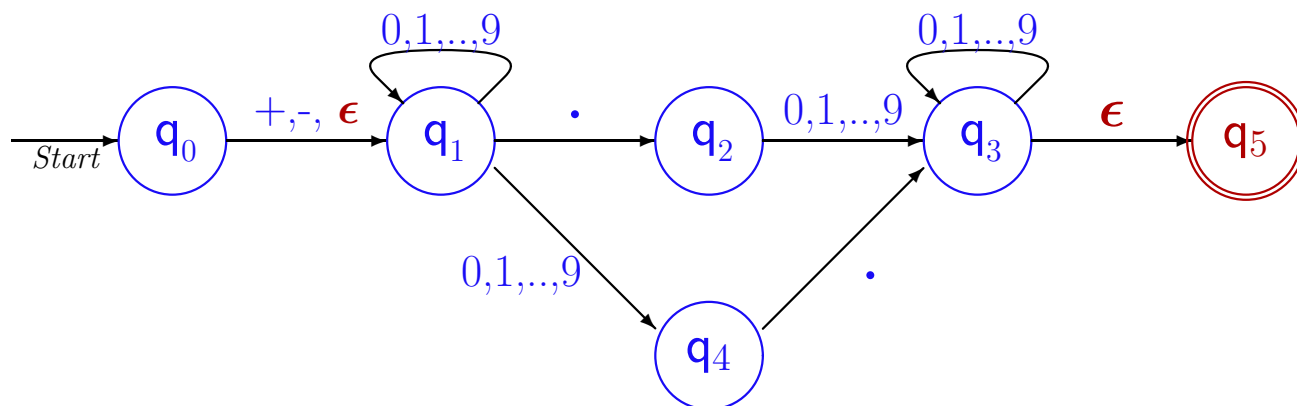
- Zwei Zeichenreihen von Ziffern getrennt durch Dezimalpunkt
- Eine der beiden Zeichenreihen darf leer sein, aber nicht beide
- Optionales Vorzeichen + oder -



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 $-$



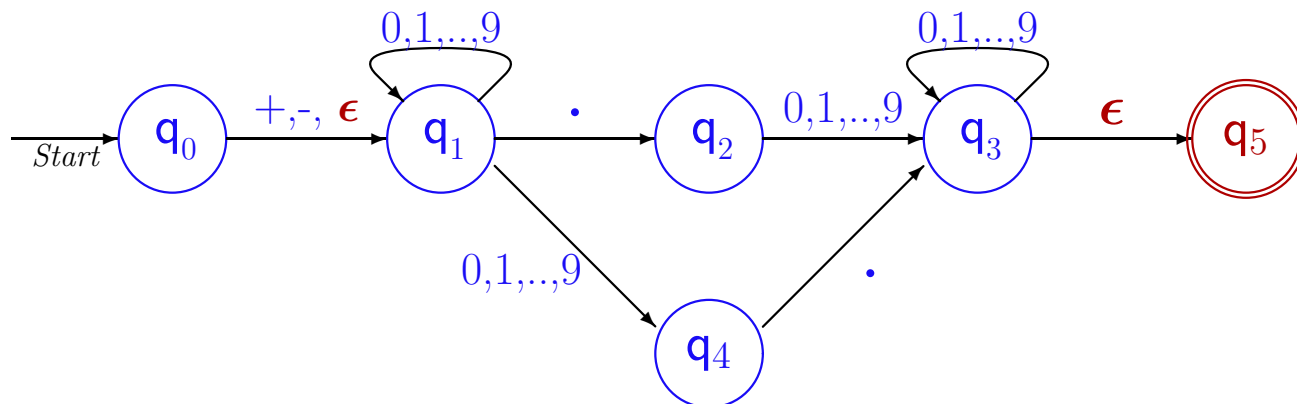
● **50c Kaffeeautomat**

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

ϵ -ÜBERGÄNGE – VERARBEITUNG OPTIONALER EINGABEN

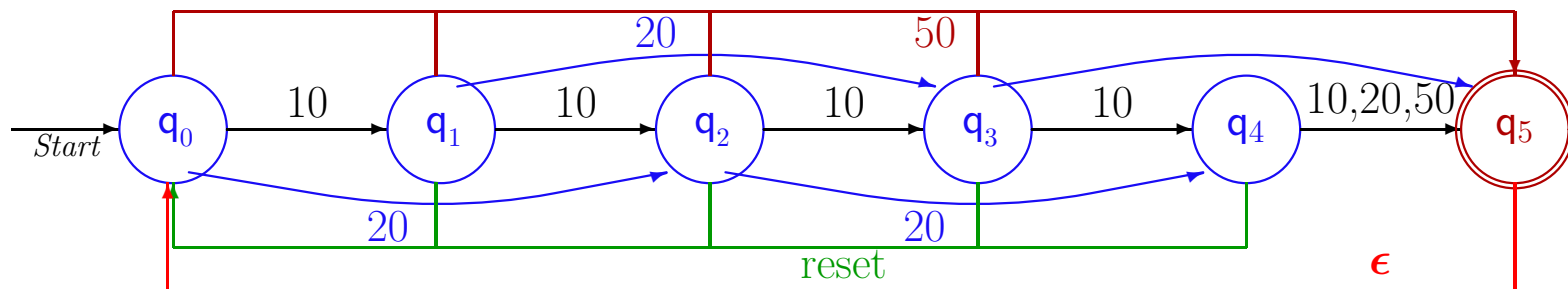
● **Erkenne Dezimalzahlen im Programmcode**

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

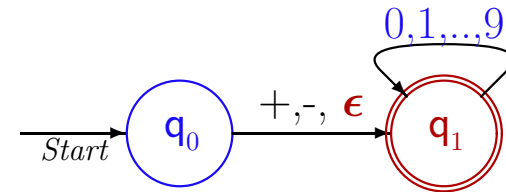
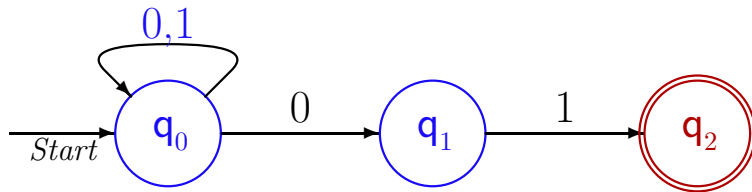


● **50c Kaffeeautomat**

- 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 $\mathbf{A} = (Q, \Sigma, \delta, q_0, F)$ mit

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

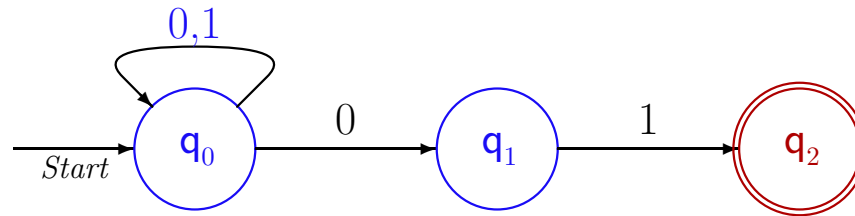
Ein **NEA** ist ein nichtdet. endlicher Automat ohne ϵ -Übergänge

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

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

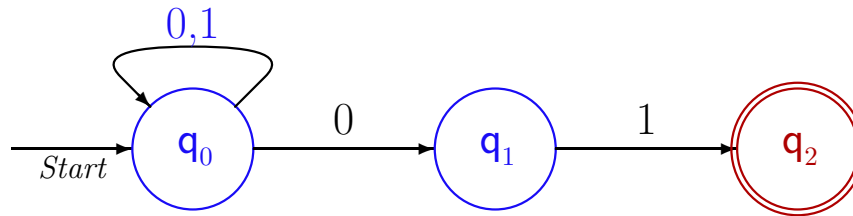
ARBEITSWEISE VON NEAs

Erkenne Strings, die mit 01 enden



ARBEITSWEISE VON NEAs

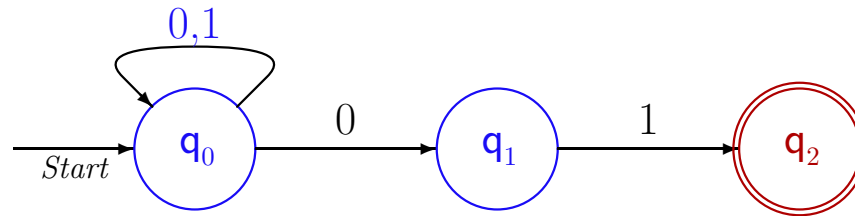
Erkenne Strings, die mit 01 enden



(1) Jedes Teilwort kann in q_0 bleiben

ARBEITSWEISE VON NEAs

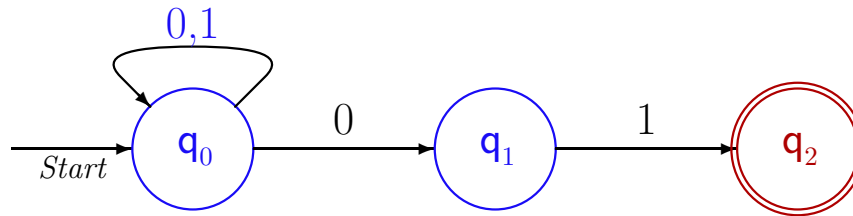
Erkenne Strings, die mit 01 enden



- (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

ARBEITSWEISE VON NEAs

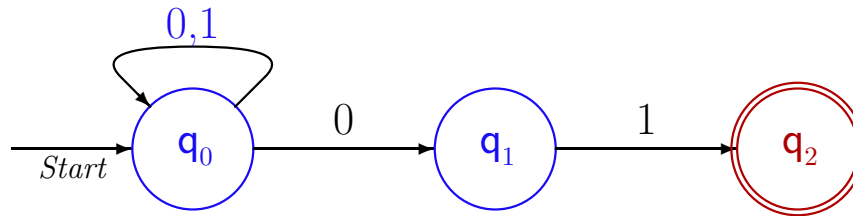
Erkenne Strings, die mit 01 enden



- (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



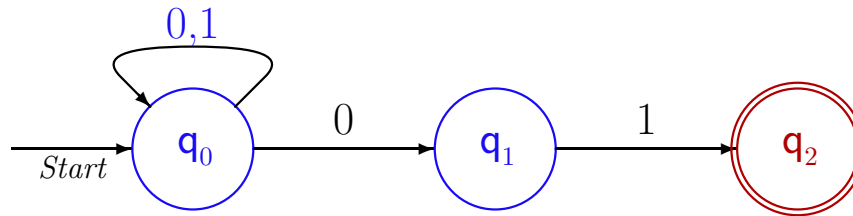
- (1) Jedes Teilwort kann in q_0 bleiben
- (2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen
- (3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen
- (4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101

q_0

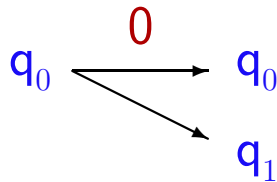
ARBEITSWEISE VON NEAs

Erkenne Strings, die mit 01 enden



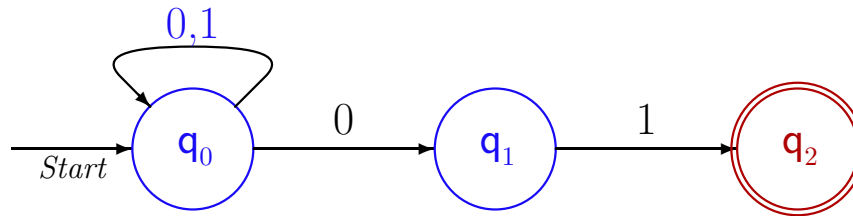
- (1) Jedes Teilwort kann in q_0 bleiben
- (2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen
- (3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen
- (4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101



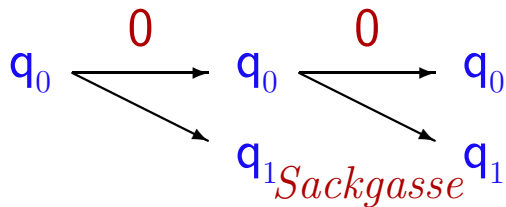
ARBEITSWEISE VON NEAs

Erkenne Strings, die mit 01 enden



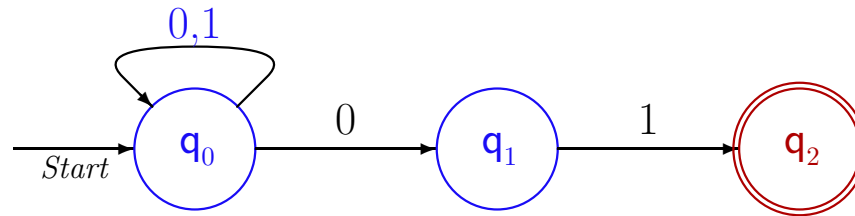
- (1) Jedes Teilwort kann in q_0 bleiben
- (2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen
- (3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen
- (4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101



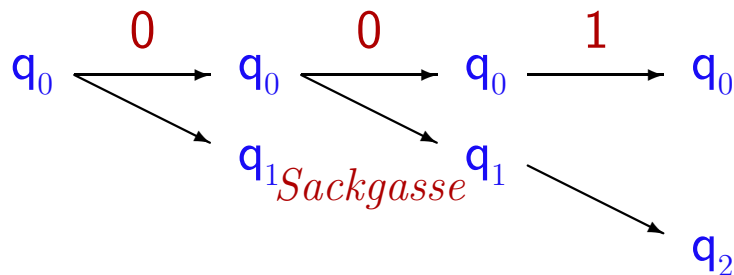
ARBEITSWEISE VON NEAs

Erkenne Strings, die mit 01 enden



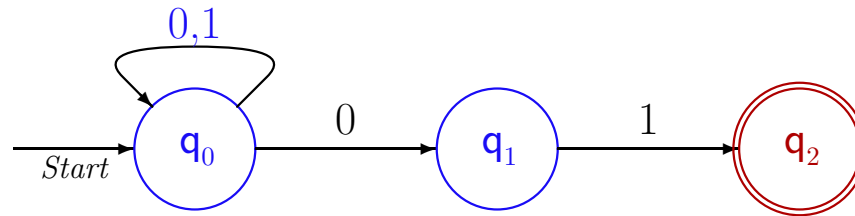
- (1) Jedes Teilwort kann in q_0 bleiben
- (2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen
- (3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen
- (4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101



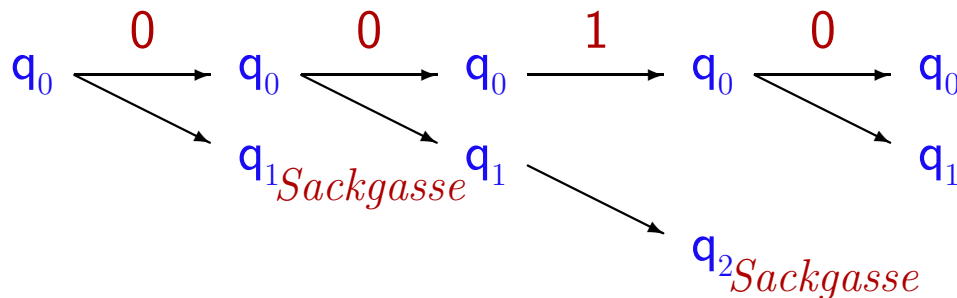
ARBEITSWEISE VON NEAs

Erkenne Strings, die mit 01 enden



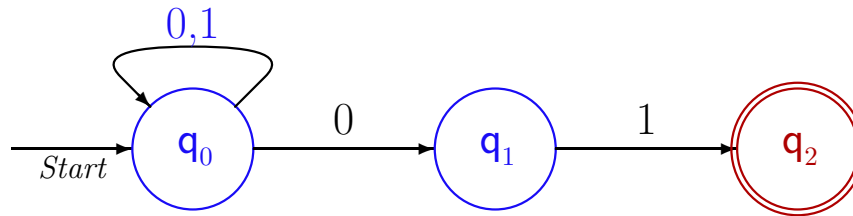
- (1) Jedes Teilwort kann in q_0 bleiben
- (2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen
- (3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen
- (4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101



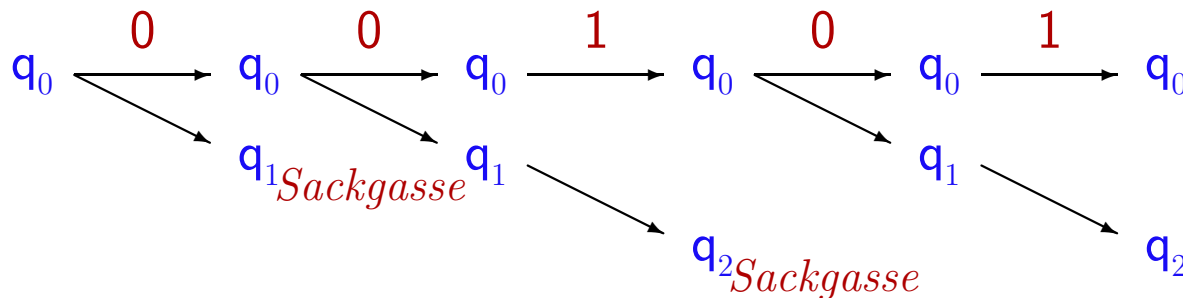
ARBEITSWEISE VON NEAs

Erkenne Strings, die mit 01 enden



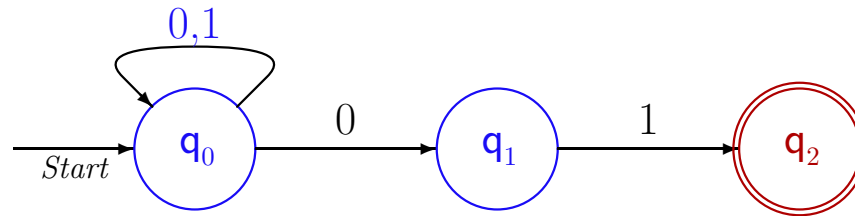
- (1) Jedes Teilwort kann in q_0 bleiben
- (2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen
- (3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen
- (4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101



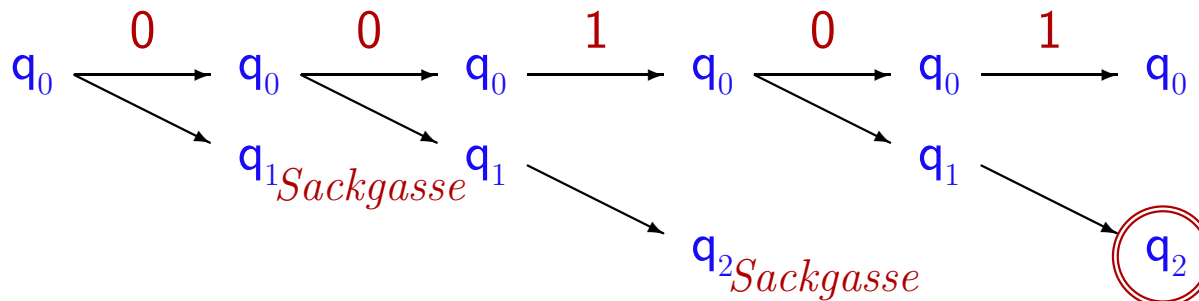
ARBEITSWEISE VON NEAs

Erkenne Strings, die mit 01 enden



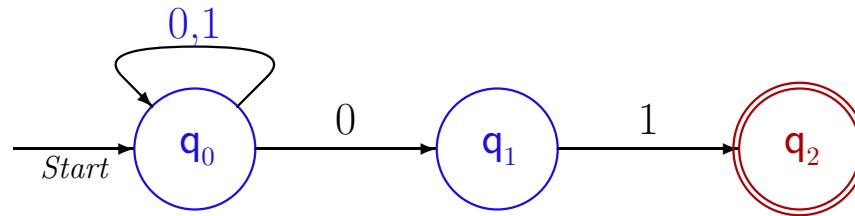
- (1) Jedes Teilwort kann in q_0 bleiben
- (2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen
- (3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen
- (4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101



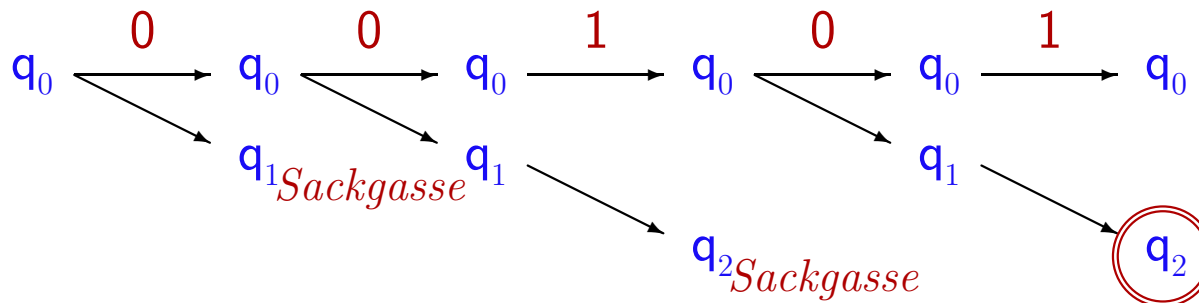
ARBEITSWEISE VON NEAs

Erkenne Strings, die mit 01 enden



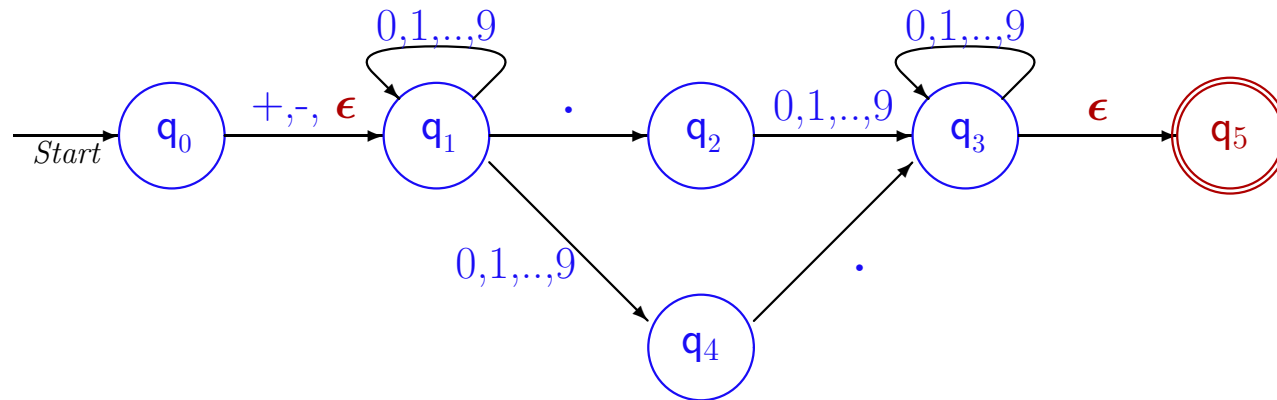
- (1) Jedes Teilwort kann in q_0 bleiben
- (2) Ein Teilwort muss mit 0 enden, um nach q_1 zu führen
- (3) Ein Teilwort muss mit 01 enden, um nach q_2 zu führen
- (4) In q_2 muss das Wort abgearbeitet sein

Beispiel: Abarbeitung von 00101

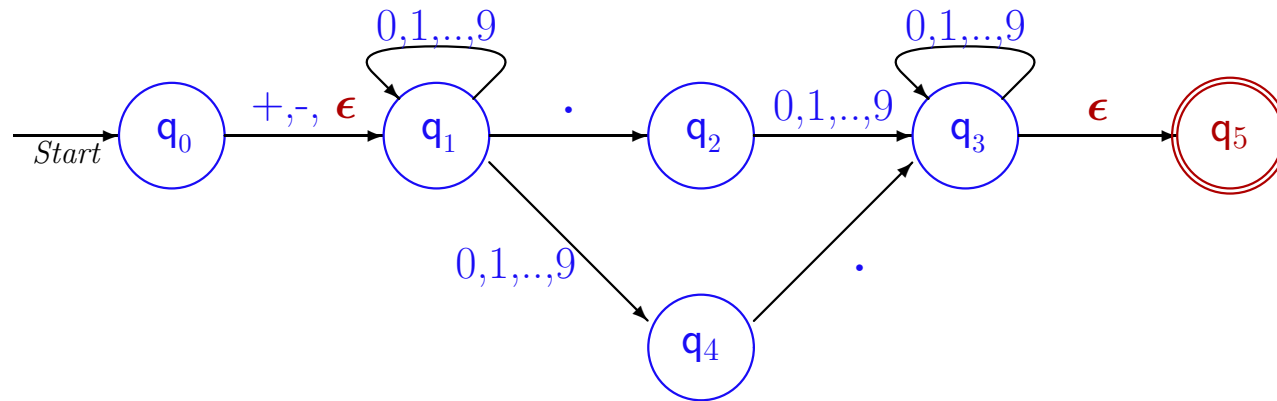


Ein Abarbeitungsweg führt zu einem akzeptierenden Zustand

ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

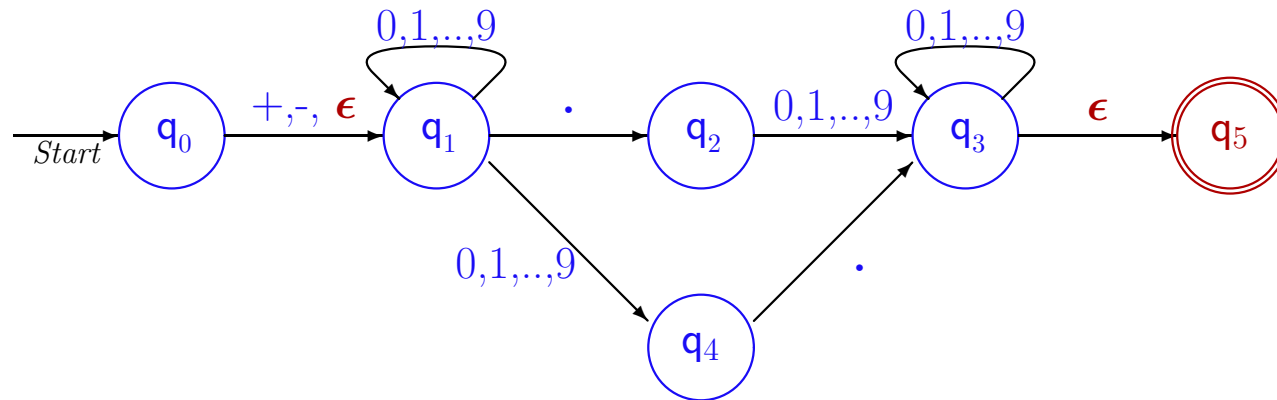


ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN



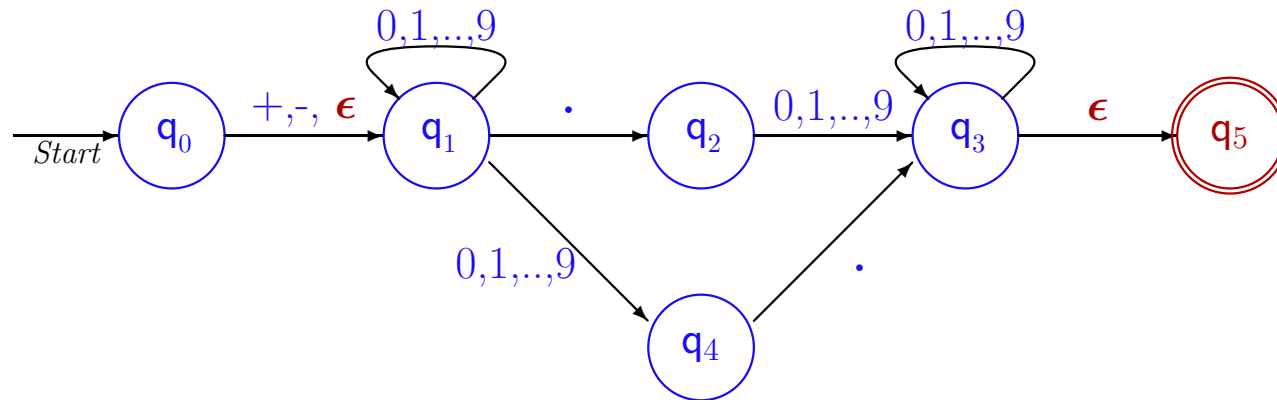
(1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1

ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN



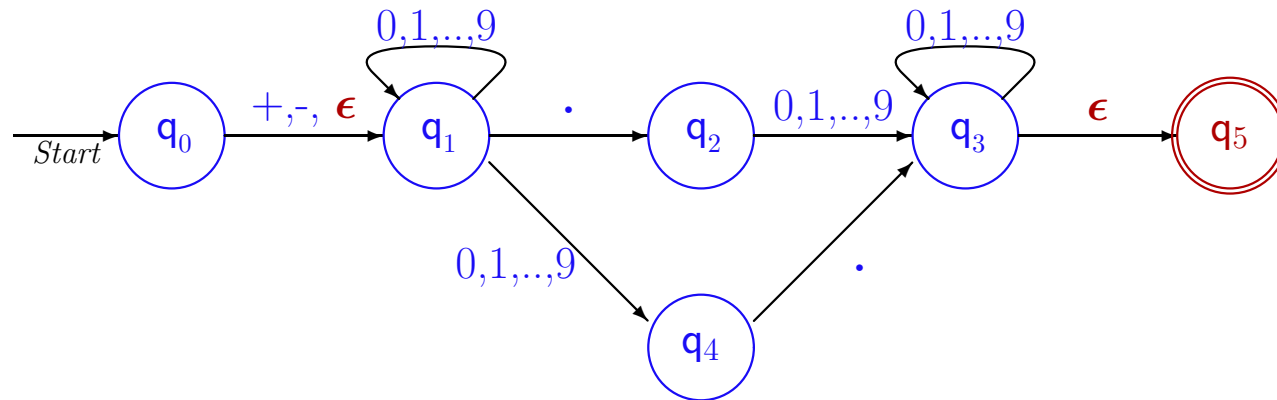
- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0\dots9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4

ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN



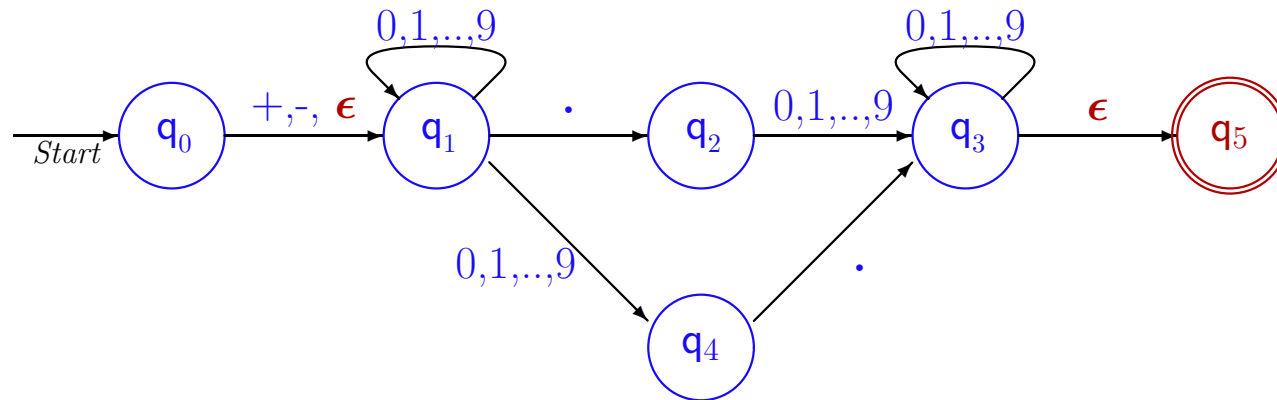
- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3

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

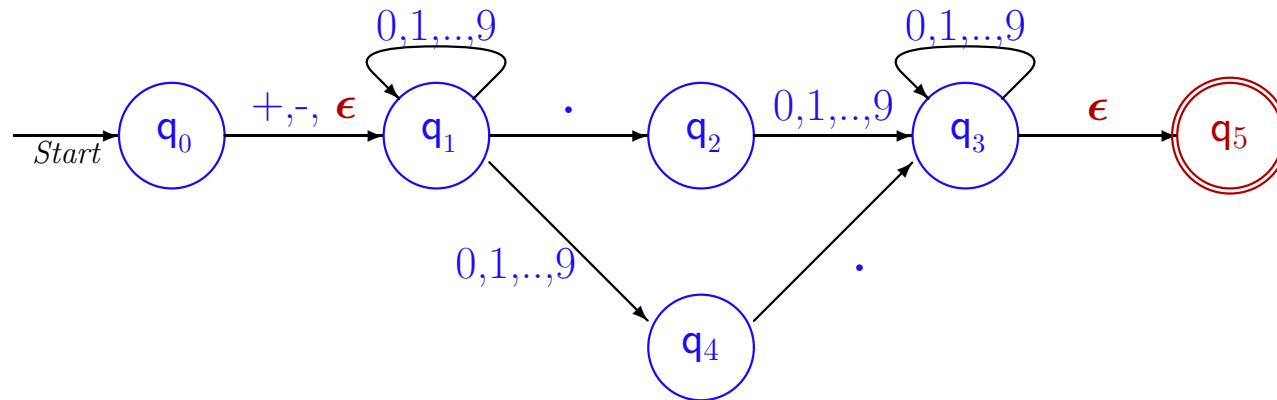


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

q_0

ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

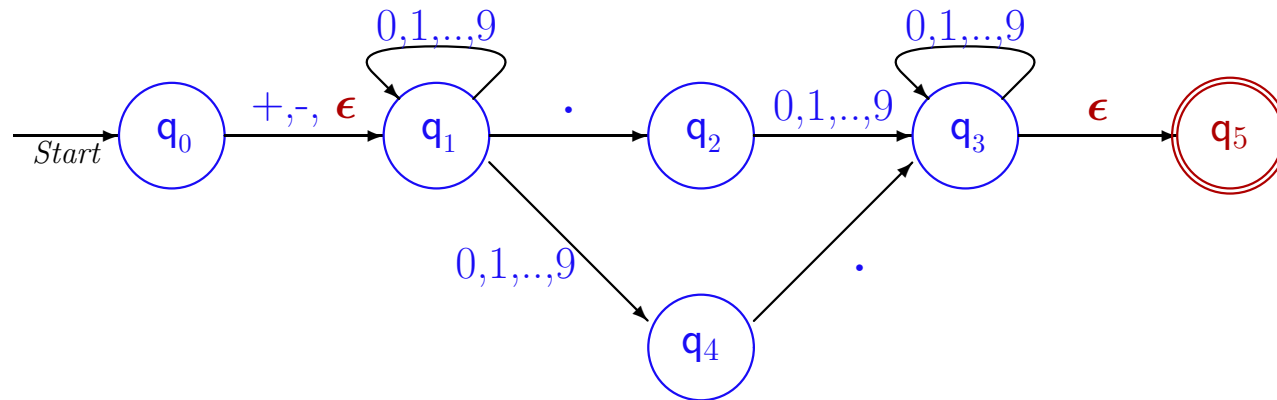


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

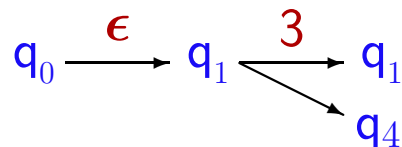
$q_0 \xrightarrow{\epsilon} q_1$

ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

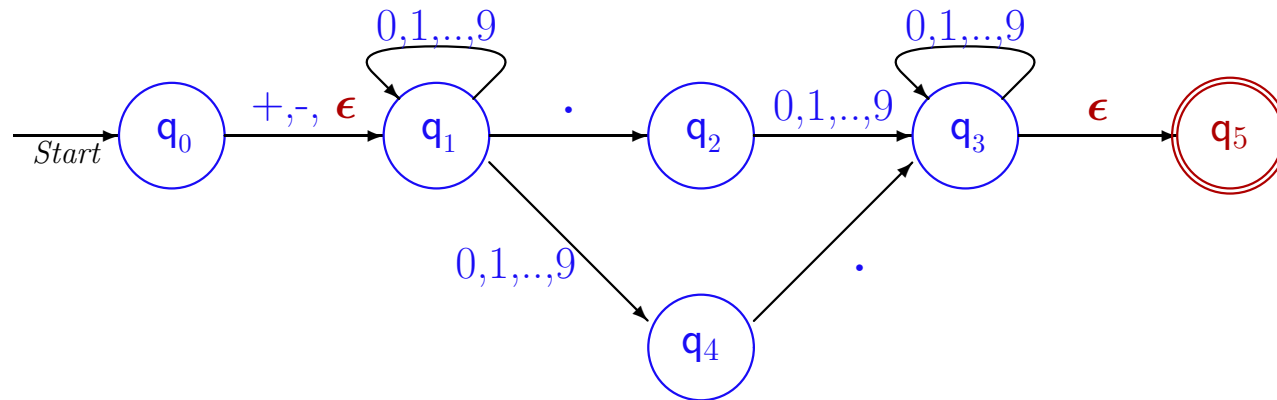


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

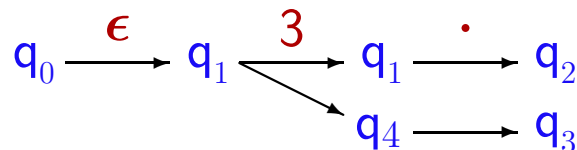


ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

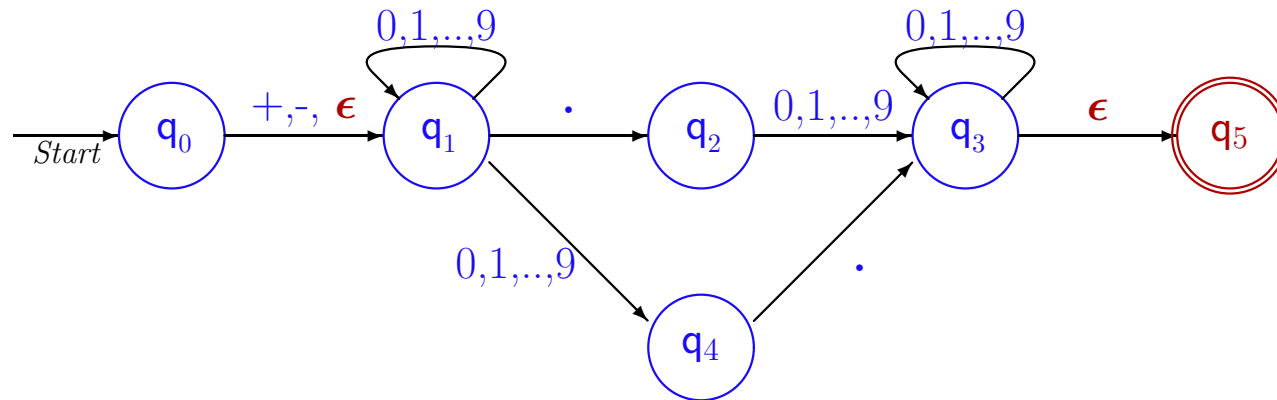


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

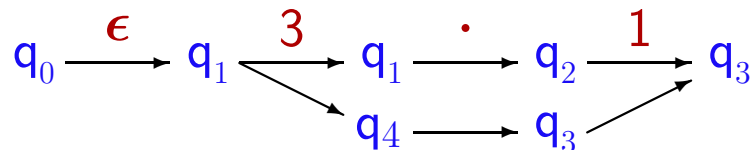


ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

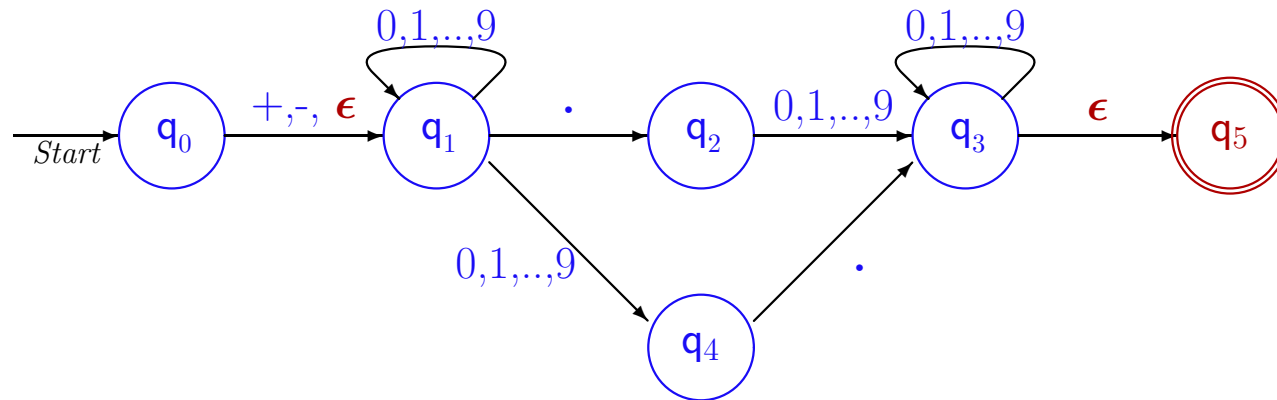


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

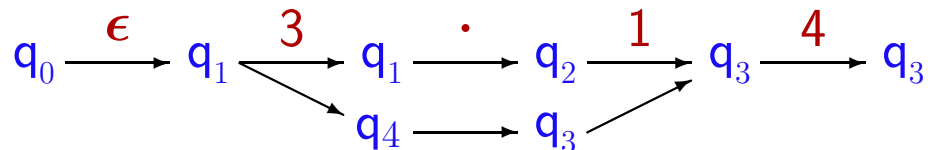


ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

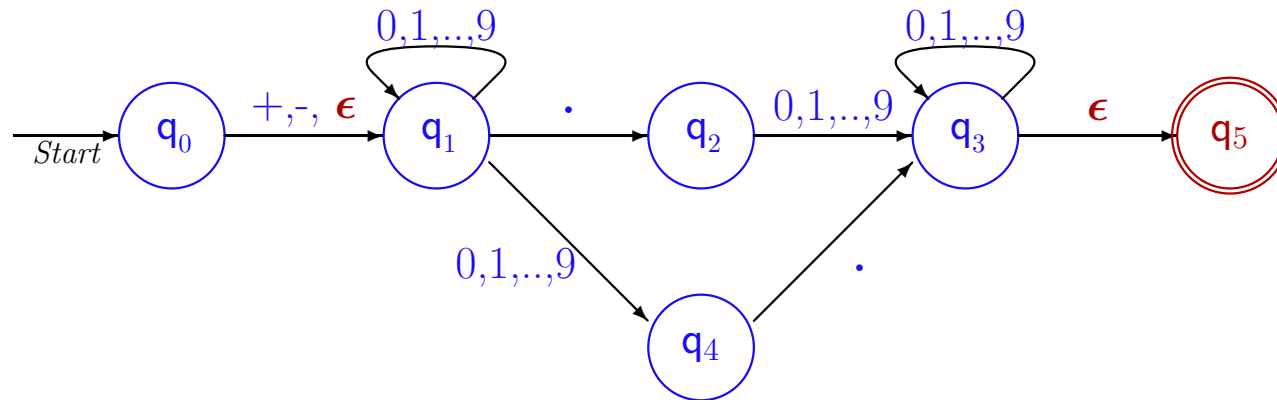


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

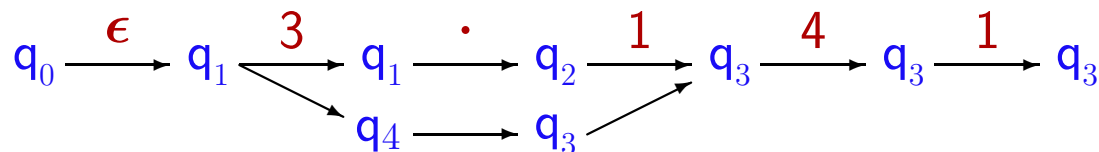


ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

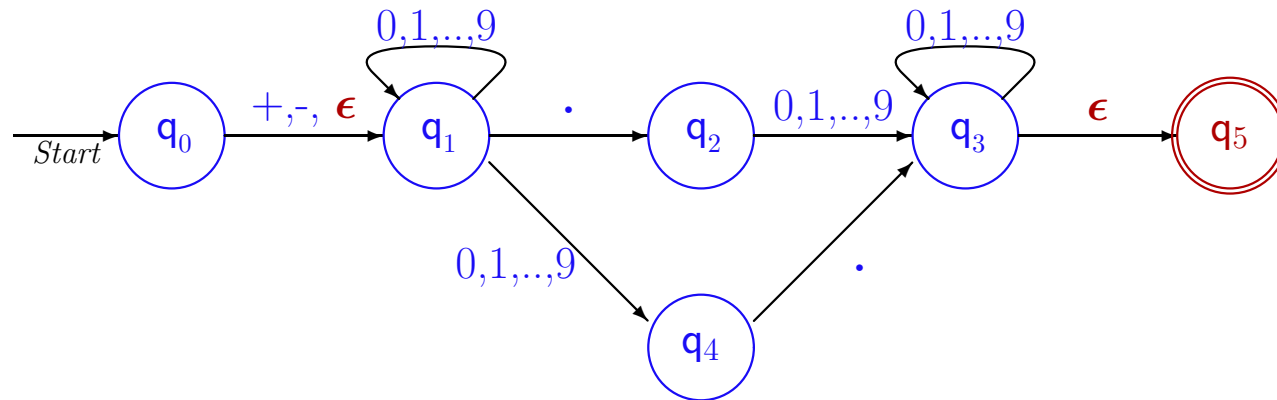


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

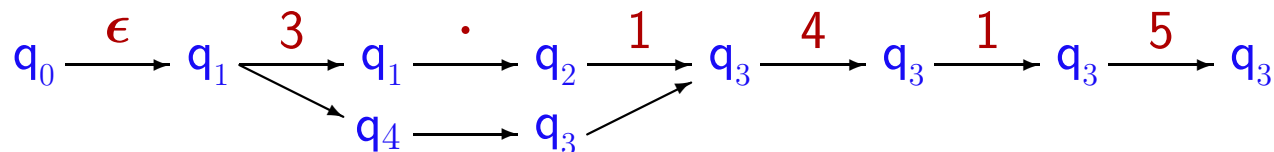


ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

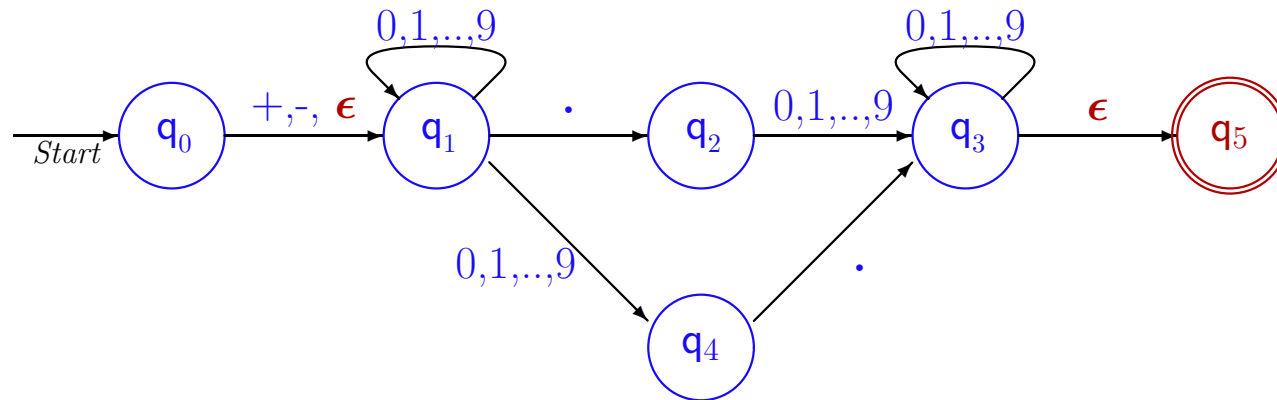


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

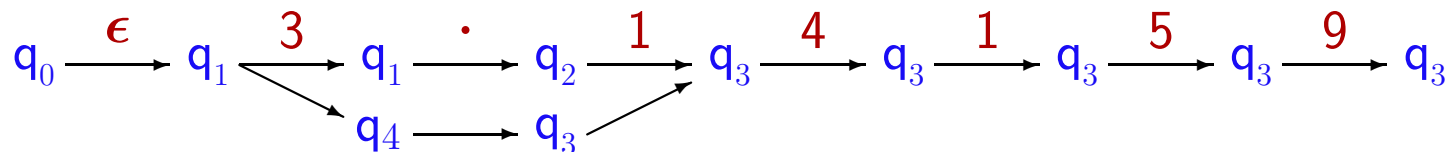


ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

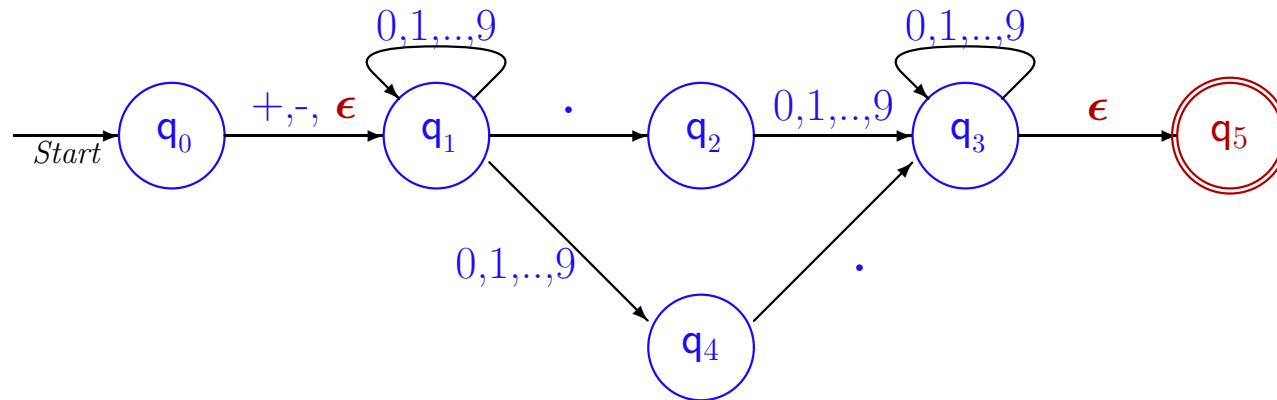


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

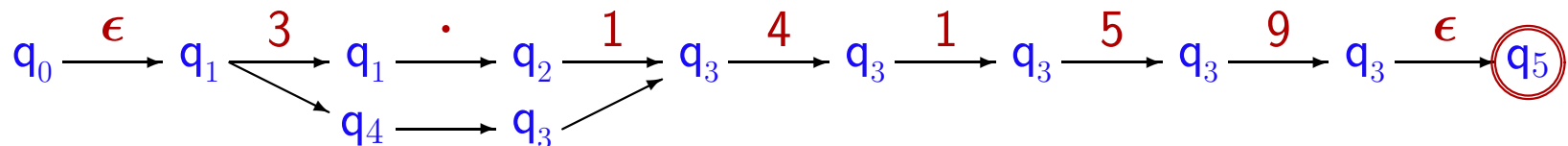


ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN

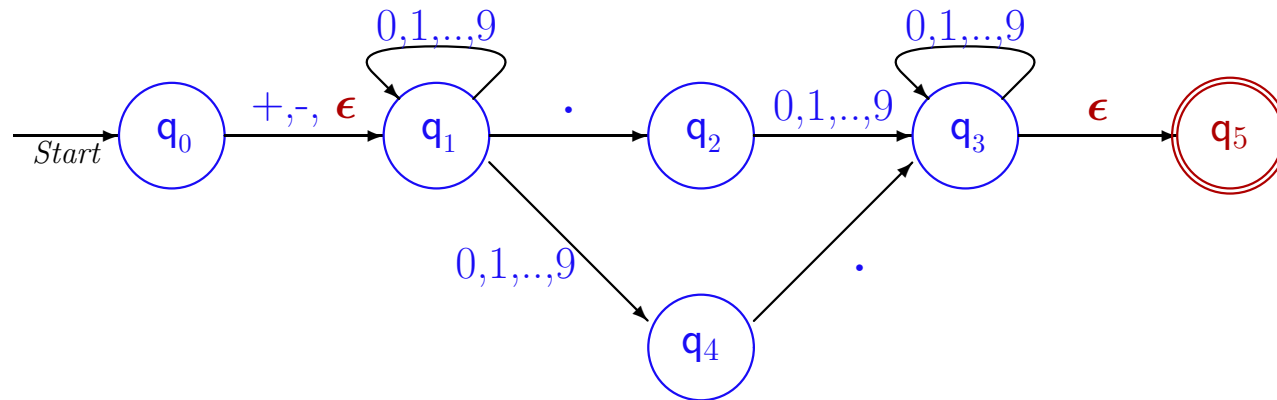


- (1) Die Teilwörter $+$, $-$, und ϵ führen nach q_1
- (2) Teilwörter der Form $v\{0..9\}^+$ mit $v \in \{+, -, \epsilon\}$ führen nach q_1 oder q_4
- (3) Teilwörter der Form $v\{0..9\}^+.$ führen nach q_2 oder q_3
- (4) Teilwörter der Form $v\{0..9\}^*.\{0..9\}^+$ führen nach q_3
- (5) Wörter die nach q_3 führen, führen auch zum Endzustand q_5

Beispiel: Abarbeitung von 3.14159

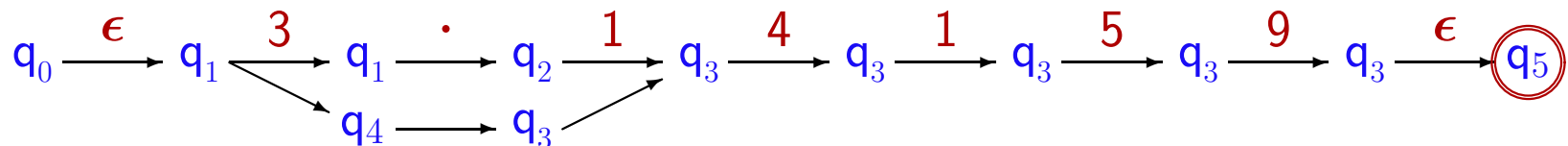


ARBEITSWEISE VON NEAs MIT ϵ -ÜBERGÄNGEN



- (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

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 $p \in \epsilon\text{-Hülle}(q) \wedge r \in \delta(p, \epsilon) \Rightarrow r \in \epsilon\text{-Hülle}(q)$

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
$$q \in \epsilon\text{-Hülle}(q) \text{ und } p \in \epsilon\text{-Hülle}(q) \wedge r \in \delta(p, \epsilon) \Rightarrow r \in \epsilon\text{-Hülle}(q)$$
- **Erweiterte Überföhrungsfunktion $\hat{\delta} : Q \times \Sigma^* \rightarrow \mathcal{P}(Q)$**
 - Aufsammeln **aller** bei der Abarbeitung erreichbaren Zustände einschließlich derjenigen, die ohne Eingabe erreicht werden

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
$$q \in \epsilon\text{-Hülle}(q) \text{ und } p \in \epsilon\text{-Hülle}(q) \wedge r \in \delta(p, \epsilon) \Rightarrow r \in \epsilon\text{-Hülle}(q)$$
- **Erweiterte Überföhrungsfunktion $\hat{\delta} : Q \times \Sigma^* \rightarrow \mathcal{P}(Q)$**
 - Aufsammeln **aller** bei der Abarbeitung erreichbaren Zustände einschließlich derjenigen, die ohne Eingabe erreicht werden
 - **Induktive Definition** (kaskadisches Aufsammeln von Zuständen)

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

* 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 $p \in \epsilon\text{-Hülle}(q'')$

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
$$q \in \epsilon\text{-Hülle}(q) \text{ und } p \in \epsilon\text{-Hülle}(q) \wedge r \in \delta(p, \epsilon) \Rightarrow r \in \epsilon\text{-Hülle}(q)$$
- **Erweiterte Überföhrungsfunktion $\hat{\delta} : Q \times \Sigma^* \rightarrow \mathcal{P}(Q)$**
 - Aufsammeln **aller** bei der Abarbeitung erreichbaren Zustände einschließlich derjenigen, die ohne Eingabe erreicht werden
 - **Induktive Definition** (kaskadisches Aufsammeln von Zuständen)
$$\hat{\delta}(q, w) = \begin{cases} \epsilon\text{-Hülle}(q) & \text{falls } w = \epsilon, \\ \bigcup_{q' \in \hat{\delta}(q, v)} \bigcup_{q'' \in \delta(q', a)} \epsilon\text{-Hülle}(q'') & \text{falls } w = v a \text{ (} a \in \Sigma \text{)} \end{cases}^*$$

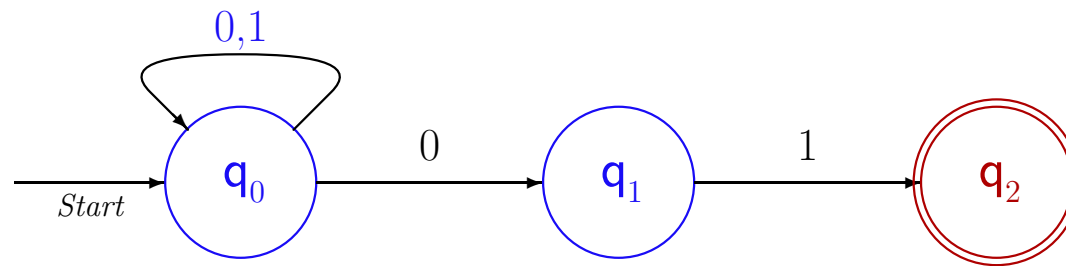
* d.h. $p \in \hat{\delta}(q, w)$ gdw. es gibt ein $q' \in \hat{\delta}(q, v)$ und $q'' \in \delta(q', a)$ so dass $p \in \epsilon\text{-Hülle}(q'')$
- **Von A akzeptierte Sprache**
 - Menge der Eingaben w , für die $\hat{\delta}(q_0, w)$ einen Endzustand enthält

ARBEITSWEISE VON ϵ -NEAS – PRÄZISIERT

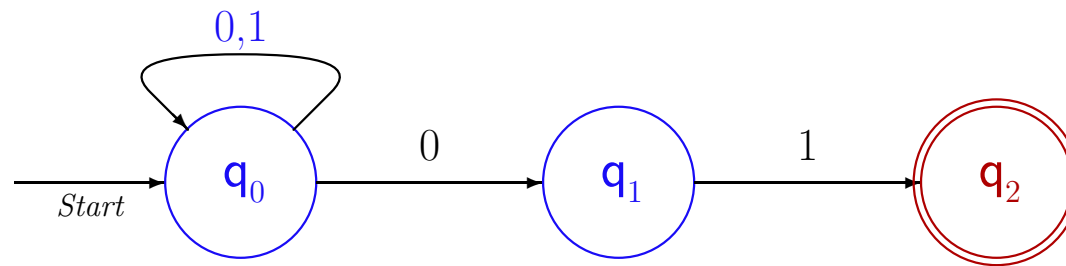
- **Beschreibe ϵ -Hülle** eines Zustands q
 - Die von q mit ϵ -Übergängen erreichbaren Zustände
 - **Iterative Definition**: Kleinste Menge mit der Eigenschaft
$$q \in \epsilon\text{-Hülle}(q) \text{ und } p \in \epsilon\text{-Hülle}(q) \wedge r \in \delta(p, \epsilon) \Rightarrow r \in \epsilon\text{-Hülle}(q)$$
 - **Erweiterte Überföhrungsfunktion $\hat{\delta} : Q \times \Sigma^* \rightarrow \mathcal{P}(Q)$**
 - Aufsammeln **aller** bei der Abarbeitung erreichbaren Zustände einschließlich derjenigen, die ohne Eingabe erreicht werden
 - **Induktive Definition** (kaskadisches Aufsammeln von Zuständen)
$$\hat{\delta}(q, w) = \begin{cases} \epsilon\text{-Hülle}(q) & \text{falls } w=\epsilon, \\ \bigcup_{q' \in \hat{\delta}(q, v)} \bigcup_{q'' \in \delta(q', a)} \epsilon\text{-Hülle}(q'') & \text{falls } w=va \text{ (} a \in \Sigma \text{)} \end{cases}^*$$
- * 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 $p \in \epsilon\text{-Hülle}(q'')$
- **Von A akzeptierte Sprache**
 - Menge der Eingaben w , für die $\hat{\delta}(q_0, w)$ einen Endzustand enthält

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

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

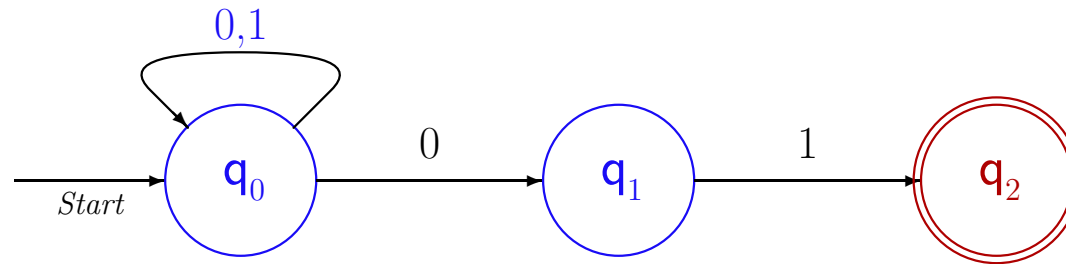


ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (OHNE ϵ -ÜBERGÄNGE)



- Abarbeitung von 00101

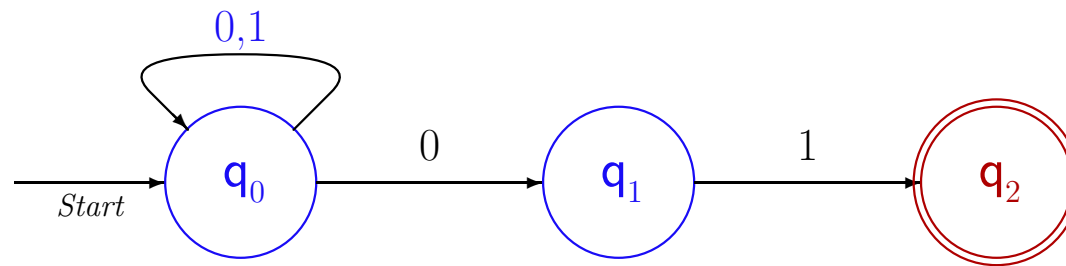
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (OHNE ϵ -ÜBERGÄNGE)



- Abarbeitung von 00101

– $\hat{\delta}(q_0, \epsilon) = \{q_0\}$

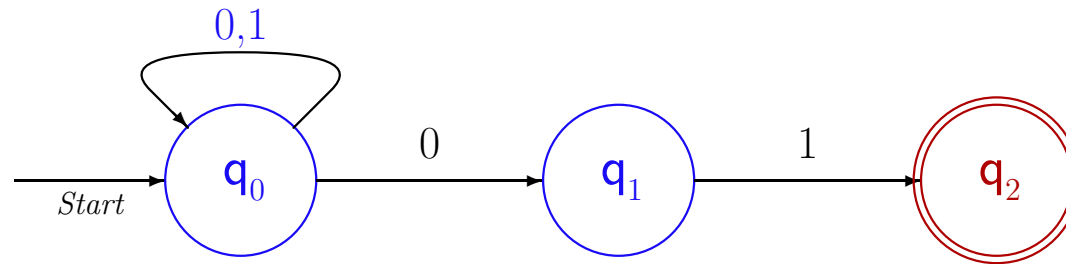
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (OHNE ϵ -ÜBERGÄNGE)



● Abarbeitung von 00101

- $\hat{\delta}(q_0, \epsilon) = \{q_0\}$
- $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$

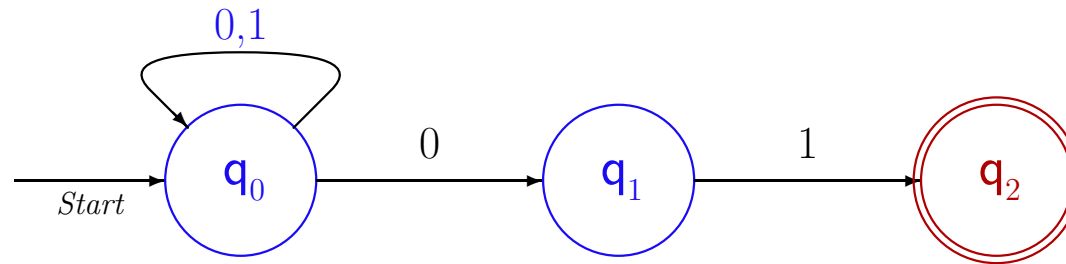
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (OHNE ϵ -ÜBERGÄNGE)



● Abarbeitung von 00101

- $\hat{\delta}(q_0, \epsilon) = \{q_0\}$
- $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$

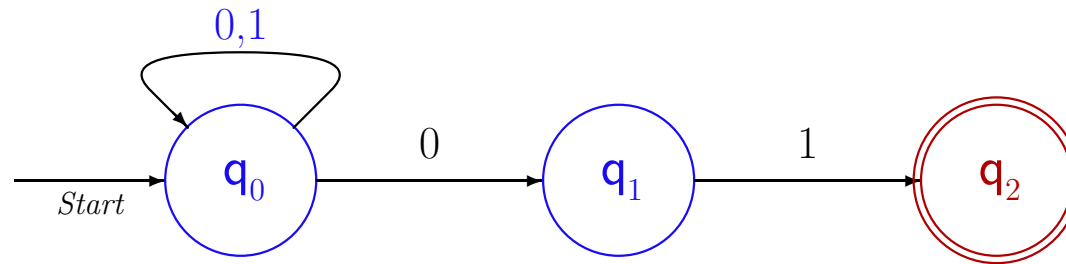
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (OHNE ϵ -ÜBERGÄNGE)



● Abarbeitung von 00101

- $\hat{\delta}(q_0, \epsilon) = \{q_0\}$
- $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$

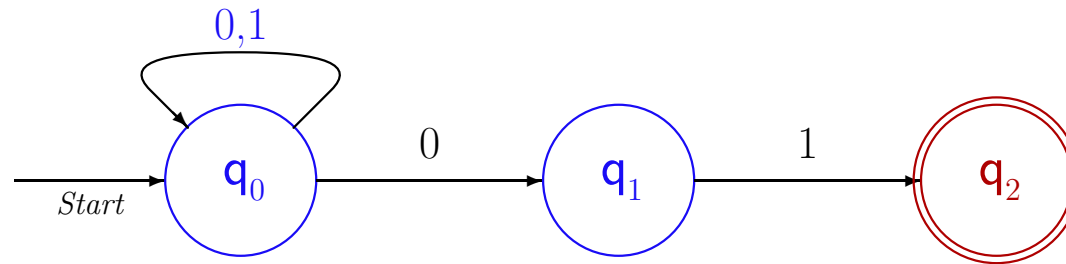
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (OHNE ϵ -ÜBERGÄNGE)



● Abarbeitung von 00101

- $\hat{\delta}(q_0, \epsilon) = \{q_0\}$
- $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$
- $\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$

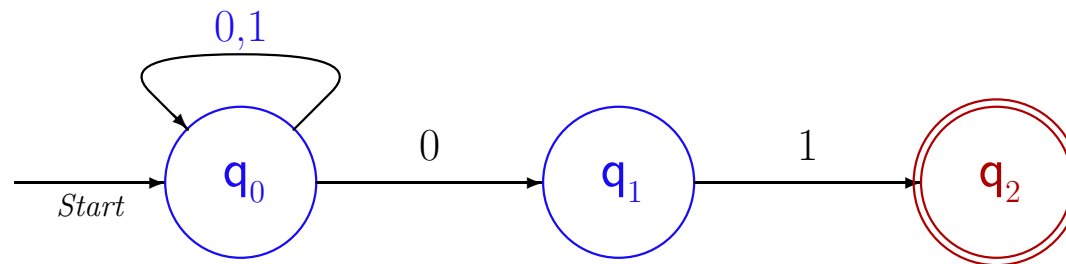
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (OHNE ϵ -ÜBERGÄNGE)



● Abarbeitung von 00101

- $\hat{\delta}(q_0, \epsilon) = \{q_0\}$
- $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$
- $\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 00101) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$

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



● Abarbeitung von 00101

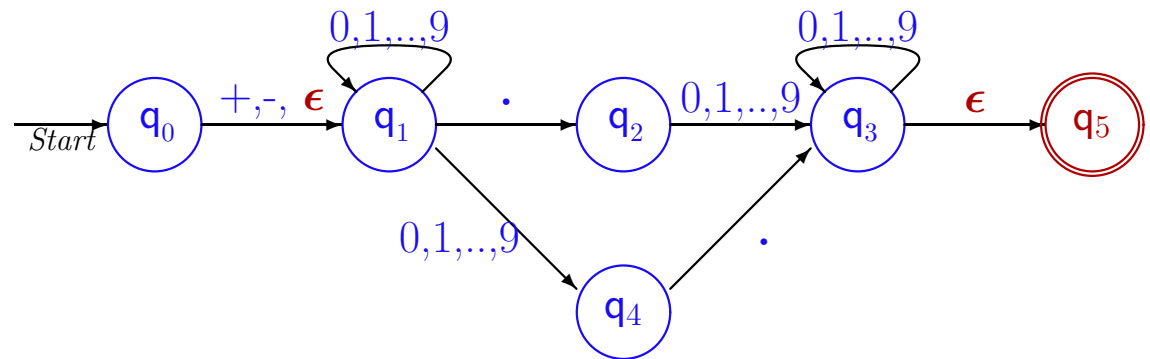
- $\hat{\delta}(q_0, \epsilon) = \{q_0\}$
- $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$
- $\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 00101) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$

● 00101 wird akzeptiert da $\hat{\delta}(q_0, 00101) \cap F = \{q_2\}$

BESTIMMUNG DER ϵ -HÜLLE

● Dezimalautomat

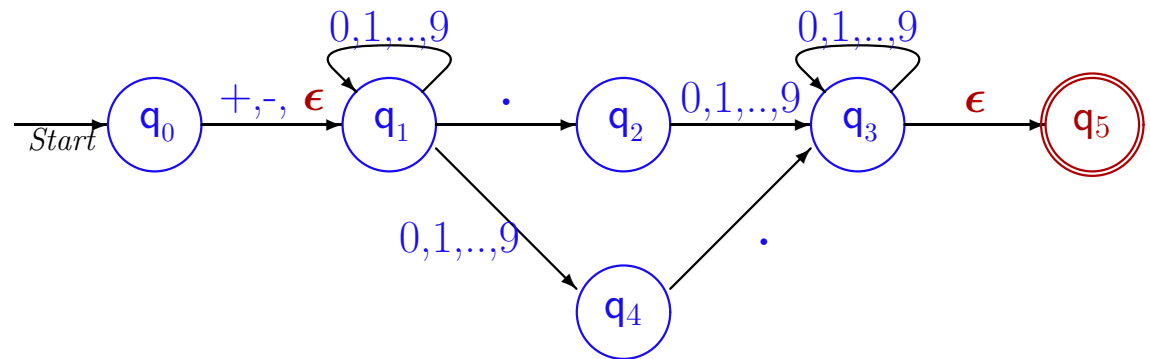
- Nur zwei ϵ -Übergänge



BESTIMMUNG DER ϵ -HÜLLE

● Dezimalautomat

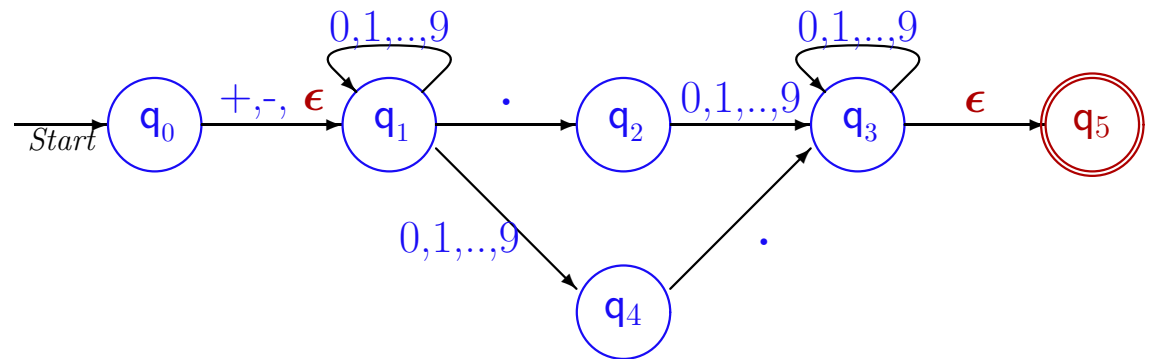
- Nur zwei ϵ -Übergänge
- ϵ -Hülle(q_0) = $\{q_0, q_1\}$



BESTIMMUNG DER ϵ -HÜLLE

● Dezimalautomat

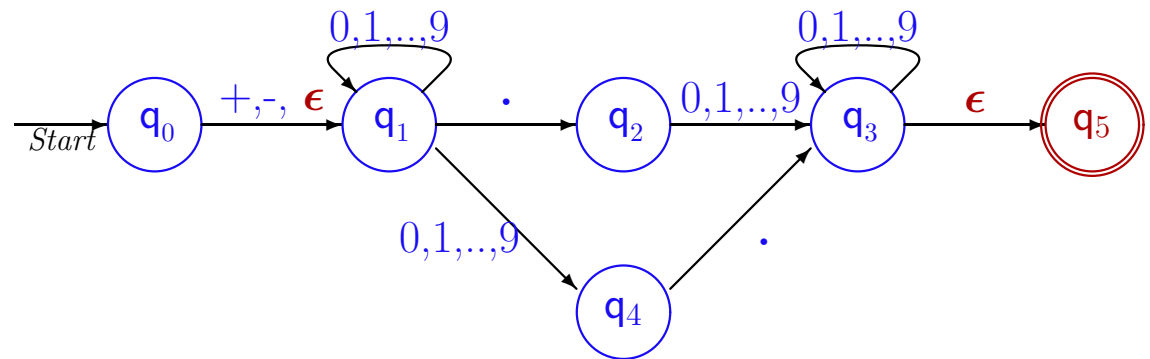
- Nur zwei ϵ -Übergänge
- ϵ -Hülle(q_0) = $\{q_0, q_1\}$
- ϵ -Hülle(q_3) = $\{q_3, q_5\}$



BESTIMMUNG DER ϵ -HÜLLE

● Dezimalautomat

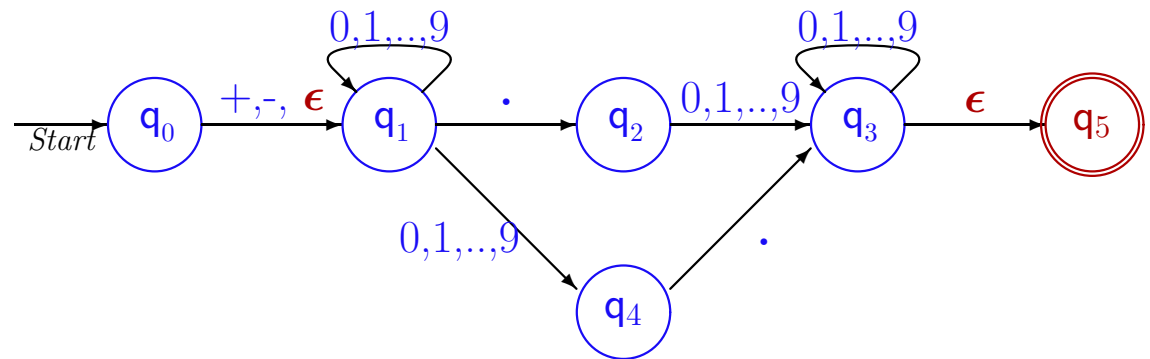
- Nur zwei ϵ -Übergänge
- ϵ -Hülle(q_0) = $\{q_0, q_1\}$
- ϵ -Hülle(q_3) = $\{q_3, q_5\}$
- ϵ -Hülle(q_i) = $\{q_i\}$ sonst



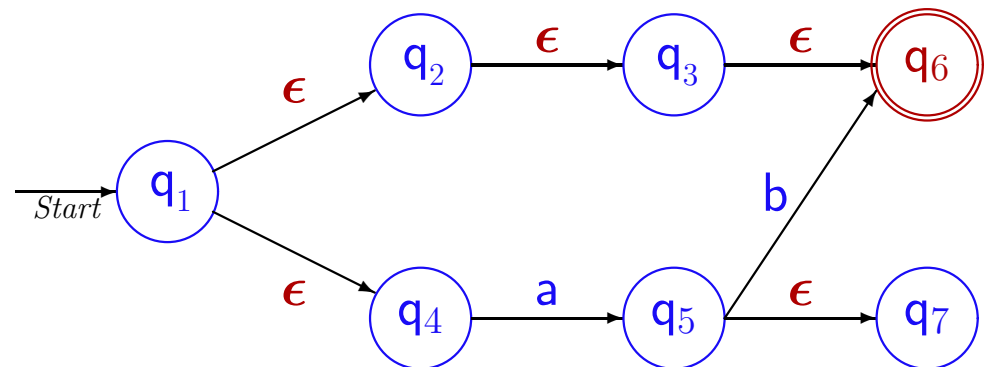
BESTIMMUNG DER ϵ -HÜLLE

● 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



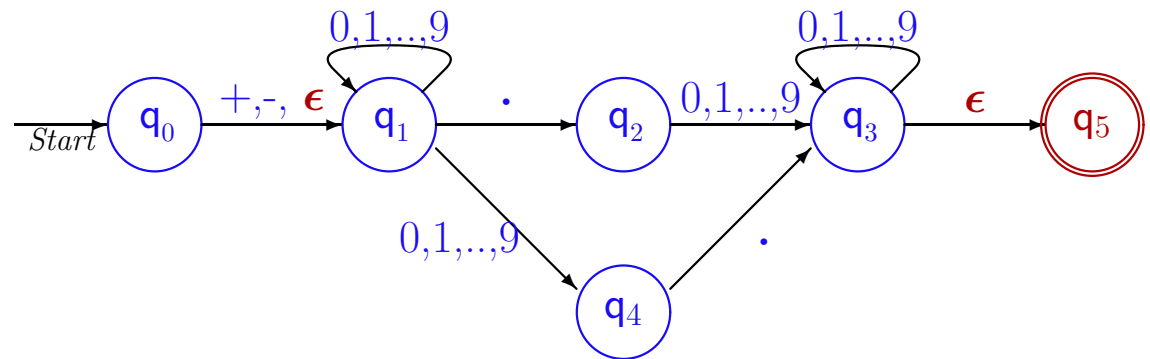
● Viele ϵ -Übergänge



BESTIMMUNG DER ϵ -HÜLLE

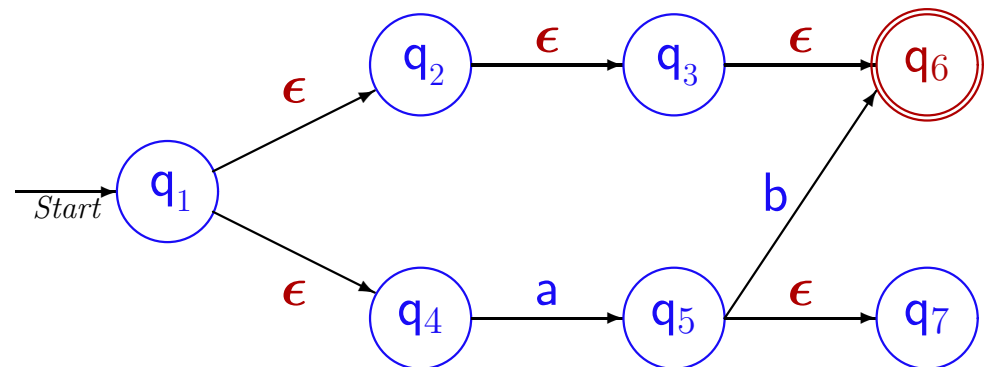
● Dezimalautomat

- 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

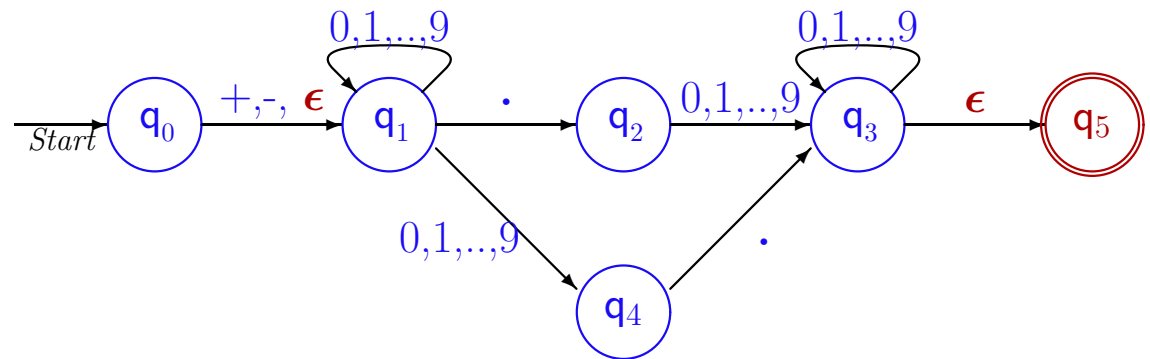
- ϵ -Hülle(q_1) = $\{q_1, q_2, q_3, q_4, q_6\}$



BESTIMMUNG DER ϵ -HÜLLE

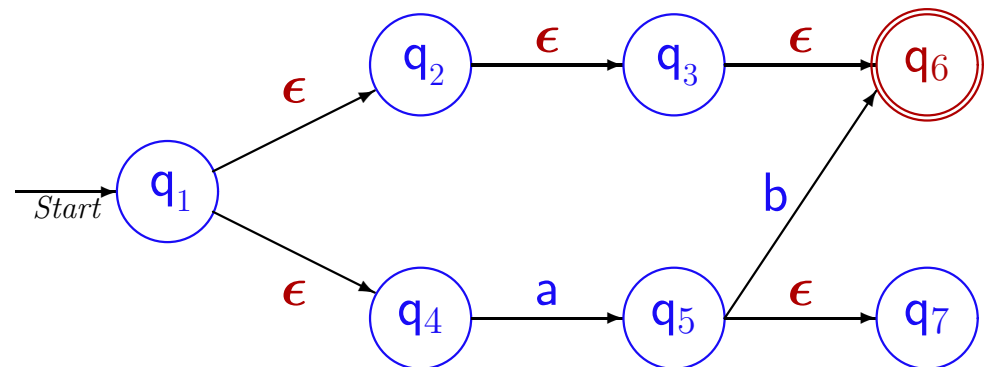
● Dezimalautomat

- 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

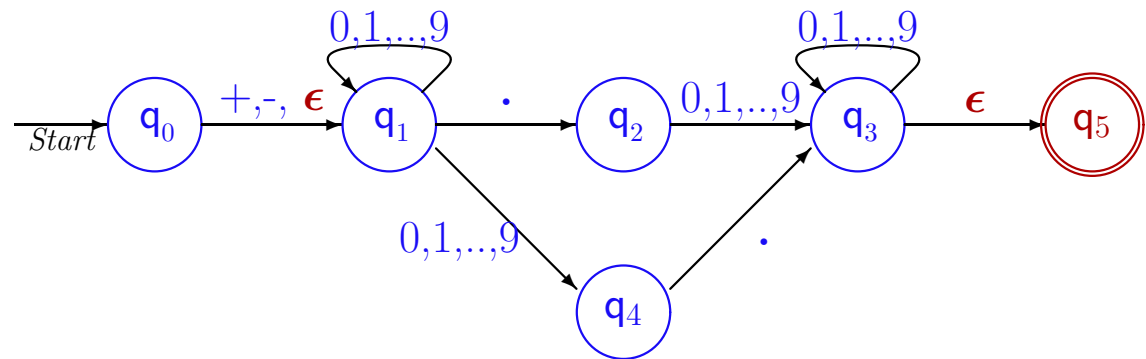
- ϵ -Hülle(q_1) = $\{q_1, q_2, q_3, q_4, q_6\}$
- ϵ -Hülle(q_2) = $\{q_2, q_3, q_6\}$
- ϵ -Hülle(q_3) = $\{q_3, q_6\}$
- ϵ -Hülle(q_4) = $\{q_4\}$



BESTIMMUNG DER ϵ -HÜLLE

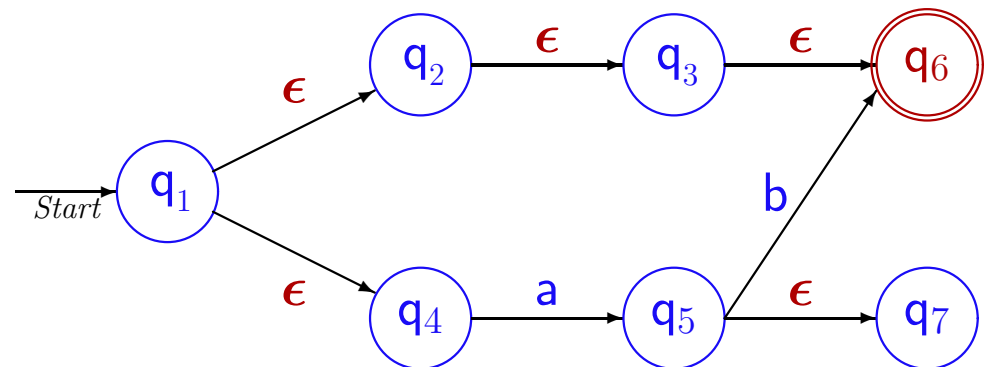
● Dezimalautomat

- 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

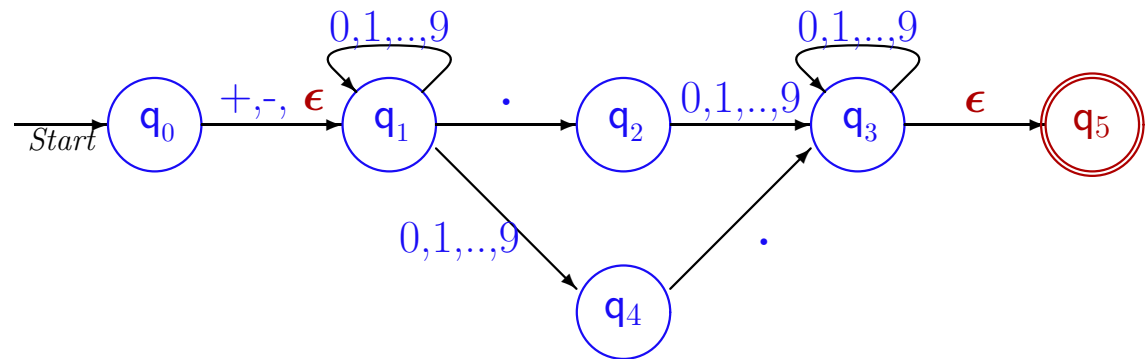
- ϵ -Hülle(q_1) = $\{q_1, q_2, q_3, q_4, q_6\}$
- ϵ -Hülle(q_2) = $\{q_2, q_3, q_6\}$
- ϵ -Hülle(q_3) = $\{q_3, q_6\}$
- ϵ -Hülle(q_4) = $\{q_4\}$
- ϵ -Hülle(q_5) = $\{q_5, q_7\}$



BESTIMMUNG DER ϵ -HÜLLE

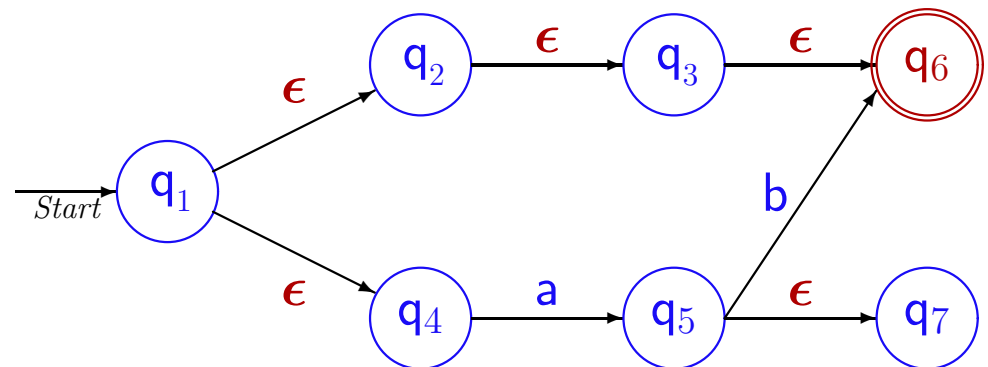
● Dezimalautomat

- 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

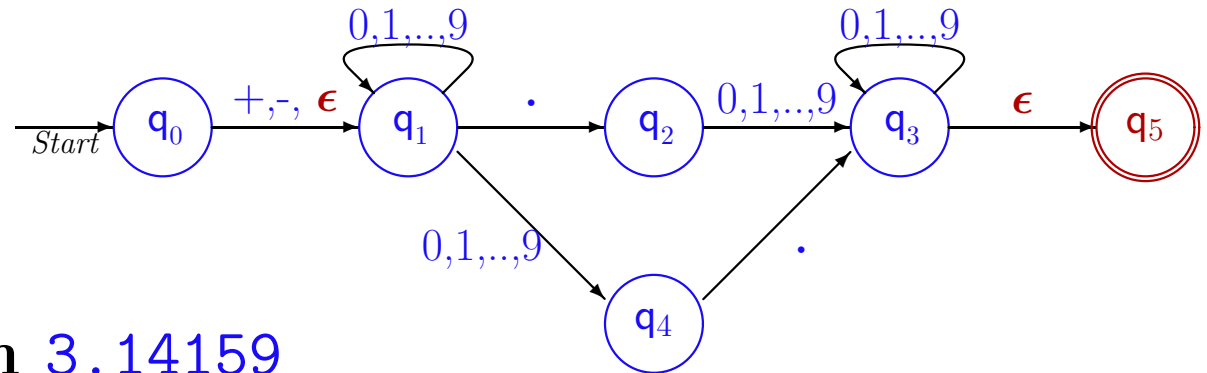


● Viele ϵ -Übergänge

- 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) = $\{q_7\}$

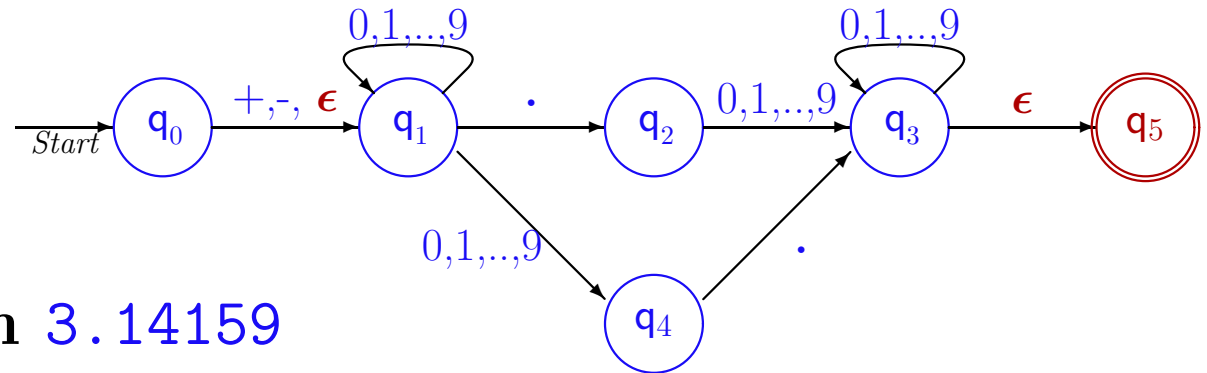


ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (MIT ϵ -ÜBERGÄNGEN)



Abarbeitung von 3.14159

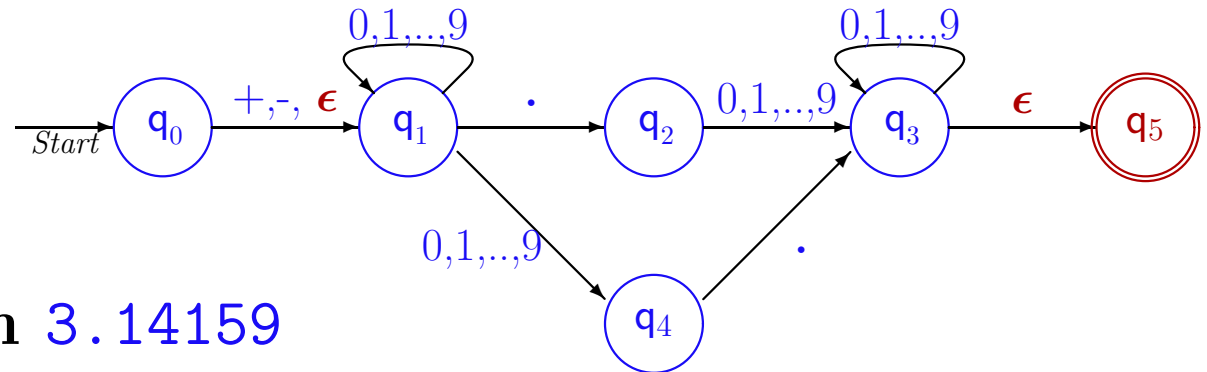
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (MIT ϵ -ÜBERGÄNGEN)



Abarbeitung von 3.14159

– $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-H\ddot{u}lle}(q_0) = \{q_0, q_1\}$

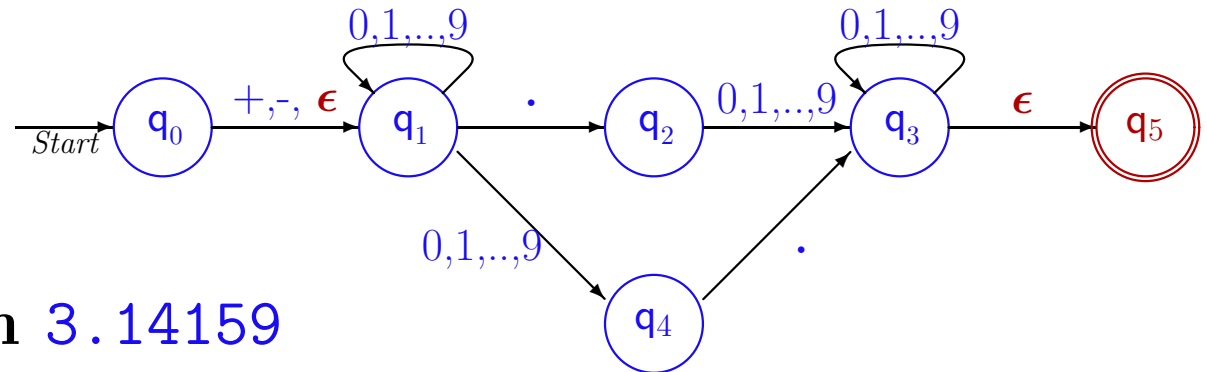
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (MIT ϵ -ÜBERGÄNGEN)



Abarbeitung von 3.14159

- $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-Hülle}(q_0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 3): \delta(q_0, 3) \cup \delta(q_1, 3) = \emptyset \cup \{q_1, q_4\} = \{q_1, q_4\}$

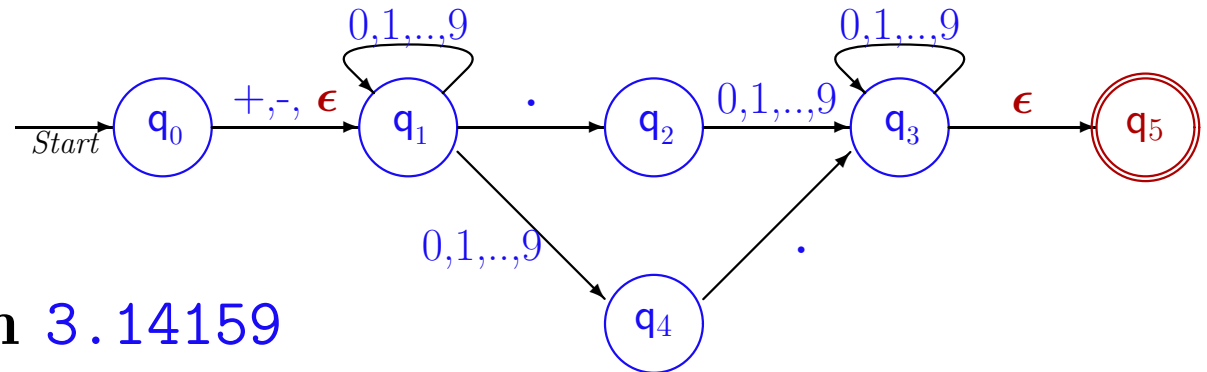
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (MIT ϵ -ÜBERGÄNGEN)



Abarbeitung von 3.14159

- $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-Hülle}(q_0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 3): \delta(q_0, 3) \cup \delta(q_1, 3) = \emptyset \cup \{q_1, q_4\} = \{q_1, q_4\}$
 $\hat{\delta}(q_0, 3) = \epsilon\text{-Hülle}(q_1) \cup \epsilon\text{-Hülle}(q_4) = \{q_1\} \cup \{q_4\} = \{q_1, q_4\}$

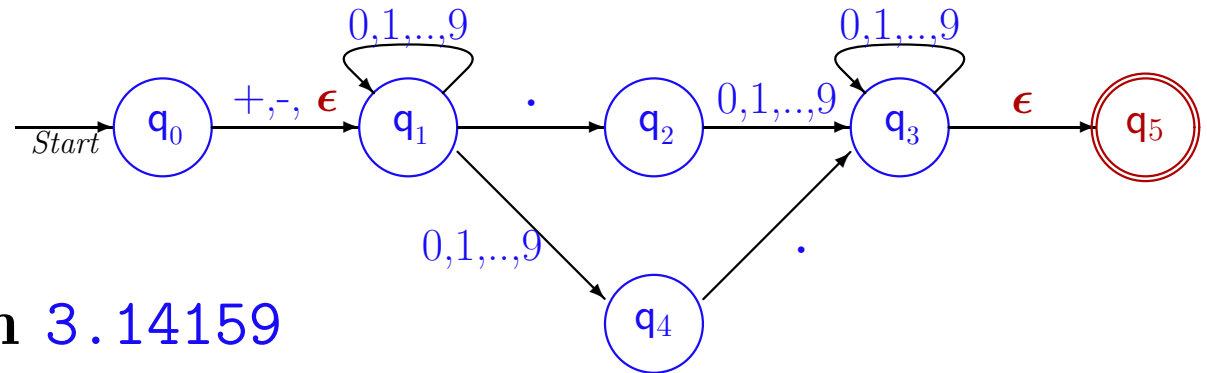
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (MIT ϵ -ÜBERGÄNGEN)



Abarbeitung von 3.14159

- $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-Hülle}(q_0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 3): \delta(q_0, 3) \cup \delta(q_1, 3) = \emptyset \cup \{q_1, q_4\} = \{q_1, q_4\}$
 $\hat{\delta}(q_0, 3) = \epsilon\text{-Hülle}(q_1) \cup \epsilon\text{-Hülle}(q_4) = \{q_1\} \cup \{q_4\} = \{q_1, q_4\}$
- $\hat{\delta}(q_0, 3.): \delta(q_1, .) \cup \delta(q_4, .) = \{q_2\} \cup \{q_3\} = \{q_2, q_3\}$
 $\hat{\delta}(q_0, 3.) = \epsilon\text{-Hülle}(q_2) \cup \epsilon\text{-Hülle}(q_3) = \{q_2\} \cup \{q_3, q_5\} = \{q_2, q_3, q_5\}$

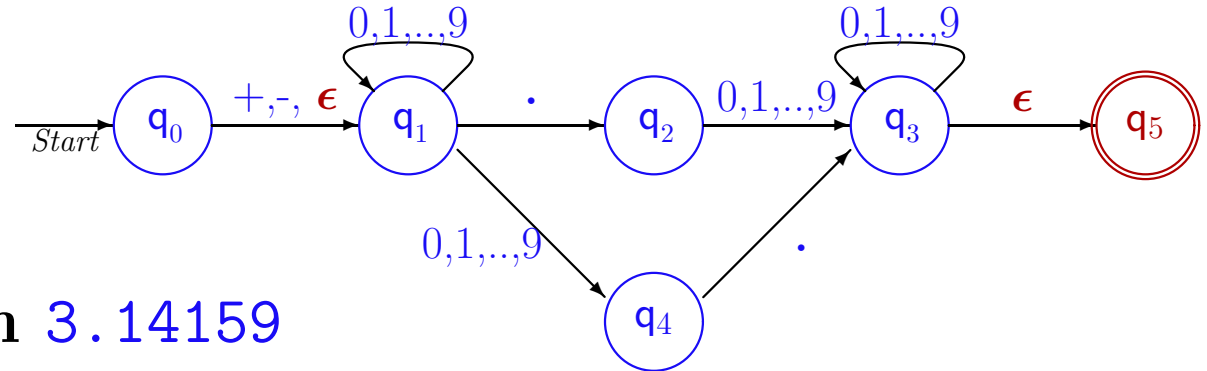
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (MIT ϵ -ÜBERGÄNGEN)



Abarbeitung von 3.14159

- $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-Hülle}(q_0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 3): \delta(q_0, 3) \cup \delta(q_1, 3) = \emptyset \cup \{q_1, q_4\} = \{q_1, q_4\}$
 $\hat{\delta}(q_0, 3) = \epsilon\text{-Hülle}(q_1) \cup \epsilon\text{-Hülle}(q_4) = \{q_1\} \cup \{q_4\} = \{q_1, q_4\}$
- $\hat{\delta}(q_0, 3.): \delta(q_1, .) \cup \delta(q_4, .) = \{q_2\} \cup \{q_3\} = \{q_2, q_3\}$
 $\hat{\delta}(q_0, 3.) = \epsilon\text{-Hülle}(q_2) \cup \epsilon\text{-Hülle}(q_3) = \{q_2\} \cup \{q_3, q_5\} = \{q_2, q_3, q_5\}$
- $\hat{\delta}(q_0, 3.1): \delta(q_2, 1) \cup \delta(q_3, 1) \cup \delta(q_5, 1) = \{q_3\} \cup \{q_3\} \cup \emptyset = \{q_3\}$
 $\hat{\delta}(q_0, 3.1) = \epsilon\text{-Hülle}(q_3) = \{q_3, q_5\}$

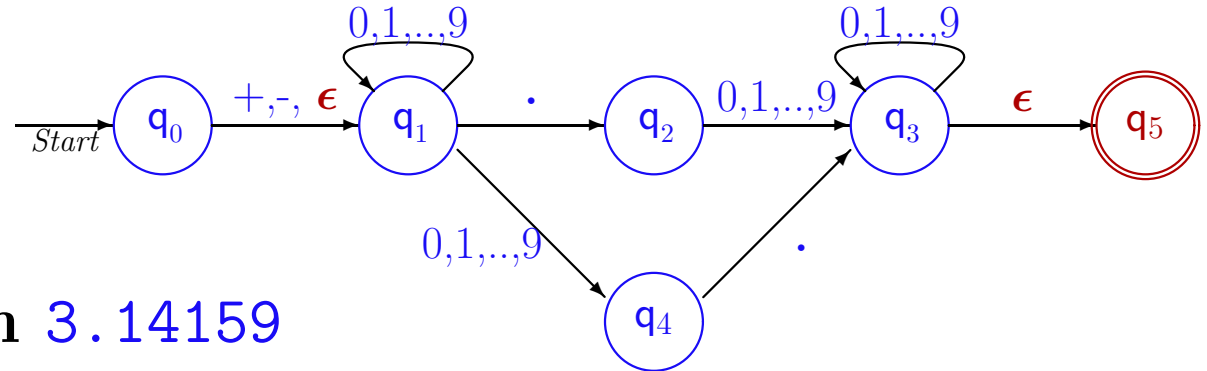
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (MIT ϵ -ÜBERGÄNGEN)



Abarbeitung von 3.14159

- $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-Hülle}(q_0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 3)$: $\delta(q_0, 3) \cup \delta(q_1, 3) = \emptyset \cup \{q_1, q_4\} = \{q_1, q_4\}$
 $\hat{\delta}(q_0, 3) = \epsilon\text{-Hülle}(q_1) \cup \epsilon\text{-Hülle}(q_4) = \{q_1\} \cup \{q_4\} = \{q_1, q_4\}$
- $\hat{\delta}(q_0, 3.)$: $\delta(q_1, .) \cup \delta(q_4, .) = \{q_2\} \cup \{q_3\} = \{q_2, q_3\}$
 $\hat{\delta}(q_0, 3.) = \epsilon\text{-Hülle}(q_2) \cup \epsilon\text{-Hülle}(q_3) = \{q_2\} \cup \{q_3, q_5\} = \{q_2, q_3, q_5\}$
- $\hat{\delta}(q_0, 3.1)$: $\delta(q_2, 1) \cup \delta(q_3, 1) \cup \delta(q_5, 1) = \{q_3\} \cup \{q_3\} \cup \emptyset = \{q_3\}$
 $\hat{\delta}(q_0, 3.1) = \epsilon\text{-Hülle}(q_3) = \{q_3, q_5\}$
- $\hat{\delta}(q_0, 3.14) = \epsilon\text{-Hülle}(q_3) = \{q_3, q_5\}$

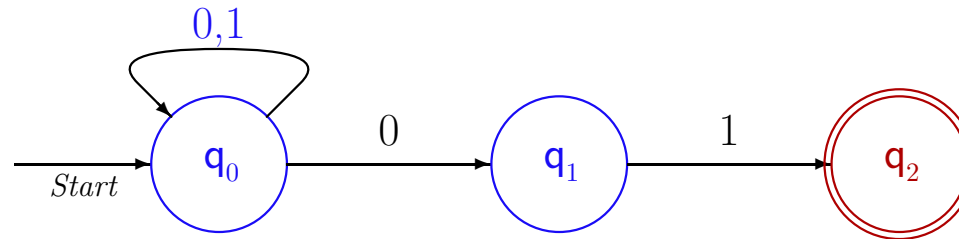
ÜBERFÜHRUNGSFUNKTION $\hat{\delta}$ (MIT ϵ -ÜBERGÄNGEN)



Abarbeitung von 3.14159

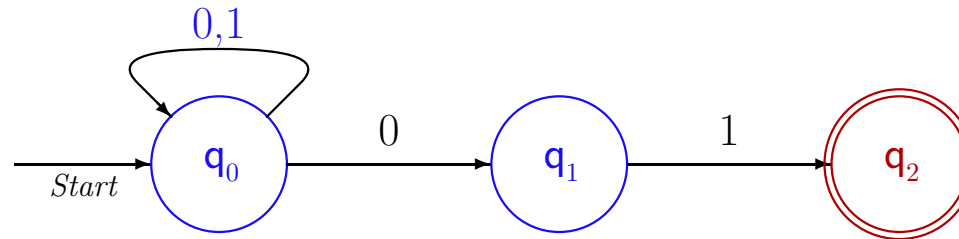
- $\hat{\delta}(q_0, \epsilon) = \epsilon\text{-Hülle}(q_0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 3): \delta(q_0, 3) \cup \delta(q_1, 3) = \emptyset \cup \{q_1, q_4\} = \{q_1, q_4\}$
 $\hat{\delta}(q_0, 3) = \epsilon\text{-Hülle}(q_1) \cup \epsilon\text{-Hülle}(q_4) = \{q_1\} \cup \{q_4\} = \{q_1, q_4\}$
- $\hat{\delta}(q_0, 3.): \delta(q_1, .) \cup \delta(q_4, .) = \{q_2\} \cup \{q_3\} = \{q_2, q_3\}$
 $\hat{\delta}(q_0, 3.) = \epsilon\text{-Hülle}(q_2) \cup \epsilon\text{-Hülle}(q_3) = \{q_2\} \cup \{q_3, q_5\} = \{q_2, q_3, q_5\}$
- $\hat{\delta}(q_0, 3.1): \delta(q_2, 1) \cup \delta(q_3, 1) \cup \delta(q_5, 1) = \{q_3\} \cup \{q_3\} \cup \emptyset = \{q_3\}$
 $\hat{\delta}(q_0, 3.1) = \epsilon\text{-Hülle}(q_3) = \{q_3, q_5\}$
- $\hat{\delta}(q_0, 3.14) = \epsilon\text{-Hülle}(q_3) = \{q_3, q_5\}$
- ⋮
- $\hat{\delta}(q_0, 3.14159) = \epsilon\text{-Hülle}(q_3) = \{q_3, q_5\}$

NACHWEIS DER ERKANNTEN SPRACHE (OHNE ϵ -ÜBERGÄNGE)



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

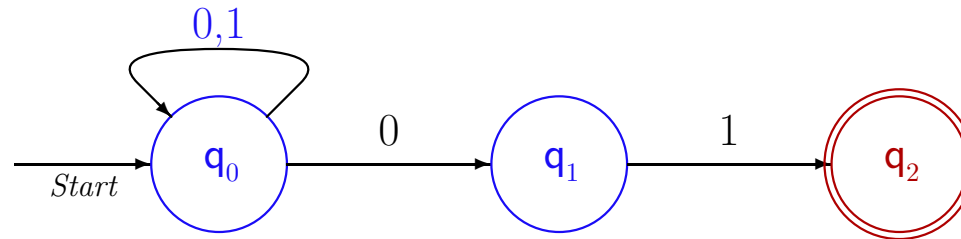
NACHWEIS DER ERKANNTEN SPRACHE (OHNE ϵ -ÜBERGÄNGE)



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

- Zeige durch **simultane Induktion** für alle $w \in \{0, 1\}^*$
 - a) $q_0 \in \hat{\delta}(q_0, w)$
 - b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
 - c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

NACHWEIS DER ERKANNTEN SPRACHE (OHNE ϵ -ÜBERGÄNGE)



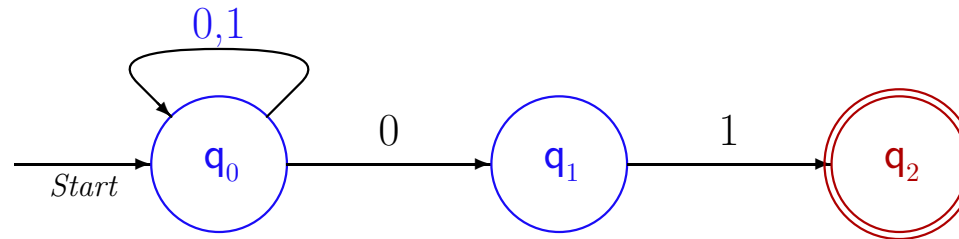
$$L(A) = \{w \in \{0, 1\}^* \mid w \text{ endet mit } 01\}$$

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

- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

Es 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)



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

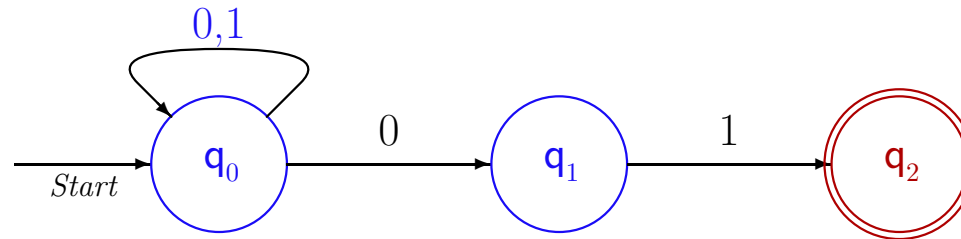
- Zeige durch **simultane Induktion** für alle $w \in \{0, 1\}^*$

- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

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

- Induktionsanfang $w = \epsilon$

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 $w \in \{0, 1\}^*$

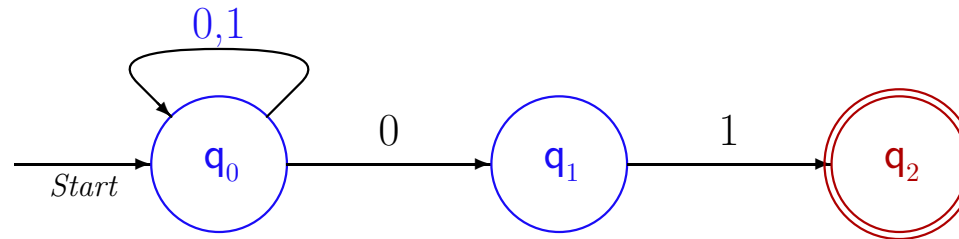
- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

Es 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 Aussage a)

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 $w \in \{0, 1\}^*$

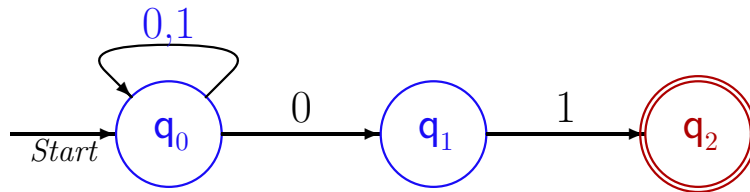
- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

Es 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 Aussage a)
- w endet weder mit 0 noch mit 01. Aussagen b) und c) gelten trivialerweise

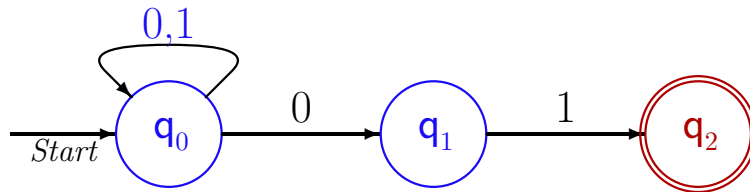
NACHWEIS DER ERKANNTEN SPRACHE II



- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

- Induktionsschritt: $w = va$ für $v \in \{0, 1\}^*$, $a \in \{0, 1\}$

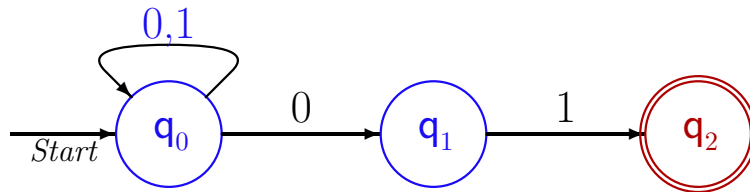
NACHWEIS DER ERKANNTEN SPRACHE II



- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

- **Induktionsschritt:** $w = va$ für $v \in \{0, 1\}^*$, $a \in \{0, 1\}$
 - Die Aussagen a), b), und c) seien für v gültig

NACHWEIS DER ERKANNTEN SPRACHE II



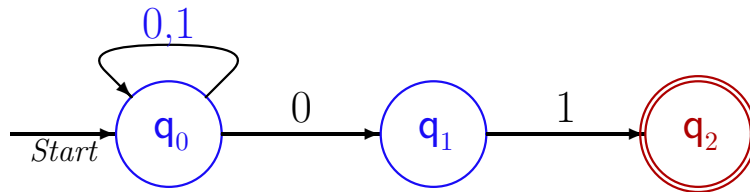
- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

● Induktionsschritt: $w = va$ für $v \in \{0, 1\}^*$, $a \in \{0, 1\}$

– Die Aussagen a), b), und c) seien für v gültig

a) Wegen $q_0 \in \hat{\delta}(q_0, v)$ 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



- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

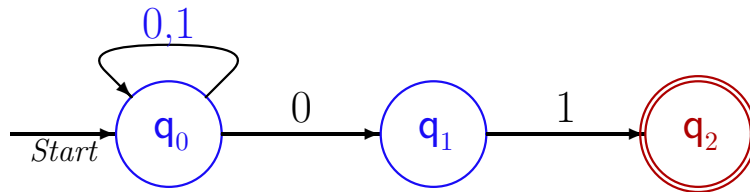
● Induktionsschritt: $w = va$ für $v \in \{0, 1\}^*$, $a \in \{0, 1\}$

– Die Aussagen a), b), und c) seien für v gültig

a) Wegen $q_0 \in \hat{\delta}(q_0, v)$ und $q_0 \in \delta(q_0, a)$ für $a \in \{0, 1\}$ folgt $q_0 \in \hat{\delta}(q_0, w)$

b) 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

NACHWEIS DER ERKANNTEN SPRACHE II



- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

● Induktionsschritt: $w = va$ für $v \in \{0, 1\}^*$, $a \in \{0, 1\}$

– Die Aussagen a), b), und c) seien für v gültig

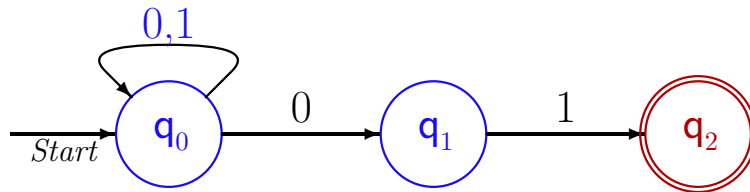
a) Wegen $q_0 \in \hat{\delta}(q_0, v)$ und $q_0 \in \delta(q_0, a)$ für $a \in \{0, 1\}$ folgt $q_0 \in \hat{\delta}(q_0, w)$

b) 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 $q_0 \in \hat{\delta}(q_0, v)$ und $q_1 \in \delta(q_0, a)$ folgt $q_1 \in \hat{\delta}(q_0, w)$

NACHWEIS DER ERKANNTEN SPRACHE II



- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

● Induktionsschritt: $w = va$ für $v \in \{0, 1\}^*$, $a \in \{0, 1\}$

– Die Aussagen a), b), und c) seien für v gültig

a) Wegen $q_0 \in \hat{\delta}(q_0, v)$ und $q_0 \in \delta(q_0, a)$ für $a \in \{0, 1\}$ folgt $q_0 \in \hat{\delta}(q_0, w)$

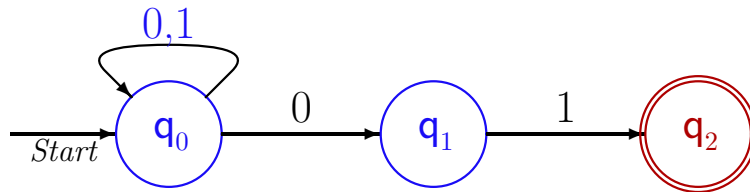
b) Sei $q_1 \in \hat{\delta}(q_0, w)$. Wegen $q_1 \in \delta(q, a) \Leftrightarrow q = q_0 \wedge a = 0$ muss w mit 0 enden

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

Wegen $q_0 \in \hat{\delta}(q_0, v)$ und $q_1 \in \delta(q_0, a)$ folgt $q_1 \in \hat{\delta}(q_0, w)$

c) 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 $q_1 \in \hat{\delta}(q_0, v)$ gelten. Wegen b) für v endet v mit 0, also w mit 01

NACHWEIS DER ERKANNTEN SPRACHE II



- a) $q_0 \in \hat{\delta}(q_0, w)$
- b) $q_1 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 0 endet
- c) $q_2 \in \hat{\delta}(q_0, w)$ genau dann, wenn w mit 01 endet

● Induktionsschritt: $w = va$ für $v \in \{0, 1\}^*$, $a \in \{0, 1\}$

– Die Aussagen a), b), und c) seien für v gültig

a) Wegen $q_0 \in \hat{\delta}(q_0, v)$ und $q_0 \in \delta(q_0, a)$ für $a \in \{0, 1\}$ folgt $q_0 \in \hat{\delta}(q_0, w)$

b) 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 $q_0 \in \hat{\delta}(q_0, v)$ und $q_1 \in \delta(q_0, a)$ folgt $q_1 \in \hat{\delta}(q_0, w)$

c) 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 $q_1 \in \hat{\delta}(q_0, v)$ gelten. Wegen b) für v endet v mit 0, also w mit 01

Wenn umgekehrt w mit 01 endet, dann ist $a = 1$ und v endet mit 0.

Wegen $q_1 \in \hat{\delta}(q_0, v)$ nach b) 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 $\mathbf{K} = (\mathbf{q}, \mathbf{w}) \in Q \times \Sigma^*$

ARBEITSWEISE VON ϵ -NEAs – ALTERNATIVE BESCHREIBUNG MIT KONFIGURATIONSÜBERGÄNGEN

- Definiere **Konfigurationen**

- Formal dargestellt als Tupel $\mathbf{K} = (\mathbf{q}, \mathbf{w}) \in Q \times \Sigma^*$

- Definiere **Konfigurationsübergangsrelation** $\vdash^*$

- Wechsel zwischen Konfigurationen durch Abarbeitung von Wörtern

ARBEITSWEISE VON ϵ -NEAs – ALTERNATIVE BESCHREIBUNG MIT KONFIGURATIONSÜBERGÄNGEN

- Definiere **Konfigurationen**

- Formal dargestellt als Tupel $\mathbf{K} = (\mathbf{q}, \mathbf{w}) \in Q \times \Sigma^*$

- Definiere **Konfigurationsübergangsrelation $\vdash^*$**

- Wechsel zwischen Konfigurationen durch Abarbeitung von Wörtern
- $(\mathbf{q}, \mathbf{aw}) \vdash (\mathbf{p}, \mathbf{w})$, falls $p \in \delta(q, a)$
- $(\mathbf{q}, \mathbf{w}) \vdash (\mathbf{p}, \mathbf{w})$, falls $p \in \delta(q, \epsilon)$

ARBEITSWEISE VON ϵ -NEAS – ALTERNATIVE BESCHREIBUNG MIT KONFIGURATIONSÜBERGÄNGEN

- Definiere **Konfigurationen**

- Formal dargestellt als Tupel $\mathbf{K} = (\mathbf{q}, \mathbf{w}) \in Q \times \Sigma^*$

- Definiere **Konfigurationsübergangsrelation $\vdash^*$**

- Wechsel zwischen Konfigurationen durch Abarbeitung von Wörtern
- $(\mathbf{q}, \mathbf{aw}) \vdash (\mathbf{p}, \mathbf{w})$, falls $p \in \delta(q, a)$
- $(\mathbf{q}, \mathbf{w}) \vdash (\mathbf{p}, \mathbf{w})$, falls $p \in \delta(q, \epsilon)$
- $\mathbf{K}_1 \vdash^* \mathbf{K}_2$, falls $K_1 = K_2$ oder
es gibt eine Konfiguration K mit $K_1 \vdash K$ und $K \vdash^* K_2$

ARBEITSWEISE VON ϵ -NEAS – ALTERNATIVE BESCHREIBUNG MIT KONFIGURATIONSÜBERGÄNGEN

- Definiere **Konfigurationen**

- Formal dargestellt als Tupel $\mathbf{K} = (\mathbf{q}, \mathbf{w}) \in Q \times \Sigma^*$

- Definiere **Konfigurationsübergangsrelation $\vdash^*$**

- Wechsel zwischen Konfigurationen durch Abarbeitung von Wörtern

- $(\mathbf{q}, \mathbf{aw}) \vdash (\mathbf{p}, \mathbf{w})$, falls $p \in \delta(q, a)$

- $(\mathbf{q}, \mathbf{w}) \vdash (\mathbf{p}, \mathbf{w})$, falls $p \in \delta(q, \epsilon)$

- $\mathbf{K}_1 \vdash^* \mathbf{K}_2$, falls $K_1 = K_2$ oder

es gibt eine Konfiguration K mit $K_1 \vdash K$ und $K \vdash^* K_2$

- **Akzeptierte Sprache**

- Menge der Eingaben, für die $\vdash^*$ zu akzeptierenden Zustand führt

ARBEITSWEISE VON ϵ -NEAS – ALTERNATIVE BESCHREIBUNG MIT KONFIGURATIONSÜBERGÄNGEN

- Definiere **Konfigurationen**

- Formal dargestellt als Tupel $\mathbf{K} = (\mathbf{q}, \mathbf{w}) \in Q \times \Sigma^*$

- Definiere **Konfigurationsübergangsrelation $\vdash^*$**

- Wechsel zwischen Konfigurationen durch Abarbeitung von Wörtern

- $(\mathbf{q}, \mathbf{aw}) \vdash (\mathbf{p}, \mathbf{w})$, falls $p \in \delta(q, a)$

- $(\mathbf{q}, \mathbf{w}) \vdash (\mathbf{p}, \mathbf{w})$, falls $p \in \delta(q, \epsilon)$

- $\mathbf{K}_1 \vdash^* \mathbf{K}_2$, falls $K_1 = K_2$ oder

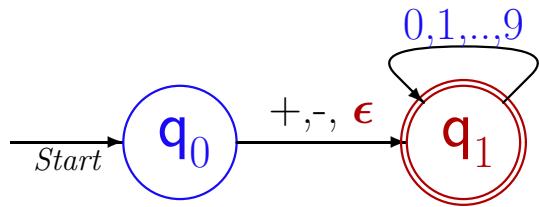
es gibt eine Konfiguration K mit $K_1 \vdash K$ und $K \vdash^* K_2$

- **Akzeptierte Sprache**

- Menge der Eingaben, für die $\vdash^*$ zu akzeptierenden Zustand führt

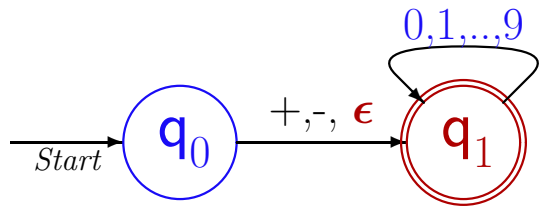
$$\mathbf{L}(\mathbf{A}) = \{\mathbf{w} \in \Sigma^* \mid \exists p \in F. (q_0, \mathbf{w}) \vdash^* (\mathbf{p}, \epsilon)\}$$

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*.$$
$$w = u \vee w = +u \vee w = -u\}$$

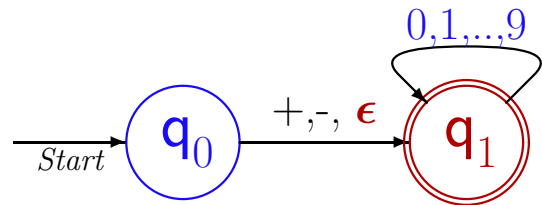
NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

- Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)

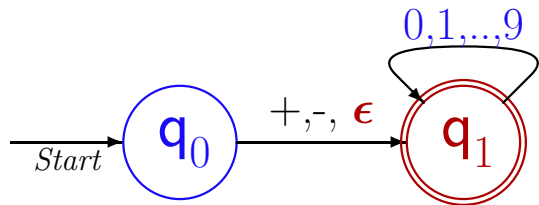


$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

- Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$
- **Basisfall $w = \epsilon$:**
 - Per Definition gilt $(q_1, v) \vdash^* (q_1, v)$ und $\epsilon \in \{0, \dots, 9\}^*$

✓

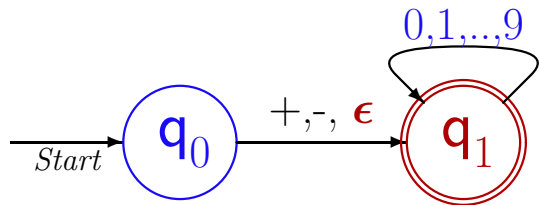
NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

- Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$
- **Basisfall** $w = \epsilon$:
 - Per Definition gilt $(q_1, v) \vdash^* (q_1, v)$ und $\epsilon \in \{0, \dots, 9\}^*$ ✓
- **Schrittfall** $w = ua$ für ein $u \in \Sigma^*, a \in \Sigma$:

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

• Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$

• **Basisfall** $w = \epsilon$:

– Per Definition gilt $(q_1, v) \vdash^* (q_1, v)$ und $\epsilon \in \{0, \dots, 9\}^*$

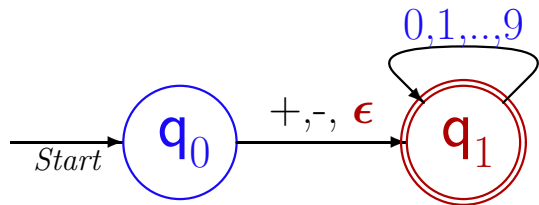
✓

• **Schrittfall** $w = ua$ für ein $u \in \Sigma^*, a \in \Sigma$:

$\Rightarrow$: Es gelte $(q_1, wv) \vdash^* (q_1, v)$.

Dann gilt $(q_1, uav) \vdash^* (p, av) \vdash (q_1, v)$ für einen Zustand p .

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

- Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$

- **Basisfall $w = \epsilon$:**

– Per Definition gilt $(q_1, v) \vdash^* (q_1, v)$ und $\epsilon \in \{0, \dots, 9\}^*$

✓

- **Schrittfall $w = ua$ für ein $u \in \Sigma^*, a \in \Sigma$:**

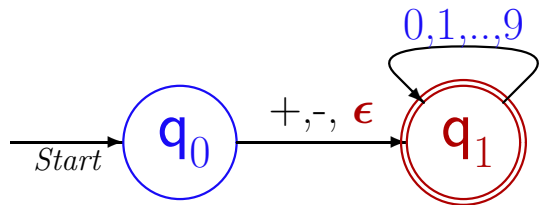
$\Rightarrow$: Es gelte $(q_1, wv) \vdash^* (q_1, v)$.

Dann gilt $(q_1, uav) \vdash^* (p, av) \vdash (q_1, v)$ für einen Zustand p .

Es folgt $p = q_1, a \in \{0, \dots, 9\}$ und per Induktion $w \in \{0, \dots, 9\}^*$

✓

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

• Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$

• **Basisfall** $w = \epsilon$:

– Per Definition gilt $(q_1, v) \vdash^* (q_1, v)$ und $\epsilon \in \{0, \dots, 9\}^*$

✓

• **Schrittfall** $w = ua$ für ein $u \in \Sigma^*, a \in \Sigma$:

$\Rightarrow$: Es gelte $(q_1, wv) \vdash^* (q_1, v)$.

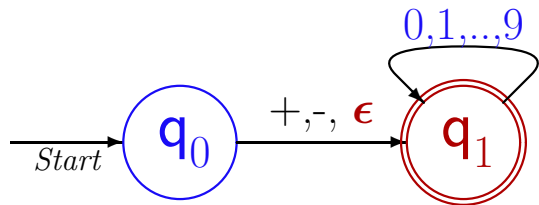
Dann gilt $(q_1, uav) \vdash^* (p, av) \vdash (q_1, v)$ für einen Zustand p .

Es folgt $p = q_1, a \in \{0, \dots, 9\}$ und per Induktion $w \in \{0, \dots, 9\}^*$

✓

$\Leftarrow$: Es sei $w \in \{0, \dots, 9\}^*$. Dann ist $u \in \{0, \dots, 9\}^*$ und $a \in \{0, \dots, 9\}$.

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

• Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$

• **Basisfall** $w = \epsilon$:

– Per Definition gilt $(q_1, v) \vdash^* (q_1, v)$ und $\epsilon \in \{0, \dots, 9\}^*$ ✓

• **Schrittfall** $w = ua$ für ein $u \in \Sigma^*, a \in \Sigma$:

$\Rightarrow$: Es gelte $(q_1, wv) \vdash^* (q_1, v)$.

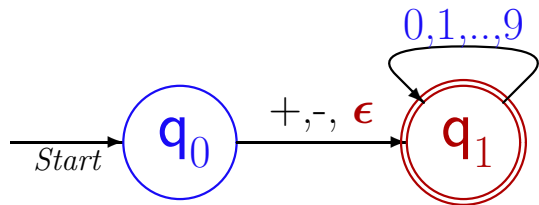
Dann gilt $(q_1, uav) \vdash^* (p, av) \vdash (q_1, v)$ für einen Zustand p .

Es folgt $p = q_1, a \in \{0, \dots, 9\}$ und per Induktion $w \in \{0, \dots, 9\}^*$ ✓

$\Leftarrow$: Es sei $w \in \{0, \dots, 9\}^*$. Dann ist $u \in \{0, \dots, 9\}^*$ und $a \in \{0, \dots, 9\}$.

Mit der Induktionsannahme folgt $(q_1, uav) \vdash^* (q_1, av) \vdash (q_1, v)$ ✓

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

- Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$

- **Basisfall** $w = \epsilon$:

– Per Definition gilt $(q_1, v) \vdash^* (q_1, v)$ und $\epsilon \in \{0, \dots, 9\}^*$ ✓

- **Schrittfall** $w = ua$ für ein $u \in \Sigma^*, a \in \Sigma$:

$\Rightarrow$: Es gelte $(q_1, wv) \vdash^* (q_1, v)$.

Dann gilt $(q_1, uav) \vdash^* (p, av) \vdash (q_1, v)$ für einen Zustand p .

Es folgt $p = q_1, a \in \{0, \dots, 9\}$ und per Induktion $w \in \{0, \dots, 9\}^*$ ✓

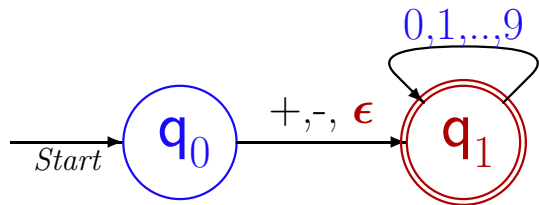
$\Leftarrow$: Es sei $w \in \{0, \dots, 9\}^*$. Dann ist $u \in \{0, \dots, 9\}^*$ und $a \in \{0, \dots, 9\}$.

Mit der Induktionsannahme folgt $(q_1, uav) \vdash^* (q_1, av) \vdash (q_1, v)$ ✓

- Es folgt

$$w \in L(A) \Leftrightarrow (q_0, w) \vdash^* (q_1, \epsilon)$$

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

- Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$

- **Basisfall $w = \epsilon$:**

– Per Definition gilt $(q_1, v) \vdash^* (q_1, v)$ und $\epsilon \in \{0, \dots, 9\}^*$ ✓

- **Schrittfall $w = ua$ für ein $u \in \Sigma^*, a \in \Sigma$:**

$\Rightarrow$: Es gelte $(q_1, wv) \vdash^* (q_1, v)$.

Dann gilt $(q_1, uav) \vdash^* (p, av) \vdash (q_1, v)$ für einen Zustand p .

Es folgt $p = q_1, a \in \{0, \dots, 9\}$ und per Induktion $w \in \{0, \dots, 9\}^*$ ✓

$\Leftarrow$: Es sei $w \in \{0, \dots, 9\}^*$. Dann ist $u \in \{0, \dots, 9\}^*$ und $a \in \{0, \dots, 9\}$.

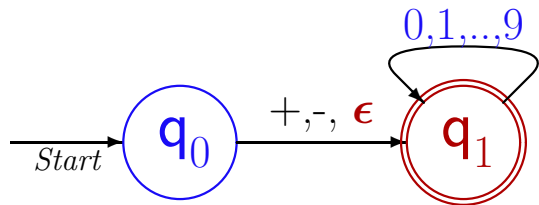
Mit der Induktionsannahme folgt $(q_1, uav) \vdash^* (q_1, av) \vdash (q_1, v)$ ✓

- **Es folgt**

$$w \in L(A) \Leftrightarrow (q_0, w) \vdash^* (q_1, \epsilon)$$

$$\Leftrightarrow \exists u \in \Sigma^*. w \in \{u, +u, -u\}. (q_0, w) \vdash (q_1, u) \vdash^* (q_1, \epsilon)$$

NACHWEIS DER ERKANNTEN SPRACHE (MIT ϵ -ÜBERGÄNGEN)



$$L(A) = \{w \in \Sigma^* \mid \exists u \in \{0, \dots, 9\}^*. \\ w = u \vee w = +u \vee w = -u\}$$

- Zeige $(q_1, wv) \vdash^* (q_1, v) \Leftrightarrow w \in \{0, \dots, 9\}^*$ für alle $w, v \in \Sigma^*$

- **Basisfall $w = \epsilon$:**

– Per Definition gilt $(q_1, v) \vdash^* (q_1, v)$ und $\epsilon \in \{0, \dots, 9\}^*$ ✓

- **Schrittfall $w = ua$ für ein $u \in \Sigma^*, a \in \Sigma$:**

$\Rightarrow$: Es gelte $(q_1, wv) \vdash^* (q_1, v)$.

Dann gilt $(q_1, uav) \vdash^* (p, av) \vdash (q_1, v)$ für einen Zustand p .

Es folgt $p = q_1, a \in \{0, \dots, 9\}$ und per Induktion $w \in \{0, \dots, 9\}^*$ ✓

$\Leftarrow$: Es sei $w \in \{0, \dots, 9\}^*$. Dann ist $u \in \{0, \dots, 9\}^*$ und $a \in \{0, \dots, 9\}$.

Mit der Induktionsannahme folgt $(q_1, uav) \vdash^* (q_1, av) \vdash (q_1, v)$ ✓

- **Es folgt**

$$w \in L(A) \Leftrightarrow (q_0, w) \vdash^* (q_1, \epsilon)$$

$$\Leftrightarrow \exists u \in \Sigma^*. w \in \{u, +u, -u\}. (q_0, w) \vdash (q_1, u) \vdash^* (q_1, \epsilon)$$

$$\Leftrightarrow \exists u \in \{0, \dots, 9\}^*. w = u \vee w = +u \vee w = -u \Leftrightarrow w \in L$$

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 Zustandsüberföhrungsfunktionen 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 Zustandsüberföhrungsfunktionen 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
 - $Q_D = \mathcal{P}(Q_N)$
 - $q_D = \epsilon\text{-Hölle}(q_0)$ (= $\{q_0\}$ bei NEAs)
 - $F_D = \{S \in Q_D \mid S \cap F_N \neq \emptyset\}$
 - $\delta_D(S, a) = \bigcup_{q \in S} \hat{\delta}_N(q, a) = \{p \mid \exists q \in S. p \in \hat{\delta}_N(q, a)\}$ (erfaßt ϵ -Hölle)

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 Zustandsüberföhrungsfunktionen 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
 - $Q_D = \mathcal{P}(Q_N)$
 - $q_D = \epsilon\text{-Hölle}(q_0)$ (= $\{q_0\}$ bei NEAs)
 - $F_D = \{S \in Q_D \mid S \cap F_N \neq \emptyset\}$
 - $\delta_D(S, a) = \bigcup_{q \in S} \hat{\delta}_N(q, a) = \{p \mid \exists q \in S. p \in \hat{\delta}_N(q, a)\}$ (erfaßt ϵ -Hölle)
 - Dann gilt $L(A_D) = L(A_N)$

BEZIEHUNG ZU DETERMINISTISCHEN AUTOMATEN

- **Nichtdeterministische Automaten sind flexibler**

- Man muss sich nicht auf eine genaue Verarbeitungsfolge festlegen
- Man kann optionale Eingaben elegant verarbeiten

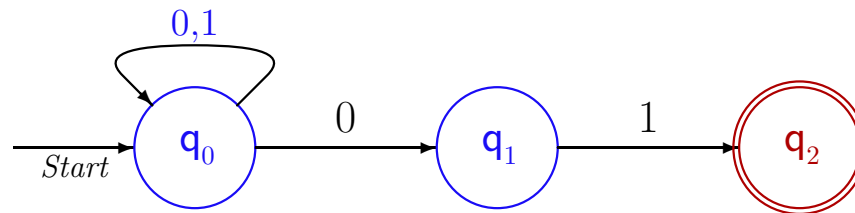
- **DEAs sind genauso ausdrucksstark wie ϵ -NEAs**

- Man kann Mengen von ϵ -NEA-Zuständen als DEA Zustände codieren
- Man kann mengenwertige Zustandsüberföhrungsfunktionen codieren

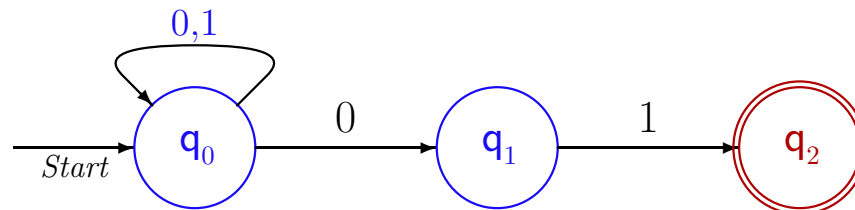
- **(Potenzmengen- oder) Teilmengenkonstruktion**

- 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
 - $Q_D = \mathcal{P}(Q_N)$
 - $q_D = \epsilon\text{-Hölle}(q_0)$ ($= \{q_0\}$ bei NEAs)
 - $F_D = \{S \in Q_D \mid S \cap F_N \neq \emptyset\}$
 - $\delta_D(S, a) = \bigcup_{q \in S} \hat{\delta}_N(q, a) = \{p \mid \exists q \in S. p \in \hat{\delta}_N(q, a)\}$ (erfaßt ϵ -Hölle)
- Dann gilt $L(A_D) = L(A_N)$
- Konstruktion benötigt $2^{|Q_N|}$ Zustände (Optimierung möglich)

TEILMENGENKONSTRUKTION AM BEISPIEL



TEILMENGENKONSTRUKTION AM BEISPIEL



Konstruierter deterministischer Automat

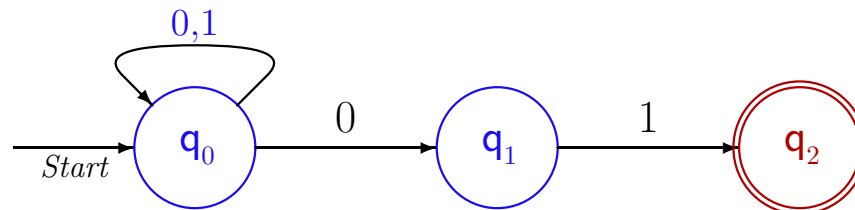
$$Q_D = \mathcal{P}(\{q_0, q_1, q_2\})$$

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

$$F_D = \{S \subseteq \{q_0, q_1, q_2\} \mid q_2 \in S\}$$

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

TEILMENGENKONSTRUKTION AM BEISPIEL



Konstruierter deterministischer Automat

$$Q_D = \mathcal{P}(\{q_0, q_1, q_2\})$$

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

$$F_D = \{S \subseteq \{q_0, q_1, q_2\} \mid q_2 \in S\}$$

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

Viele überflüssige Zustände (nur drei von $\{q_0\}$ erreichbar)

- **Optimierung:** $Q_D \hat{=}$ erreichbare Zustände
 - Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
 - Konstruiere Zustandsmenge Q_D iterativ gleichzeitig mit δ_D

● **Optimierung:** $Q_D \hat{=}$ erreichbare Zustände

- Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
- Konstruiere Zustandsmenge Q_D iterativ gleichzeitig mit δ_D
- Start: $Q_0 := \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\}$
- Schritt: $Q_{i+1} := Q_i \cup \{\delta_D(S, a) \mid S \in Q_i, a \in \Sigma\}$

Dabei konstruiere die nötigen Werte $\delta_D(S, a) = \bigcup_{q \in S} \hat{\delta}_N(q, a)$

● **Optimierung:** $Q_D \hat{=}$ erreichbare Zustände

- Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
- Konstruiere Zustandsmenge Q_D iterativ gleichzeitig mit δ_D
- Start: $Q_0 := \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\}$
- Schritt: $Q_{i+1} := Q_i \cup \{\delta_D(S, a) \mid S \in Q_i, a \in \Sigma\}$

Dabei konstruiere die nötigen Werte $\delta_D(S, a) = \bigcup_{q \in S} \hat{\delta}_N(q, a)$

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

● **Optimierung:** $Q_D \hat{=}$ erreichbare Zustände

- Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
- Konstruiere Zustandsmenge Q_D iterativ gleichzeitig mit δ_D
- Start: $Q_0 := \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\}$
- Schritt: $Q_{i+1} := Q_i \cup \{\delta_D(S, a) \mid S \in Q_i, a \in \Sigma\}$

Dabei konstruiere die nötigen Werte $\delta_D(S, a) = \bigcup_{q \in S} \hat{\delta}_N(q, a)$

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

● **Optimierung:** $Q_D \hat{=}$ erreichbare Zustände

- Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
- Konstruiere Zustandsmenge Q_D iterativ gleichzeitig mit δ_D
- Start: $Q_0 := \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\}$
- Schritt: $Q_{i+1} := Q_i \cup \{\delta_D(S, a) \mid S \in Q_i, a \in \Sigma\}$

Dabei konstruiere die nötigen Werte $\delta_D(S, a) = \bigcup_{q \in S} \hat{\delta}_N(q, a)$

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

● **Optimierung:** $Q_D \hat{=}$ erreichbare Zustände

- Sei $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ ein nichtdeterministischer Automat
- Konstruiere Zustandsmenge Q_D iterativ gleichzeitig mit δ_D
- Start: $Q_0 := \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\}$
- Schritt: $Q_{i+1} := Q_i \cup \{\delta_D(S, a) \mid S \in Q_i, a \in \Sigma\}$

Dabei konstruiere die nötigen Werte $\delta_D(S, a) = \bigcup_{q \in S} \hat{\delta}_N(q, a)$

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

● **ϵ -NEAs und DEAs akzeptieren dieselben Sprachen**

- Jeder DEA ist als “eindeutiger” ϵ -NEA beschreibbar

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

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 $w \in \Sigma^*$

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^*

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 $w = \epsilon$:

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

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 $w = \epsilon$:

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

– Induktionsschritt: Sei $w = va$ für ein $v \in \Sigma^*$ und $a \in \Sigma$:

– Induktionsannahme: Es gelte $\hat{\delta}_D(q_D, v) = \hat{\delta}_N(q_0, v)$

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 $w = \epsilon$:

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

– **Induktionsschritt:** Sei $w = va$ für ein $v \in \Sigma^*$ und $a \in \Sigma$:

– **Induktionsannahme:** Es gelte $\hat{\delta}_D(q_D, v) = \hat{\delta}_N(q_0, v)$

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

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 $w = \epsilon$:

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

– **Induktionsschritt:** Sei $w = va$ für ein $v \in \Sigma^*$ und $a \in \Sigma$:

– **Induktionsannahme:** Es gelte $\hat{\delta}_D(q_D, v) = \hat{\delta}_N(q_0, v)$

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

$$= \delta_D(\hat{\delta}_D(q_D, v), a) \quad (\text{Definition } \hat{\delta}_D)$$

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 $w = \epsilon$:

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

– **Induktionsschritt:** Sei $w = va$ für ein $v \in \Sigma^*$ und $a \in \Sigma$:

– **Induktionsannahme:** Es gelte $\hat{\delta}_D(q_D, v) = \hat{\delta}_N(q_0, v)$

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

$$= \delta_D(\hat{\delta}_D(q_D, v), a)$$

(Definition $\hat{\delta}_D$)

$$= \delta_D(\hat{\delta}_N(q_0, v), a)$$

(Induktionsannahme)

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 $w = \epsilon$:

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

– **Induktionsschritt:** Sei $w = va$ für ein $v \in \Sigma^*$ und $a \in \Sigma$:

– **Induktionsannahme:** Es gelte $\hat{\delta}_D(q_D, v) = \hat{\delta}_N(q_0, v)$

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

$$= \delta_D(\hat{\delta}_D(q_D, v), a) \quad (\text{Definition } \hat{\delta}_D)$$

$$= \delta_D(\hat{\delta}_N(q_0, v), a) \quad (\text{Induktionsannahme})$$

$$= \bigcup_{q' \in \hat{\delta}_N(q_0, v)} \hat{\delta}_N(q', a) \quad (\text{Konstruktion von } \delta_D)$$

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 $w = \epsilon$:

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

– **Induktionsschritt:** Sei $w = va$ für ein $v \in \Sigma^*$ und $a \in \Sigma$:

– **Induktionsannahme:** Es gelte $\hat{\delta}_D(q_D, v) = \hat{\delta}_N(q_0, v)$

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

$$= \delta_D(\hat{\delta}_D(q_D, v), a) \quad (\text{Definition } \hat{\delta}_D)$$

$$= \delta_D(\hat{\delta}_N(q_0, v), a) \quad (\text{Induktionsannahme})$$

$$= \bigcup_{q' \in \hat{\delta}_N(q_0, v)} \hat{\delta}_N(q', a) \quad (\text{Konstruktion von } \delta_D)$$

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

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 $w = \epsilon$:

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

– **Induktionsschritt:** Sei $w = va$ für ein $v \in \Sigma^*$ und $a \in \Sigma$:

– **Induktionsannahme:** Es gelte $\hat{\delta}_D(q_D, v) = \hat{\delta}_N(q_0, v)$

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

$$= \delta_D(\hat{\delta}_D(q_D, v), a) \quad (\text{Definition } \hat{\delta}_D)$$

$$= \delta_D(\hat{\delta}_N(q_0, v), a) \quad (\text{Induktionsannahme})$$

$$= \bigcup_{q' \in \hat{\delta}_N(q_0, v)} \hat{\delta}_N(q', a) \quad (\text{Konstruktion von } \delta_D)$$

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

$$= \hat{\delta}_N(q_0, w) \quad (\text{Definition } \hat{\delta}_N)$$

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 $w = \epsilon$:

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

– **Induktionsschritt:** Sei $w = va$ für ein $v \in \Sigma^*$ und $a \in \Sigma$:

– **Induktionsannahme:** Es gelte $\hat{\delta}_D(q_D, v) = \hat{\delta}_N(q_0, v)$

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

$$= \delta_D(\hat{\delta}_D(q_D, v), a) \quad (\text{Definition } \hat{\delta}_D)$$

$$= \delta_D(\hat{\delta}_N(q_0, v), a) \quad (\text{Induktionsannahme})$$

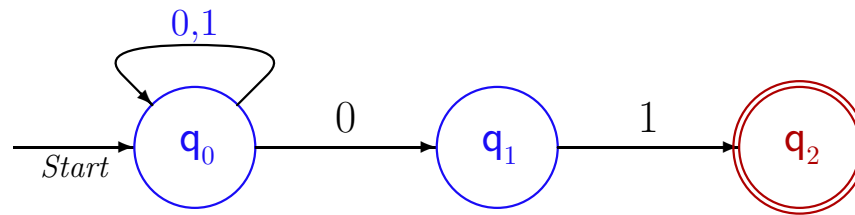
$$= \bigcup_{q' \in \hat{\delta}_N(q_0, v)} \delta_N(q', a) \quad (\text{Konstruktion von } \delta_D)$$

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

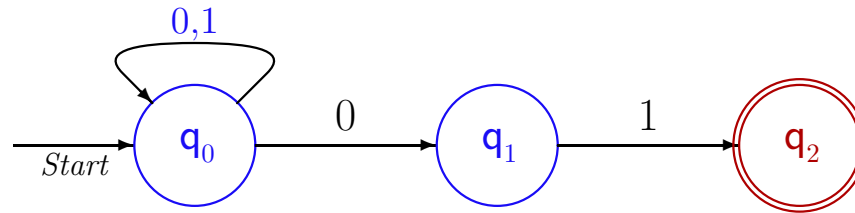
$$= \hat{\delta}_N(q_0, w) \quad (\text{Definition } \hat{\delta}_N)$$

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 NEAs



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR NEAs

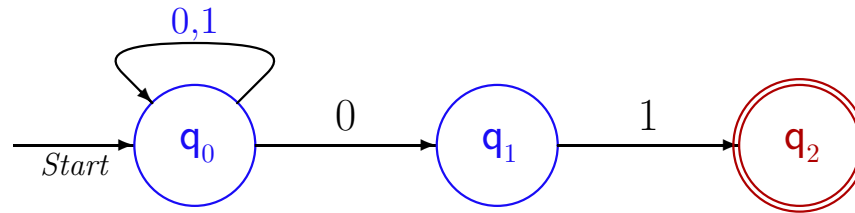


- Konstruktion von Zustandsmengen und (reduzierter) Überföhrungsfunktion

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

	0	1

OPTIMIERTE TEILMENGENKONSTRUKTION FÜR NEAs

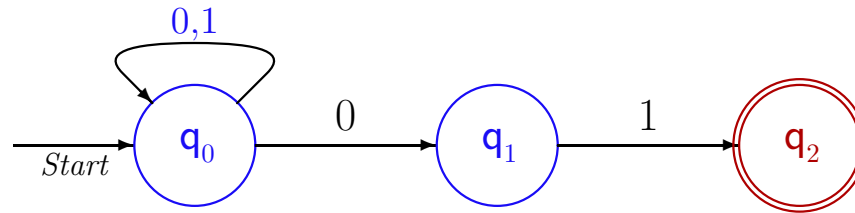


- Konstruktion von Zustandsmengen und (reduzierter) Überföhrungsfunktion

– $Q_0 := \{\{q_0\}\}$

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

OPTIMIERTE TEILMENGENKONSTRUKTION FÜR NEAs

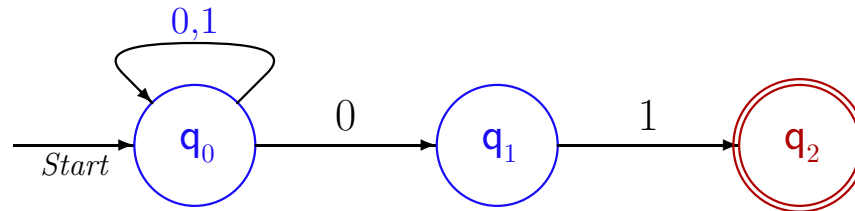


- Konstruktion von Zustandsmengen und (reduzierter) Überföhrungsfunktion

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

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

OPTIMIERTE TEILMENGENKONSTRUKTION FÜR NEAs

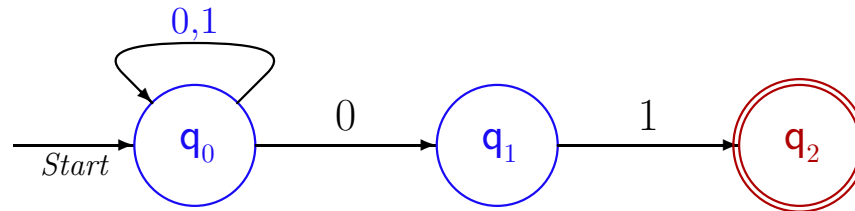


● Konstruktion von Zustandsmengen und (reduzierter) Überföhrungsfunktion

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

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

OPTIMIERTE TEILMENGENKONSTRUKTION FÜR NEAs

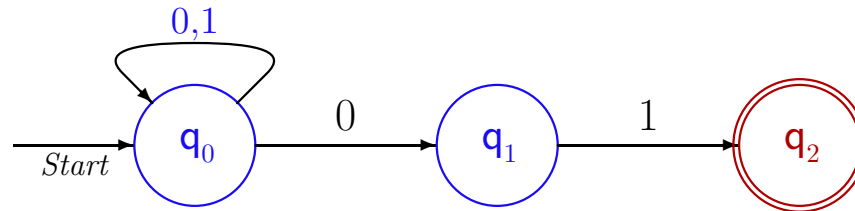


● Konstruktion von Zustandsmengen und (reduzierter) Überföhrungsfunktion

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

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

OPTIMIERTE TEILMENGENKONSTRUKTION FÜR NEAs

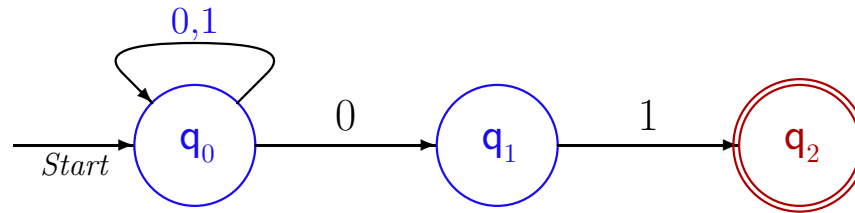


● Konstruktion von Zustandsmengen und (reduzierter) Überföhrungsfunktion

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

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

OPTIMIERTE TEILMENGENKONSTRUKTION FÜR NEAs

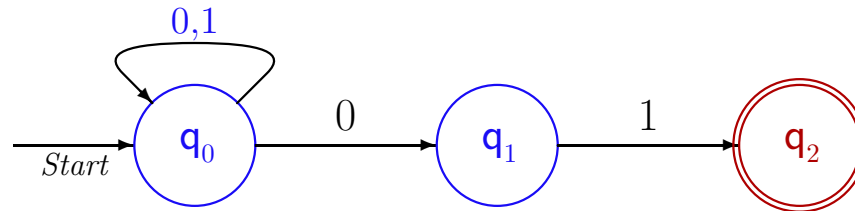


● Konstruktion von Zustandsmengen und (reduzierter) Überföhrungsfunktion

- $Q_0 := \{\{q_0\}\}$
- $Q_1 := \{\{q_0\}, \{q_0 q_1\}\}$
- $Q_2 := \{\{q_0\}, \{q_0 q_1\}, \{q_0 q_2\}\}$
- $Q_3 = Q_2$, also $Q_D = Q_2$,

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

OPTIMIERTE TEILMENGENKONSTRUKTION FÜR NEAs

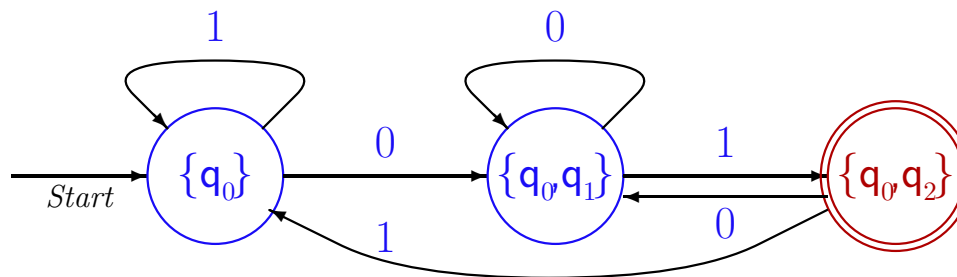


● Konstruktion von Zustandsmengen und (reduzierter) Überföhrungsfunktion

- $Q_0 := \{\{q_0\}\}$
- $Q_1 := \{\{q_0\}, \{q_0 q_1\}\}$
- $Q_2 := \{\{q_0\}, \{q_0 q_1\}, \{q_0 q_2\}\}$
- $Q_3 = Q_2$, also $Q_D = Q_2$,

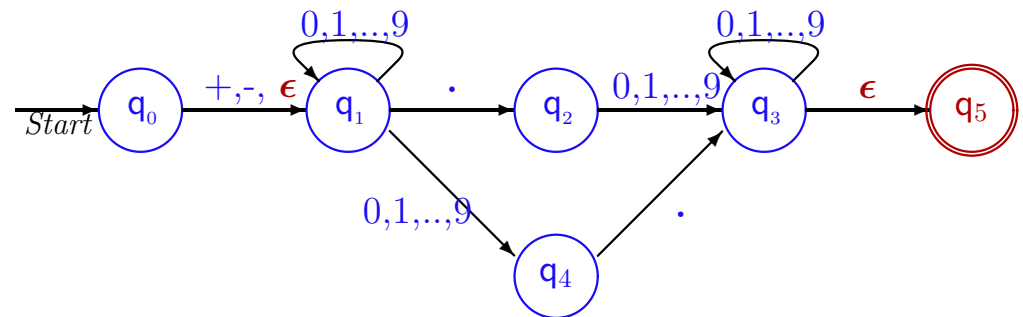
	0	1
$\rightarrow \{q_0\}$	$\{q_0 q_1\}$	$\{q_0\}$
$\{q_0 q_1\}$	$\{q_0 q_1\}$	$\{q_0 q_2\}$
$* \{q_0 q_2\}$	$\{q_0 q_1\}$	$\{q_0\}$

● Resultierender deterministischer Automat



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

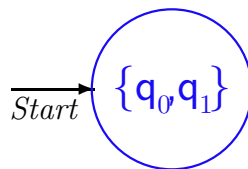
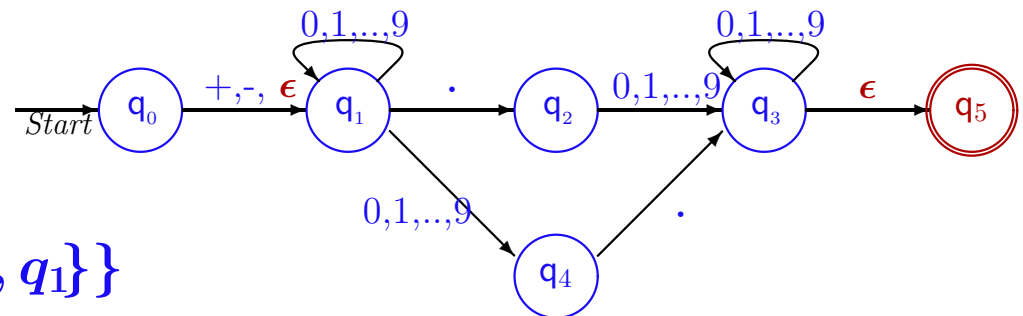
Konstruiere Q_D, δ_D



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

Konstruiere Q_D, δ_D

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

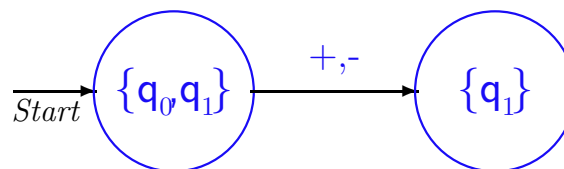
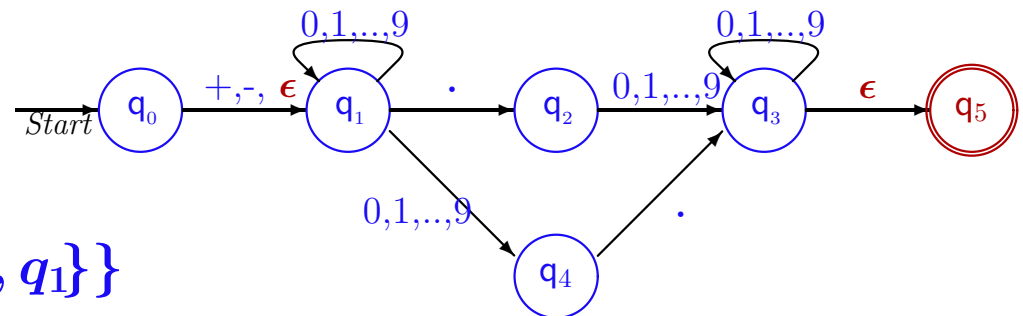


OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAS

Konstruiere Q_D, δ_D

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

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

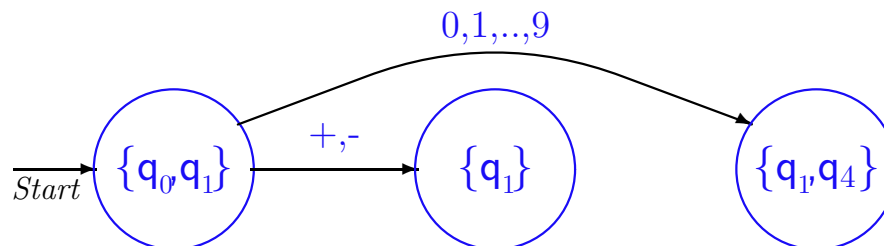
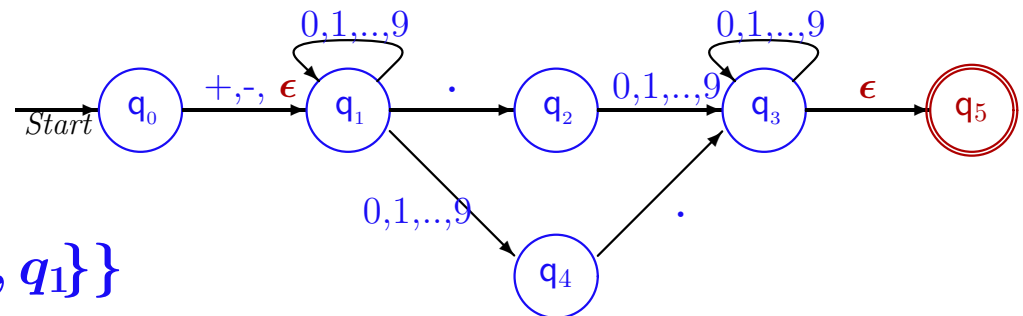


OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAS

Konstruiere Q_D, δ_D

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

- $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\delta_D(\{q_0, q_1\}, -) = \{q_1\}$
- $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \dots, \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$

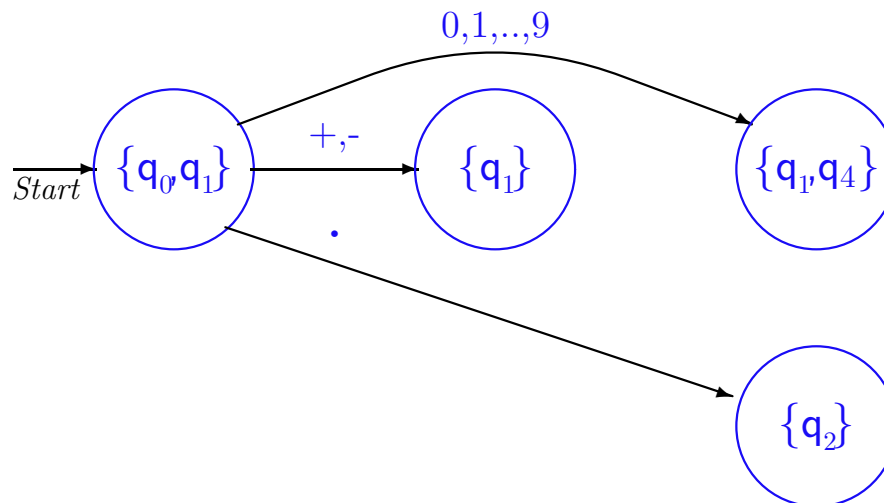
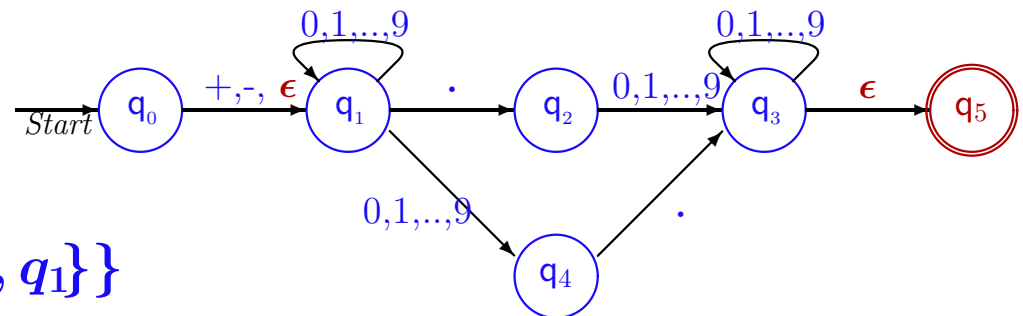


OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

Konstruiere Q_D, δ_D

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

- $\delta_D(\{q_0, q_1\}, +) = \{q_1\}$, $\delta_D(\{q_0, q_1\}, -) = \{q_1\}$
- $\delta_D(\{q_0, q_1\}, 0) = \{q_1, q_4\}, \dots, \delta_D(\{q_0, q_1\}, 9) = \{q_1, q_4\}$
- $\delta_D(\{q_0, q_1\}, \cdot) = \{q_2\}$

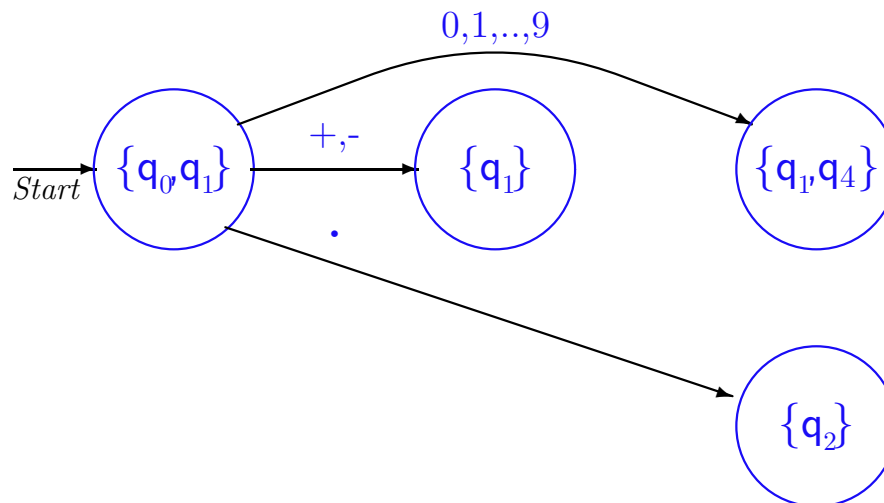
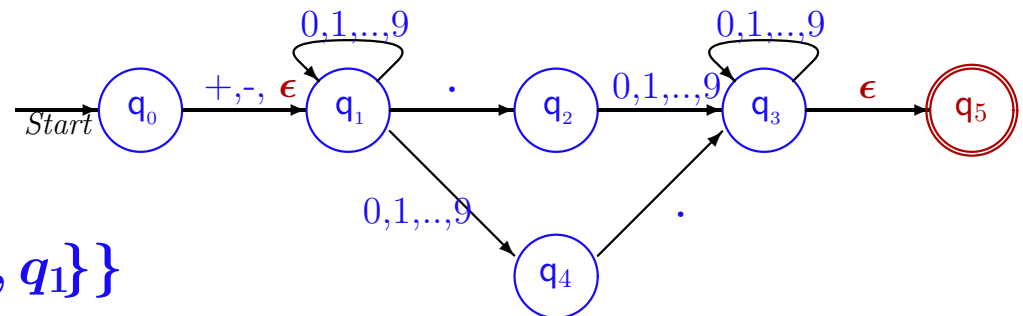


OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

Konstruiere Q_D, δ_D

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

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



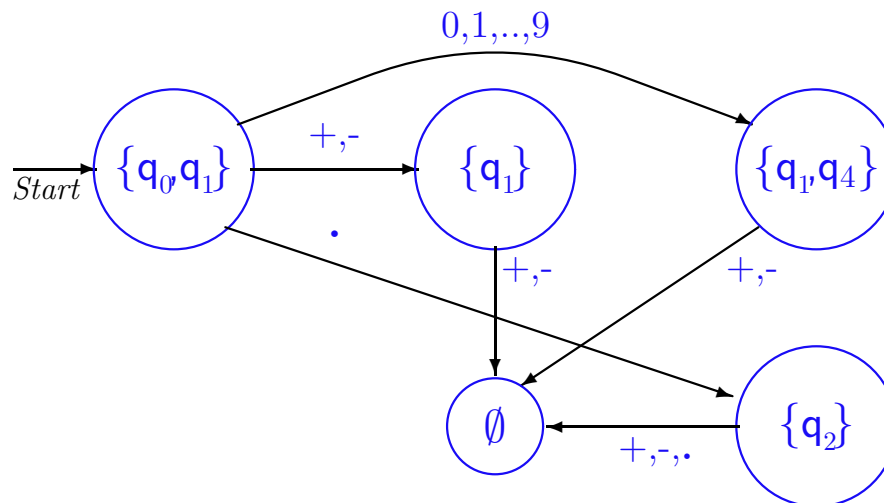
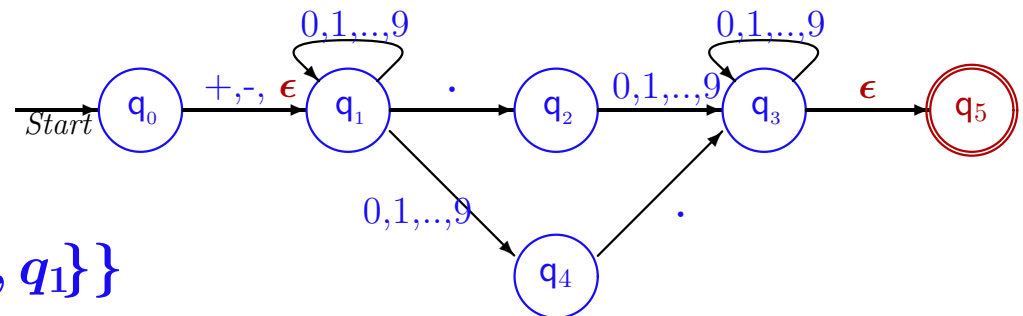
OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

Konstruiere Q_D, δ_D

$$Q_0 = \{q_D\} = \{\epsilon\text{-H\u00fclle}(q_0)\} = \{\{q_0, q_1\}\}$$

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

$$- \delta_D(\{q_1\}, +) = \delta_D(\{q_2\}, +) = \delta_D(\{q_1, q_4\}, +) = \emptyset, \dots \quad \delta_D(\{q_2\}, \cdot) = \emptyset$$



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

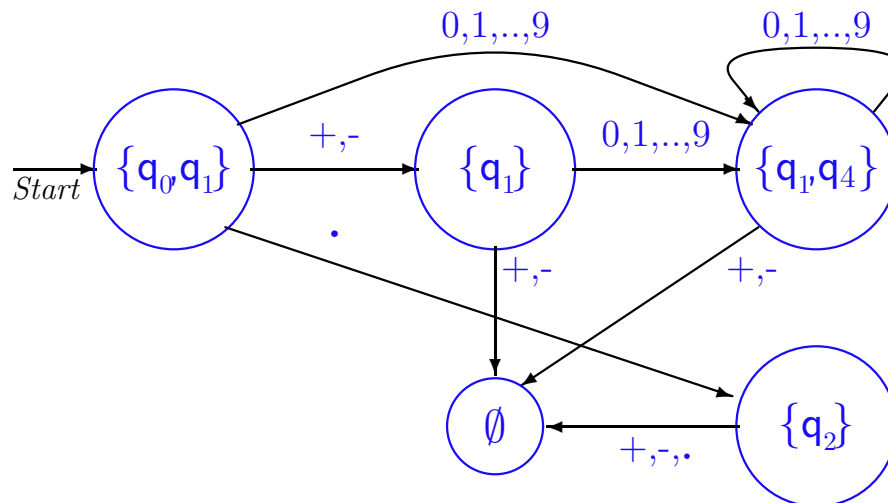
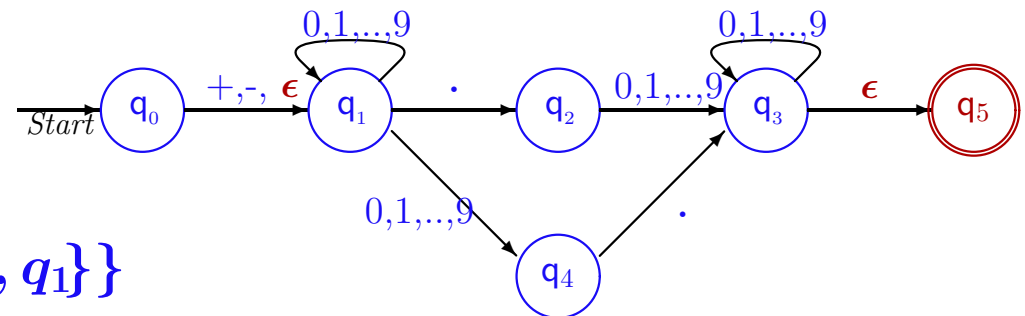
Konstruiere Q_D, δ_D

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

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

$$- \delta_D(\{q_1\}, +) = \delta_D(\{q_2\}, +) = \delta_D(\{q_1, q_4\}, +) = \emptyset, \dots \quad \delta_D(\{q_2\}, \cdot) = \emptyset$$

$$- \delta_D(\{q_1\}, 0) = \delta_D(\{q_1, q_4\}, 0) = \{q_1, q_4\}$$



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

Konstruiere Q_D, δ_D

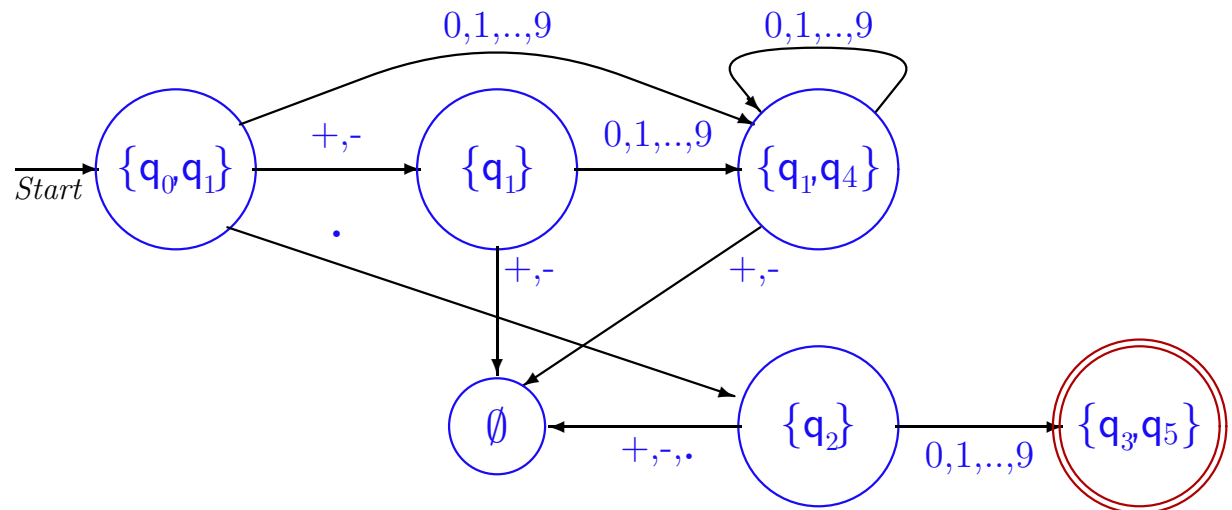
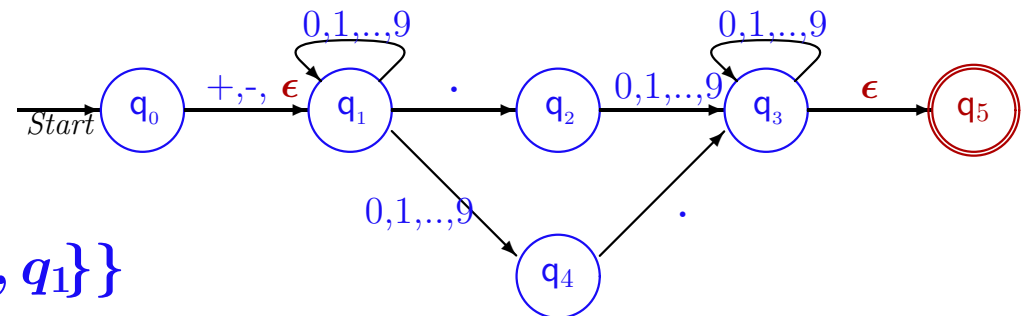
$$Q_0 = \{q_D\} = \{\epsilon\text{-H\u00fclle}(q_0)\} = \{\{q_0, q_1\}\}$$

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

$$- \delta_D(\{q_1\}, +) = \delta_D(\{q_2\}, +) = \delta_D(\{q_1, q_4\}, +) = \emptyset, \dots \quad \delta_D(\{q_2\}, \cdot) = \emptyset$$

$$- \delta_D(\{q_1\}, 0) = \delta_D(\{q_1, q_4\}, 0) = \{q_1, q_4\}$$

$$- \delta_D(\{q_2\}, 0) = \{q_3, q_5\}, \dots$$



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

Konstruiere Q_D, δ_D

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

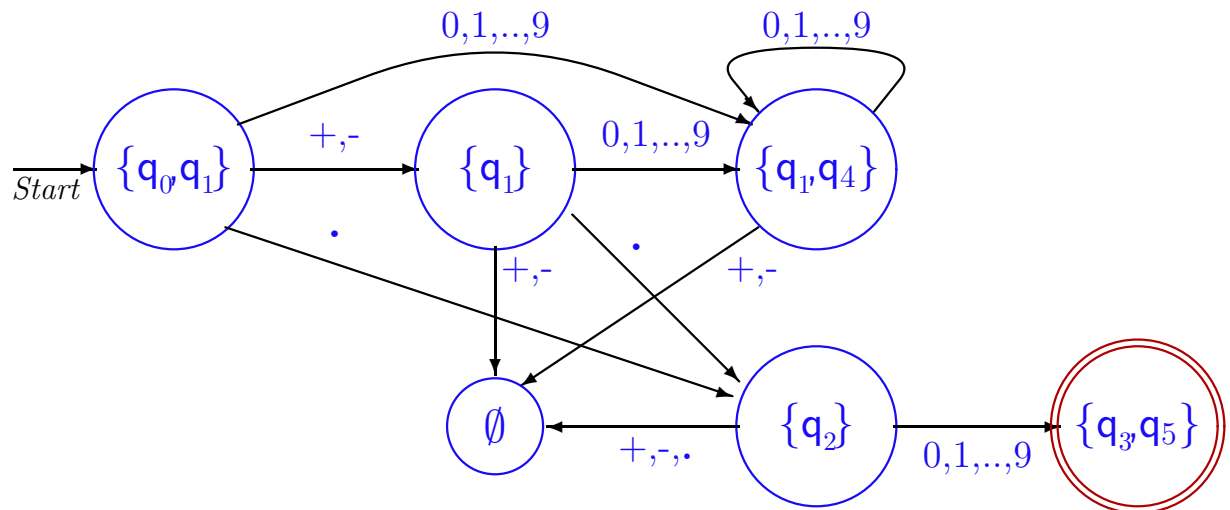
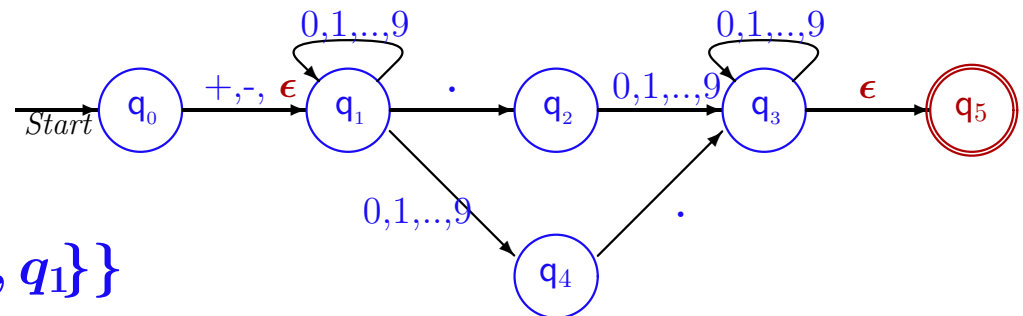
$$Q_1 = \{\{q_0, q_1\} \{q_1\}, \{q_1, q_4\}, \{q_2\}\}$$

$$- \delta_D(\{q_1\}, +) = \delta_D(\{q_2\}, +) = \delta_D(\{q_1, q_4\}, +) = \emptyset, \dots \quad \delta_D(\{q_2\}, \cdot) = \emptyset$$

$$- \delta_D(\{q_1\}, 0) = \delta_D(\{q_1, q_4\}, 0) = \{q_1, q_4\}$$

$$- \delta_D(\{q_2\}, 0) = \{q_3, q_5\}, \dots$$

$$- \delta_D(\{q_1\}, \cdot) = \{q_2\}$$



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

Konstruiere Q_D, δ_D

$$Q_0 = \{q_D\} = \{\epsilon\text{-H\u00fclle}(q_0)\} = \{\{q_0, q_1\}\}$$

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

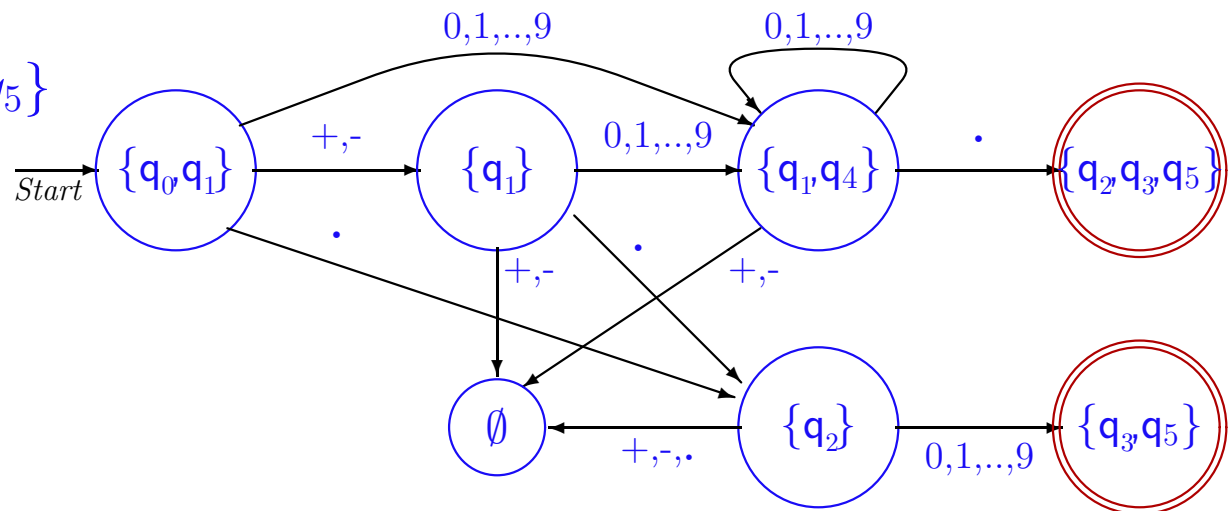
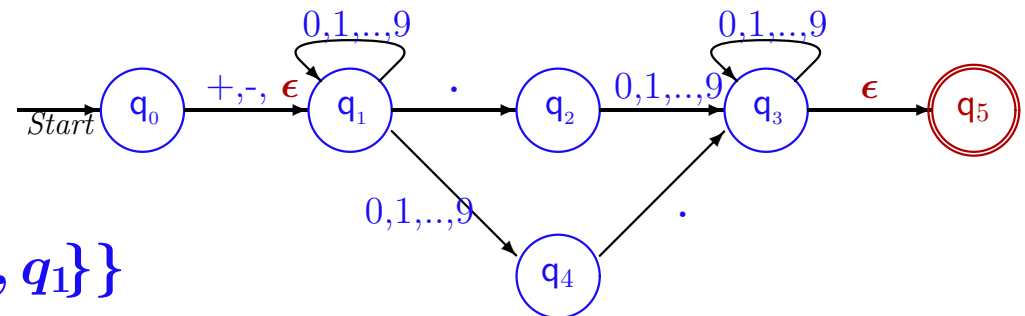
$$- \delta_D(\{q_1\}, +) = \delta_D(\{q_2\}, +) = \delta_D(\{q_1, q_4\}, +) = \emptyset, \dots \quad \delta_D(\{q_2\}, \cdot) = \emptyset$$

$$- \delta_D(\{q_1\}, 0) = \delta_D(\{q_1, q_4\}, 0) = \{q_1, q_4\}$$

$$- \delta_D(\{q_2\}, 0) = \{q_3, q_5\}, \dots$$

$$- \delta_D(\{q_1\}, \cdot) = \{q_2\}$$

$$- \delta_D(\{q_1, q_4\}, \cdot) = \{q_2, q_3, q_5\}$$



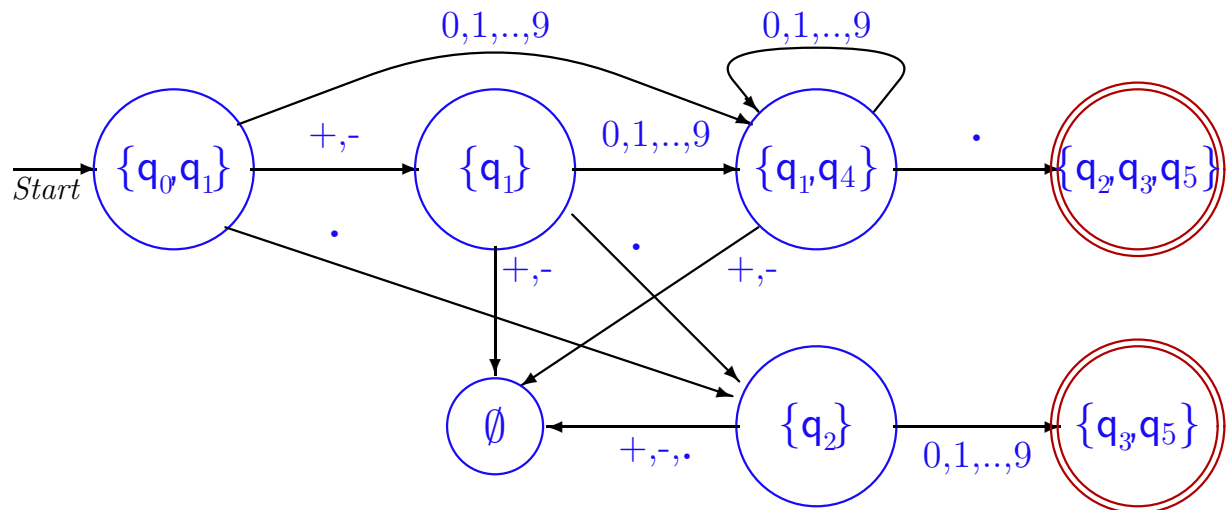
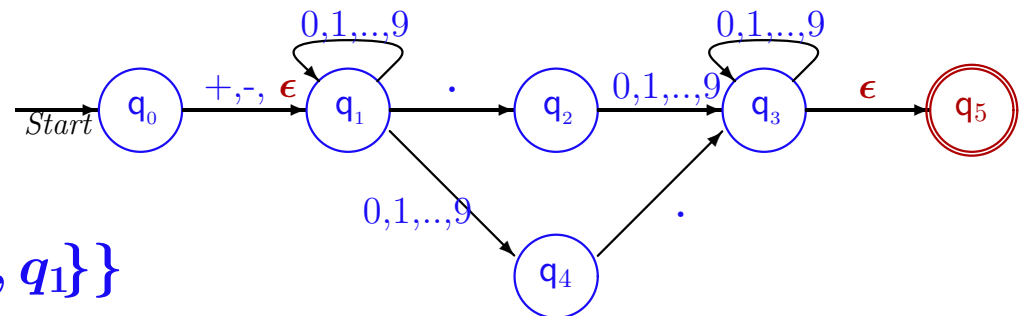
OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAS

Konstruiere Q_D, δ_D

$$Q_0 = \{q_D\} = \{\epsilon\text{-H\u00fclle}(q_0)\} = \{\{q_0, q_1\}\}$$

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

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



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

Konstruiere Q_D, δ_D

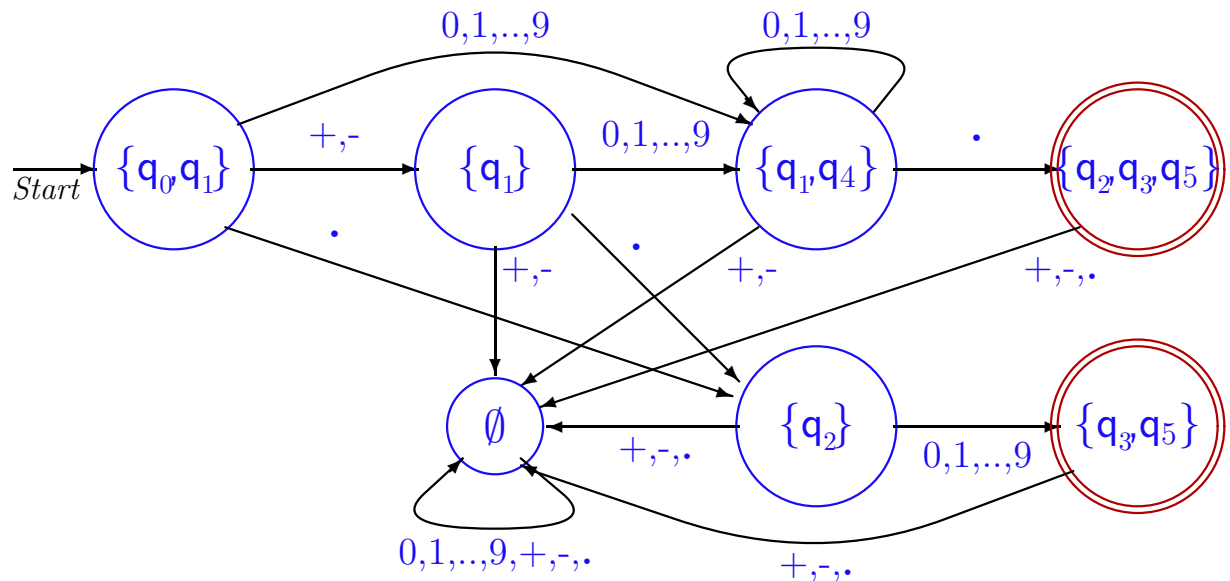
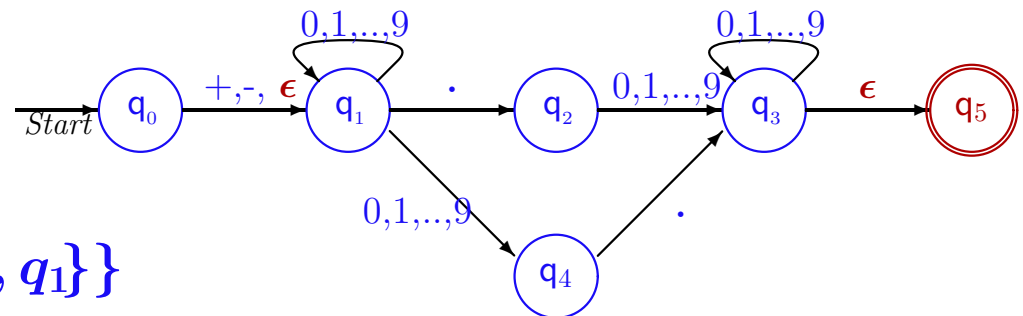
$$Q_0 = \{q_D\} = \{\epsilon\text{-Hülle}(q_0)\} = \{\{q_0, q_1\}\}$$

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

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

$$- \delta_D(\emptyset, +) = \delta_D(\{q_2, q_3, q_5\}, +) = \delta_D(\{q_3, q_5\}, +) = \emptyset, \dots$$

$$- \delta_D(\emptyset, \cdot) = \delta_D(\{q_2, q_3, q_5\}, \cdot) = \delta_D(\{q_3, q_5\}, \cdot) = \emptyset$$



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

Konstruiere Q_D , δ_D

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

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

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

$$- \delta_D(\emptyset, +) = \delta_D(\{q_2, q_3, q_5\}, +) = \delta_D(\{q_3, q_5\}, +) = \emptyset, \dots$$

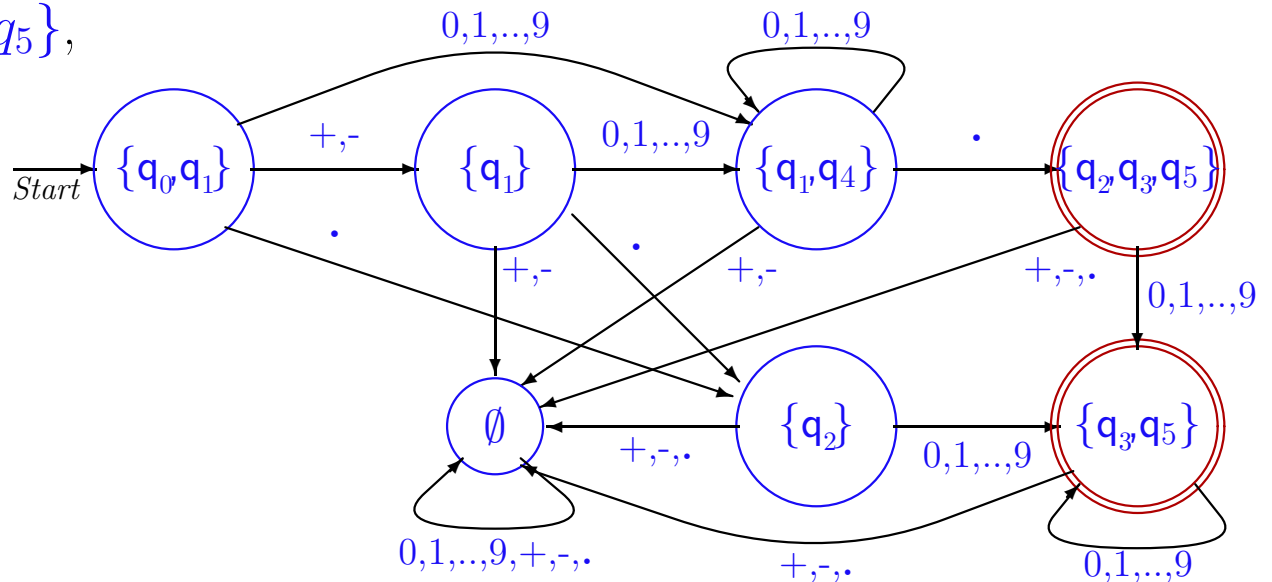
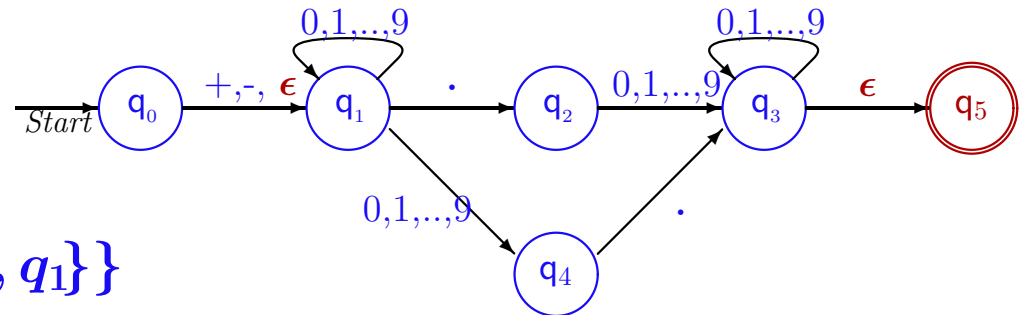
$$- \delta_D(\emptyset, \cdot) = \delta_D(\{q_2, q_3, q_5\}, \cdot) = \delta_D(\{q_3, q_5\}, \cdot) = \emptyset$$

$$- \delta_D(\{q_2, q_3, q_5\}, 0) = \{q_3, q_5\},$$

$$- \delta_D(\{q_3, q_5\}, 0) = \{q_3, q_5\}$$

$\vdots$

$$- \delta_D(\emptyset, 0) = \emptyset$$



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

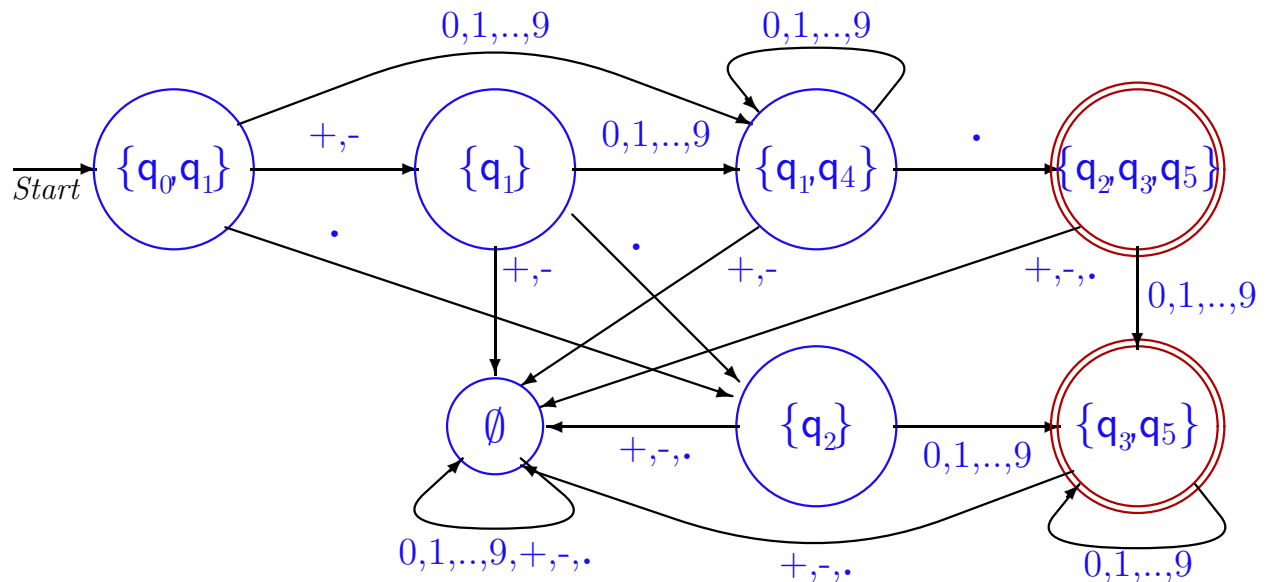
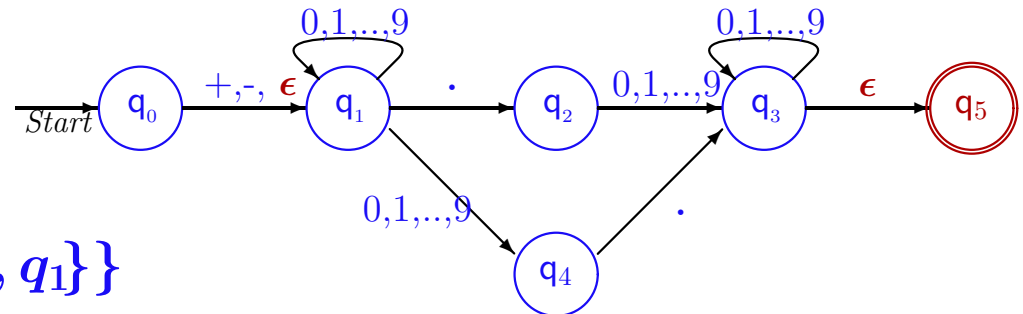
Konstruiere Q_D, δ_D

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

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

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

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



OPTIMIERTE TEILMENGENKONSTRUKTION FÜR ϵ -NEAs

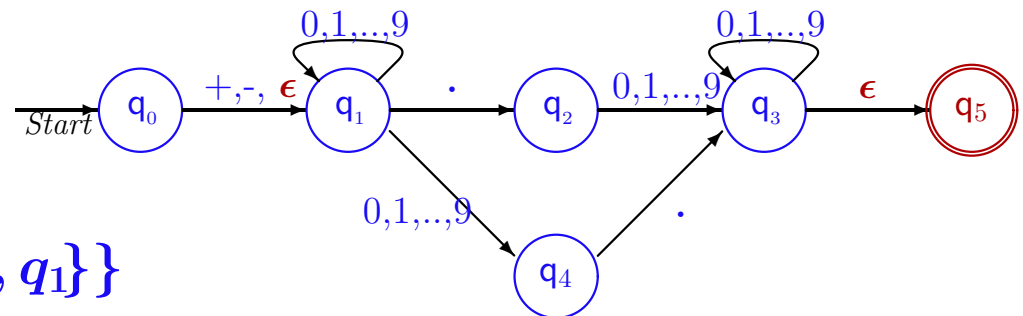
Konstruiere Q_D, δ_D

$$Q_0 = \{q_D\} = \{\epsilon\text{-H\u00fclle}(q_0)\} = \{\{q_0, q_1\}\}$$

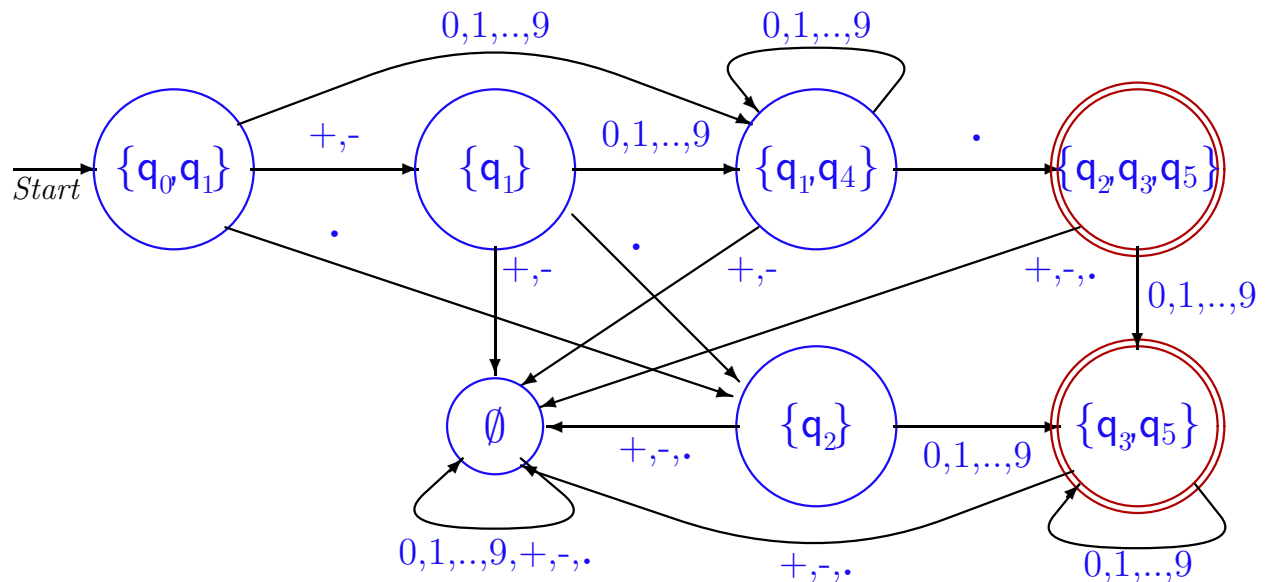
$$Q_1 = \{\{q_0, q_1\} \{q_1\}, \{q_1, q_4\}, \{q_2\}\}$$

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

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

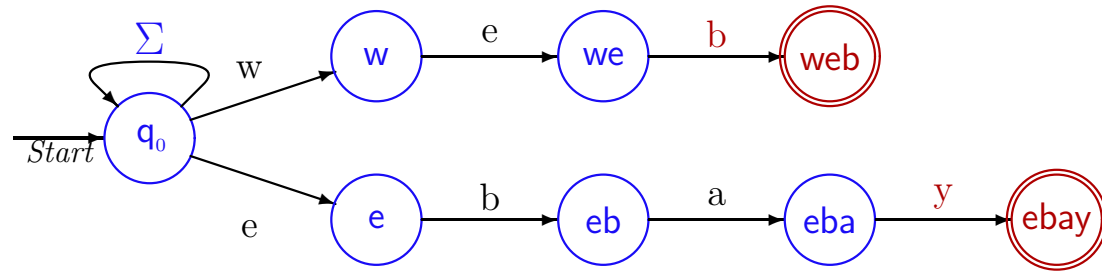


Generierter DEA



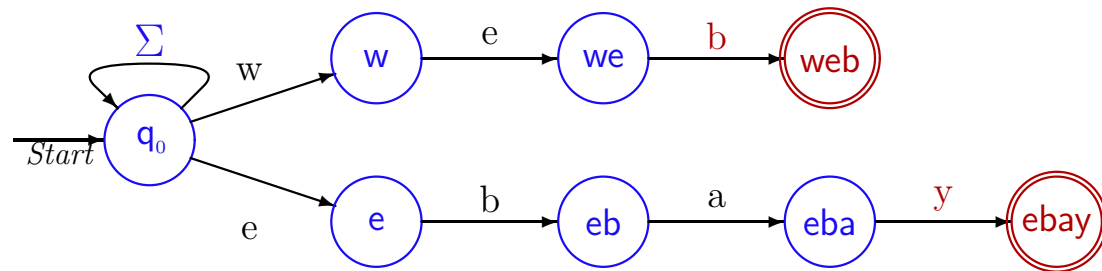
DETERMINISTISCHE AUTOMATEN FÜR TEXTANALYSE

Ursprünglicher ϵ -NEA

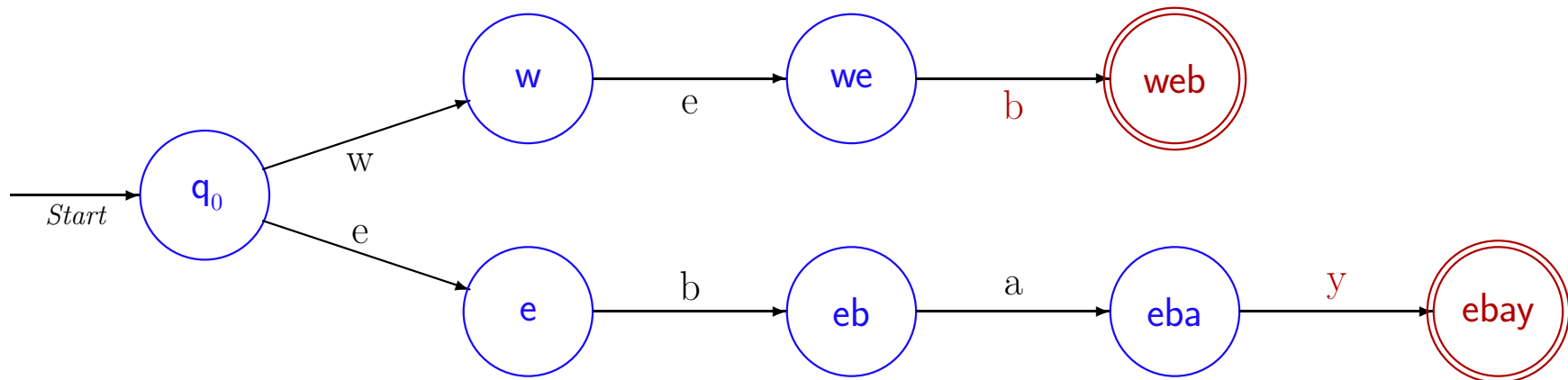


DETERMINISTISCHE AUTOMATEN FÜR TEXTANALYSE

Ursprünglicher ϵ -NEA

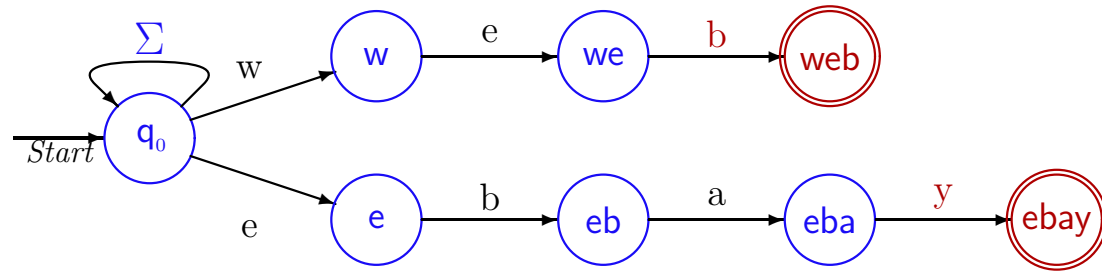


Generierter DEA

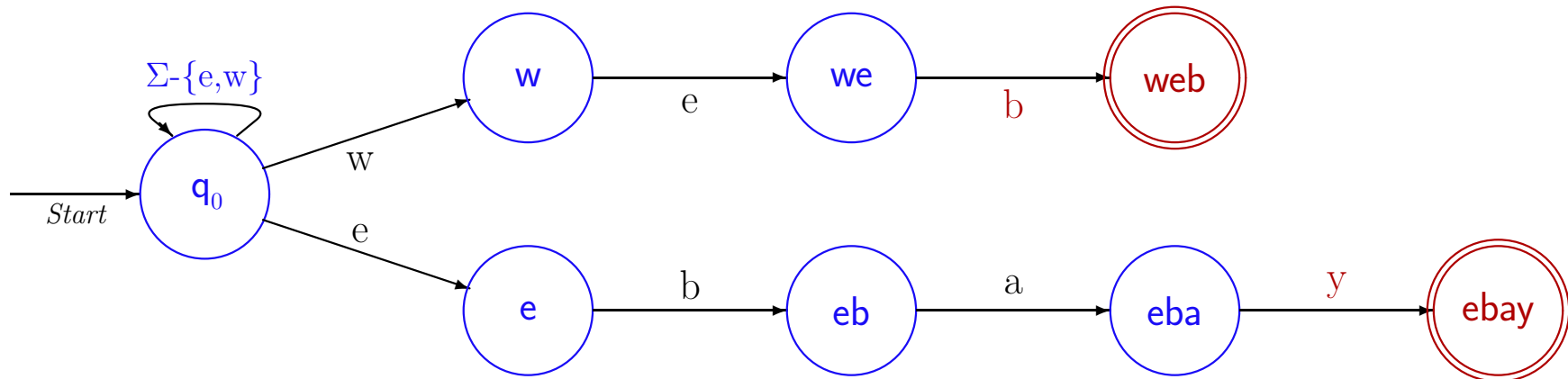


DETERMINISTISCHE AUTOMATEN FÜR TEXTANALYSE

Ursprünglicher ϵ -NEA

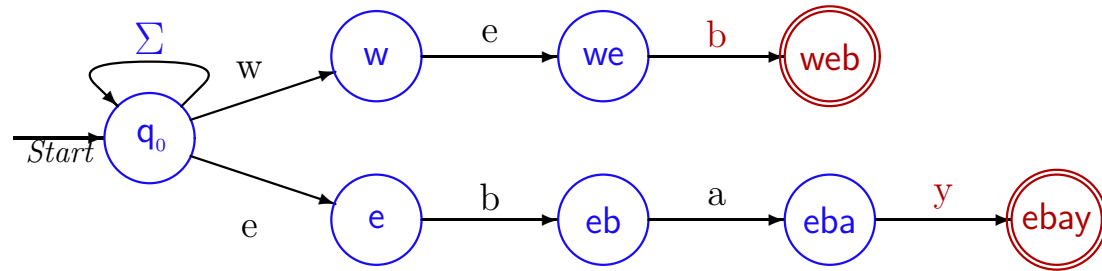


Generierter DEA

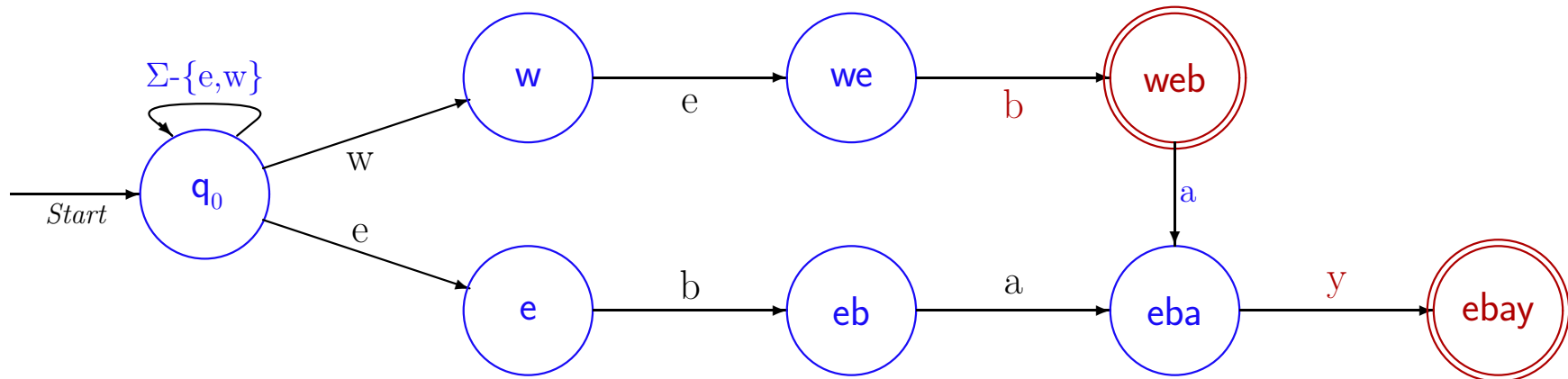


DETERMINISTISCHE AUTOMATEN FÜR TEXTANALYSE

Ursprünglicher ϵ -NEA

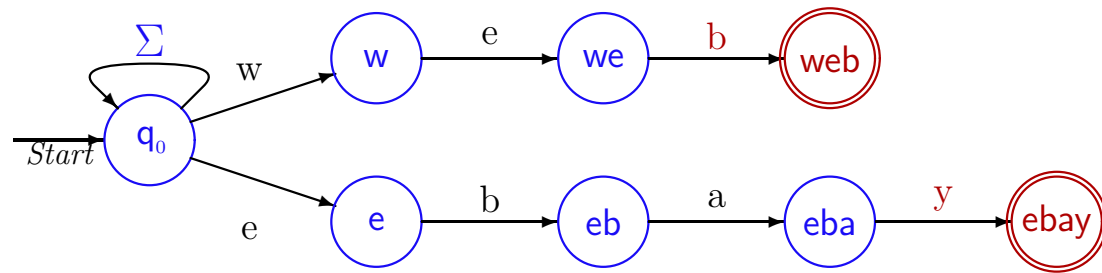


Generierter DEA

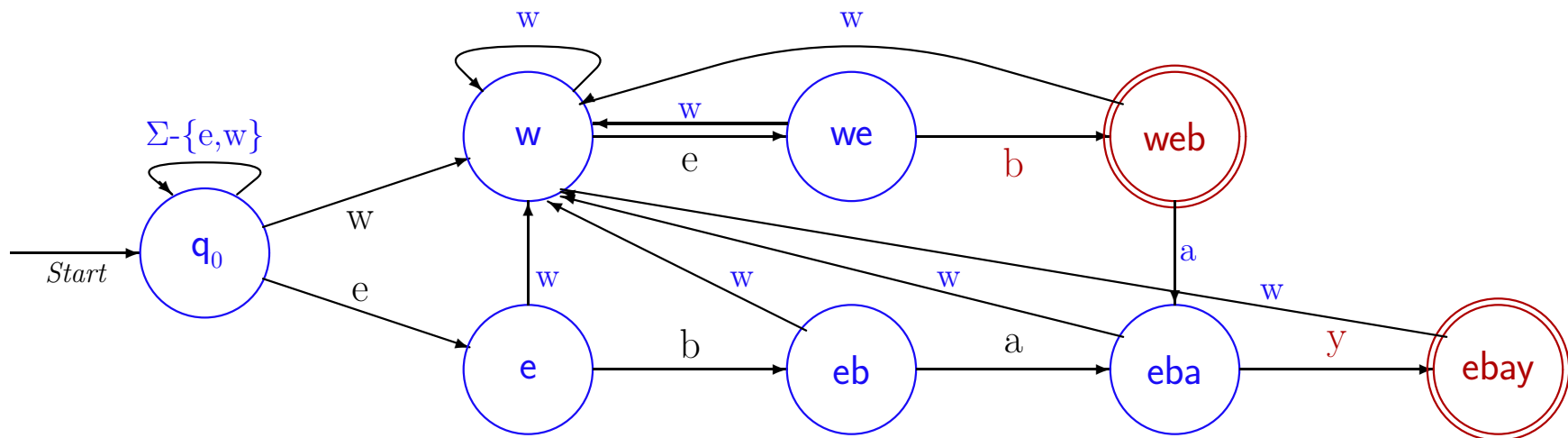


DETERMINISTISCHE AUTOMATEN FÜR TEXTANALYSE

Ursprünglicher ϵ -NEA

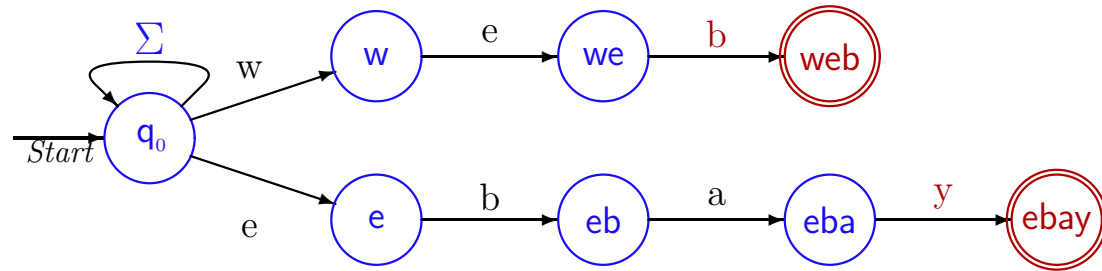


Generierter DEA

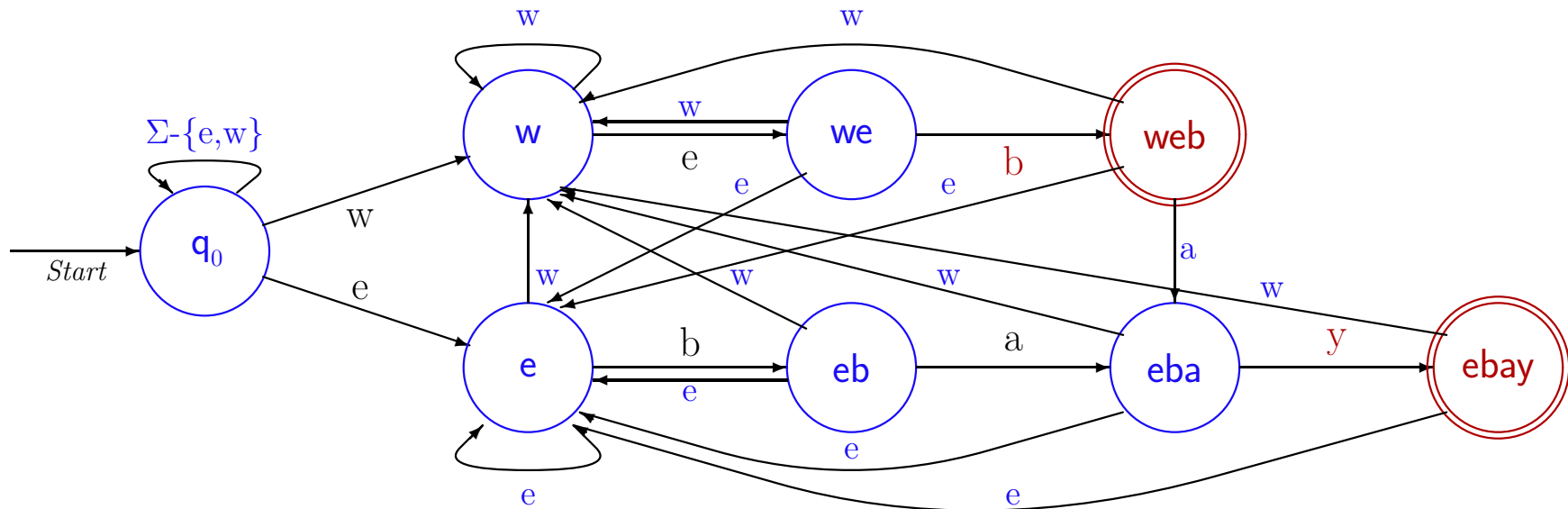


DETERMINISTISCHE AUTOMATEN FÜR TEXTANALYSE

Ursprünglicher ϵ -NEA

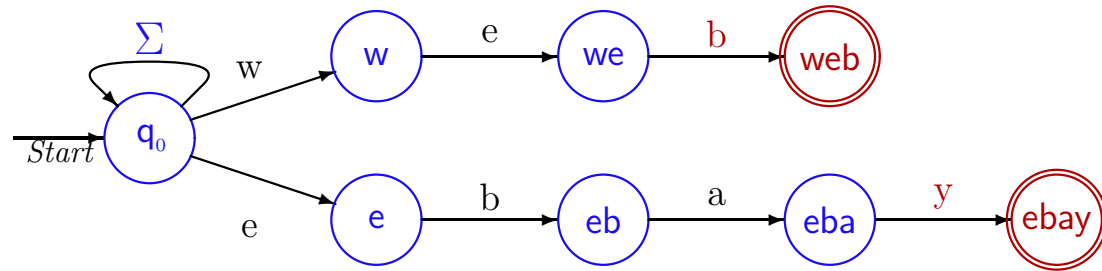


Generierter DEA

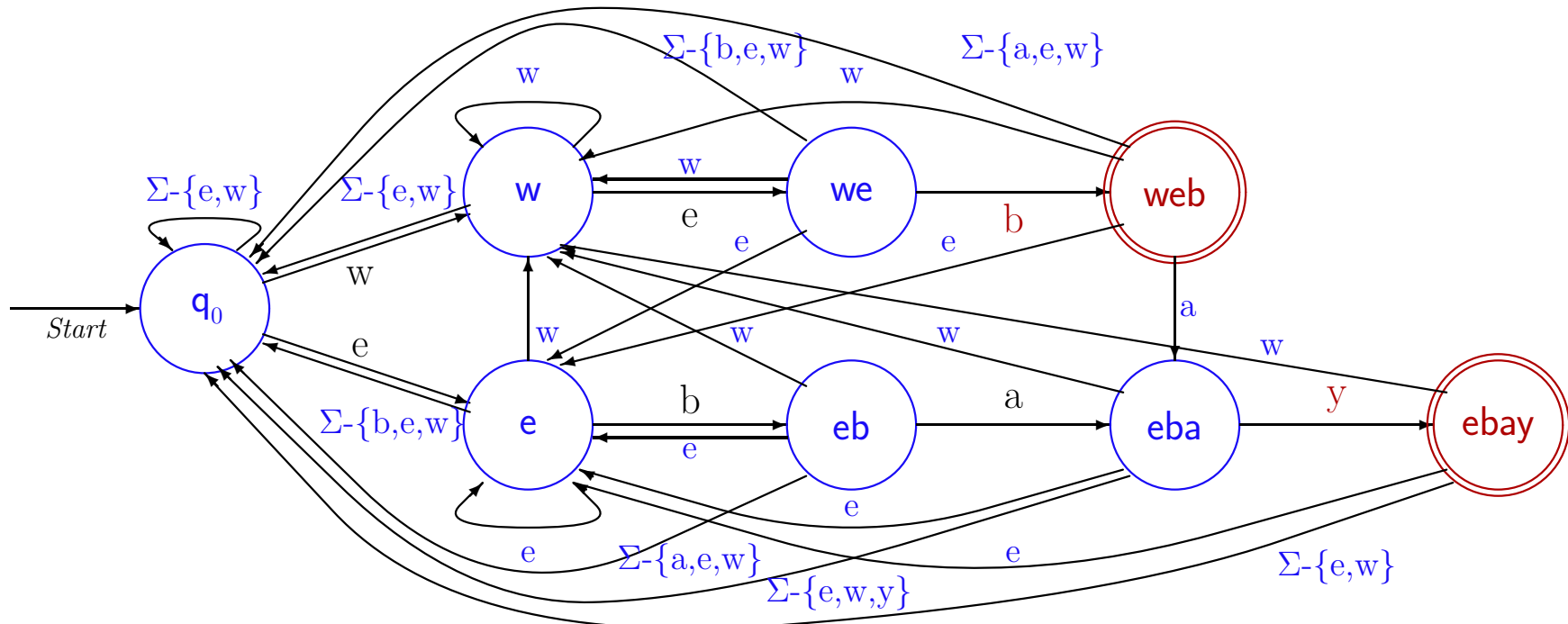


DETERMINISTISCHE AUTOMATEN FÜR TEXTANALYSE

Ursprünglicher ϵ -NEA



Generierter DEA



ANALYSE DER OPTIMIERTEN TEILMENGENKONSTRUKTION

- A_D kann so klein sein wie A_N

ANALYSE DER OPTIMIERTEN TEILMENGENKONSTRUKTION

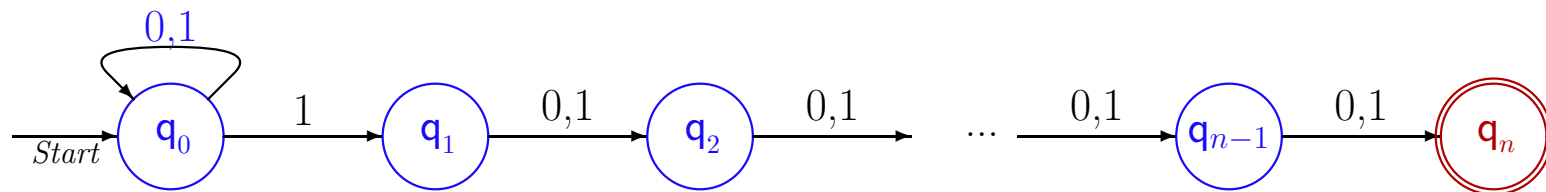
- A_D kann so klein sein wie A_N
 - Nur wenige Teilmengen von Q_N werden wirklich erreicht

ANALYSE DER OPTIMIERTEN TEILMENGENKONSTRUKTION

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

ANALYSE DER OPTIMIERTEN TEILMENGENKONSTRUKTION

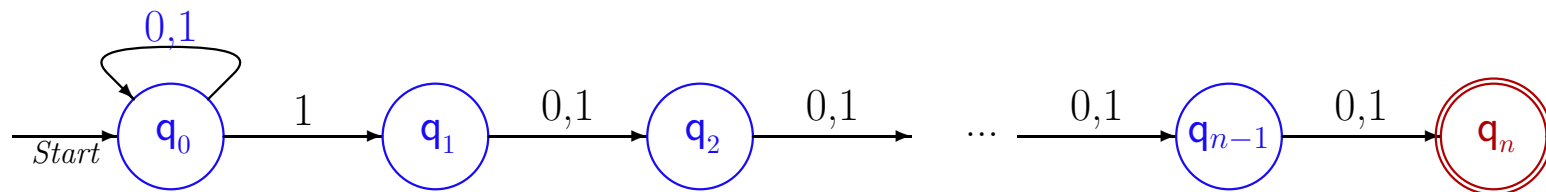
- A_D kann so klein sein wie A_N
 - Nur wenige Teilmengen von Q_N werden wirklich erreicht
- A_D kann exponentiell größer werden



- $L(A_N) = \{w \in \{0, 1\}^* \mid \text{das } n\text{-te Zeichen vor dem Ende ist eine } 1\}$

ANALYSE DER OPTIMIERTEN TEILMENGENKONSTRUKTION

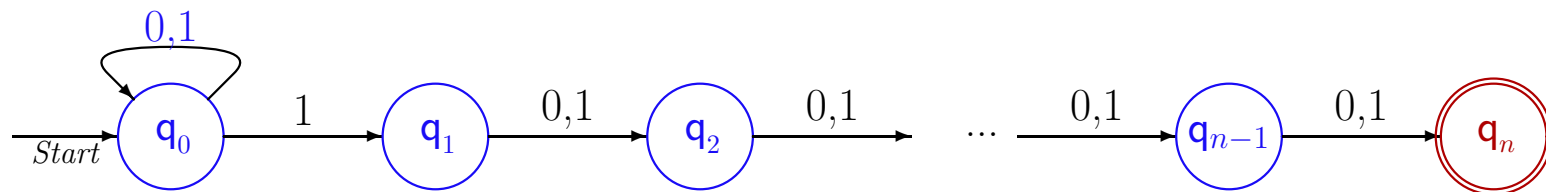
- A_D kann so klein sein wie A_N
 - Nur wenige Teilmengen von Q_N werden wirklich erreicht
- A_D kann exponentiell größer werden



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

ANALYSE DER OPTIMIERTEN TEILMENGENKONSTRUKTION

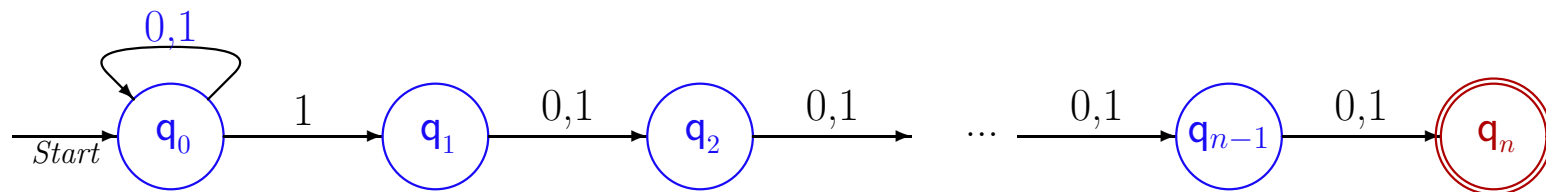
- A_D kann so klein sein wie A_N
 - Nur wenige Teilmengen von Q_N werden wirklich erreicht
- A_D kann exponentiell größer werden



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

ANALYSE DER OPTIMIERTEN TEILMENGENKONSTRUKTION

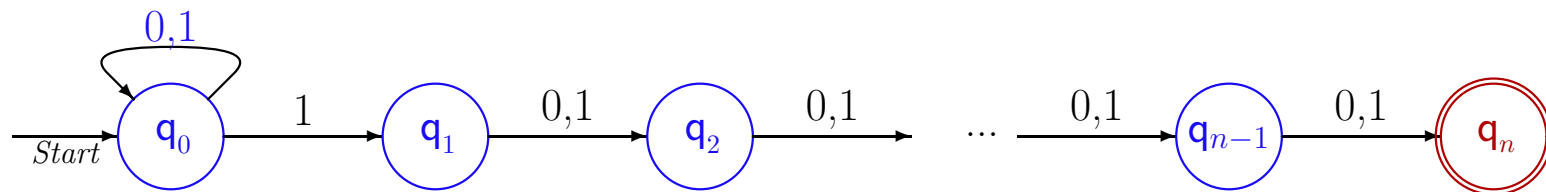
- A_D kann so klein sein wie A_N
 - Nur wenige Teilmengen von Q_N werden wirklich erreicht
- A_D kann exponentiell größer werden



- $L(A_N) = \{w \in \{0, 1\}^* \mid \text{das } n\text{-te Zeichen vor dem Ende ist eine 1}\}$
- **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)**

ANALYSE DER OPTIMIERTEN TEILMENGENKONSTRUKTION

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



- $L(A_N) = \{w \in \{0, 1\}^* \mid \text{das } n\text{-te Zeichen vor dem Ende ist eine 1}\}$
- **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 $q = \delta_A(q_0, w0^{i-1}) = \delta_A(q_0, v0^{i-1})$ folgt $q \in F$ und $q \notin F$

● Deterministische Endliche Automaten (**DEA**)

- Endliche Menge von **Zuständen**, endliche Menge von **Eingabesymbolen**
- Ein fester Startzustand, null oder mehr **akzeptierende Zustände**
- **Überföhrungsfunktion** bestimmt Änderung des Zustands bei Abarbeitung der Eingabe
- **Erkannte Sprache**: Eingaben, deren Abarbeitung in einem akzeptierenden Zustand endet

● Deterministische Endliche Automaten (**DEA**)

- Endliche Menge von **Zuständen**, endliche Menge von **Eingabesymbolen**
- Ein fester Startzustand, null oder mehr **akzeptierende Zustände**
- **Überföhrungsfunktion** 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

● Deterministische Endliche Automaten (**DEA**)

- Endliche Menge von **Zuständen**, endliche Menge von **Eingabesymbolen**
- Ein fester Startzustand, null oder mehr **akzeptierende Zustände**
- **Überföhrungsfunktion** 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 Überföhrungsfunktion** und **Zustandsüberföhrung bei leerer Eingabe**
- Durch **Teilmengenkonstruktion** in äquivalenten DEA transformierbar