

Kapitel 2

Reguläre Sprachen

2.4

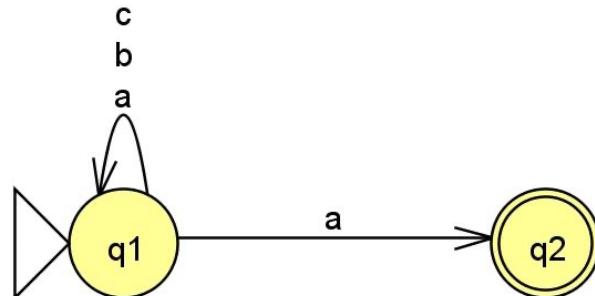
Nichtdeterministischer Endlicher Automat mit ε -Übergängen (ε -NEA)

Prof. Dr. Robert Preis
Fachbereich Informatik
Fachhochschule Dortmund
Robert.Preis@fh-dortmund.de

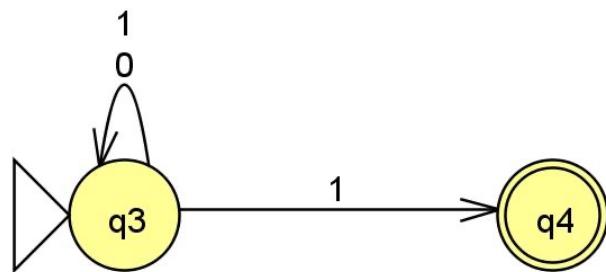
**Alle Materialien (Folien, Übungsblätter, etc.) dieser
Veranstaltung sind urheberrechtlich geschützt und nur
von Teilnehmern dieser Veranstaltung und im Rahmen
dieser zu verwenden. Eine anderweitige Verwendung
oder Verbreitung ist nicht gestattet.**

Vereinigung von NEAs

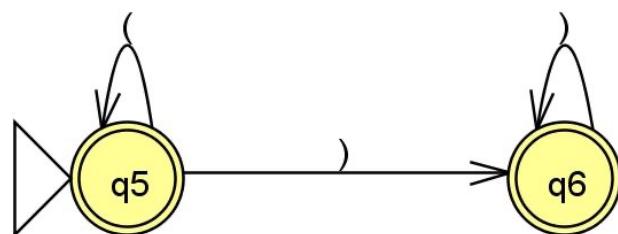
$L_1 = \{w \in \{a,b,c\}^* \mid w \text{ hört mit } a \text{ auf}\}$



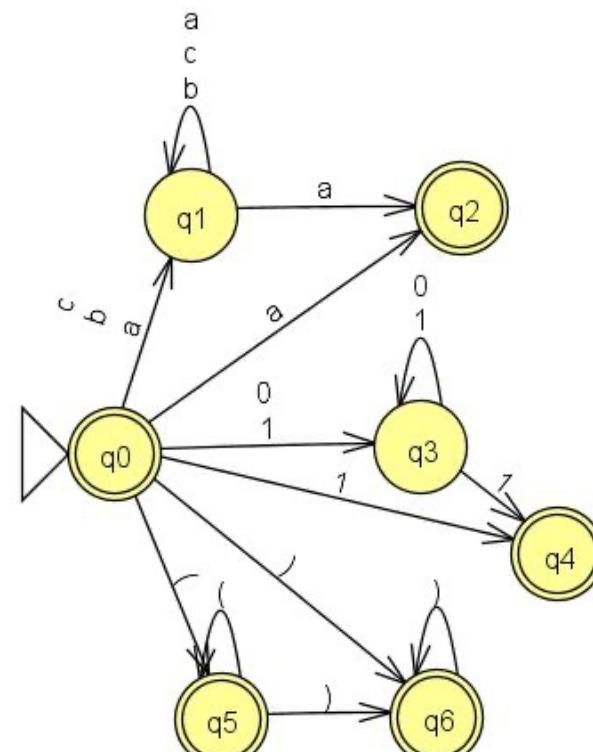
$L_2 = \{w \in \{0,1\}^* \mid w \text{ hört mit } 1 \text{ auf}\}$



$L_3 = \{w \in \{(),()\}^* \mid w = (n)^m \text{ mit } n, m \in \mathbb{N}_0\}$



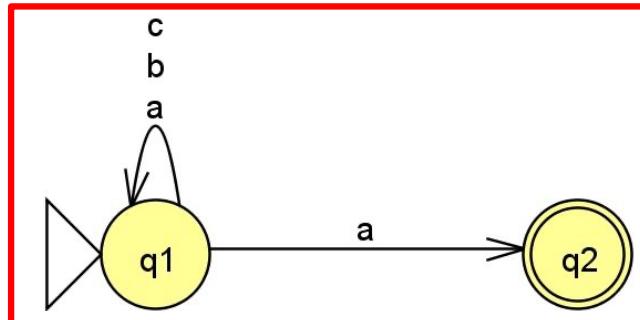
Wie sieht ein Automat aus für $L_1 \cup L_2 \cup L_3$
 $= \{ w \in \{a,b,c,0,1,(),()\}^* \mid$
 $w \in \{a,b,c\}^* \text{ hört mit } a \text{ auf} \text{ oder}$
 $w \in \{0,1\}^* \text{ hört mit } 1 \text{ auf} \text{ oder}$
 $w = (n)^m \text{ für } n, m \in \mathbb{N}_0\}$?



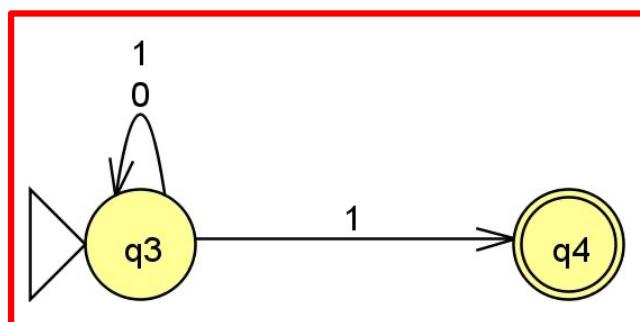
Geht das auch anders, d.h. leichter?

Vereinigung von NEAs einfach gemacht: NEAs mit ϵ -Übergängen

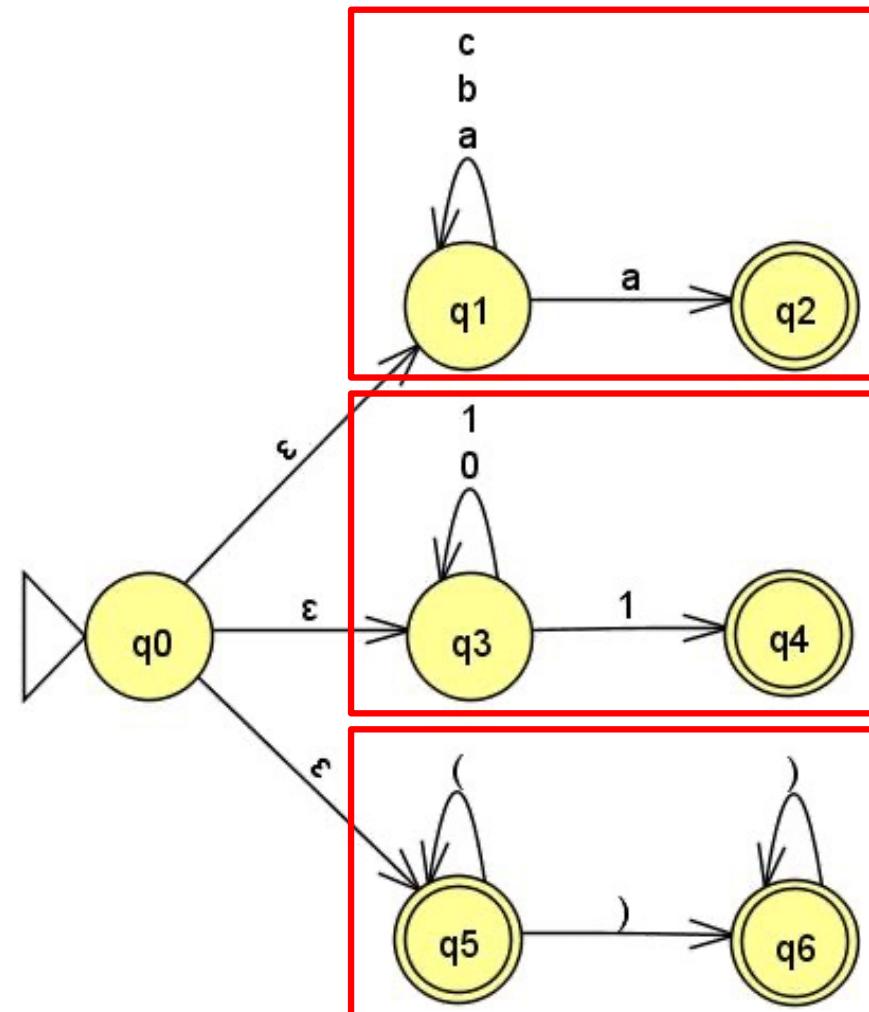
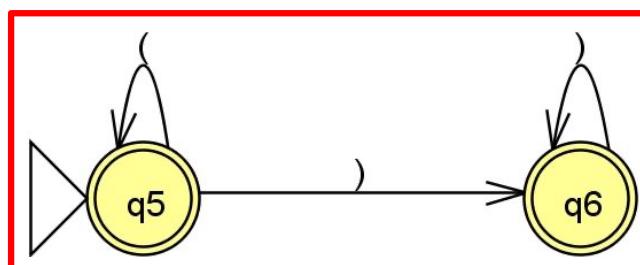
$L_1 = \{w \in \{a,b,c\}^* \mid w \text{ hört mit } a \text{ auf}\}$



$L_2 = \{w \in \{0,1\}^* \mid w \text{ hört mit } 1 \text{ auf}\}$

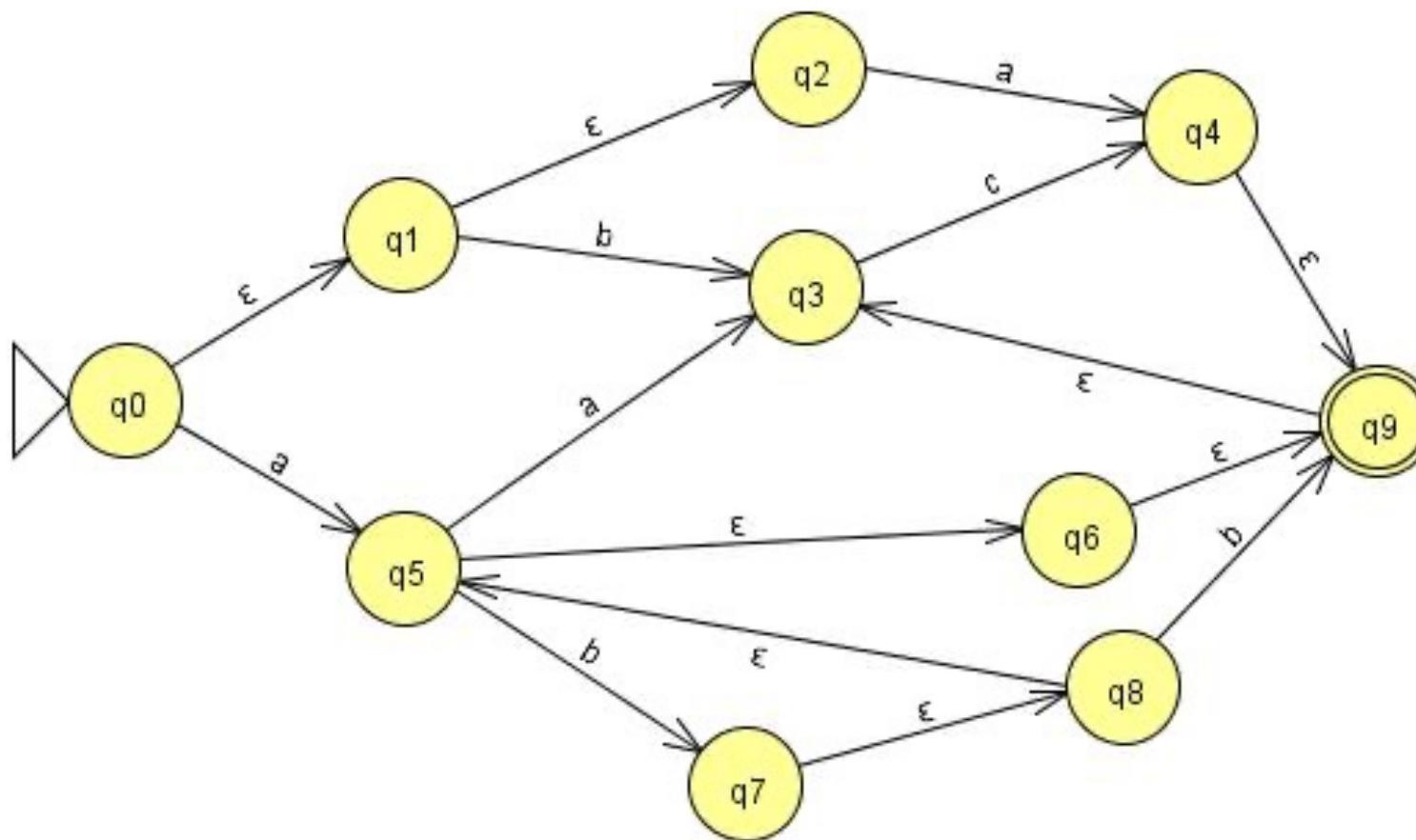


$L_3 = \{w \in \{(),()\}^* \mid w = ({}^n)^m \text{ mit } n, m \in \mathbb{N}_0\}$



$$L = L_1 \cup L_2 \cup L_3$$

NEA mit ϵ -Übergang: Beispiel



Zusätzliche ϵ -Übergänge:

- Spontane Übergänge zwischen Zuständen
- Automat geht ohne Eingabe in einen neuen Zustand über
- Kann das Programmieren erleichtern

Kann man das Wort „a“ ableiten? Und wie?

Definition eines ε -NEA

Ein Nichtdeterministischer Endlicher Automat mit ε -Übergängen (ε -NEA) ist ein 5-Tupel

$$A = (Q, \Sigma, \delta, q_0, F)$$

mit

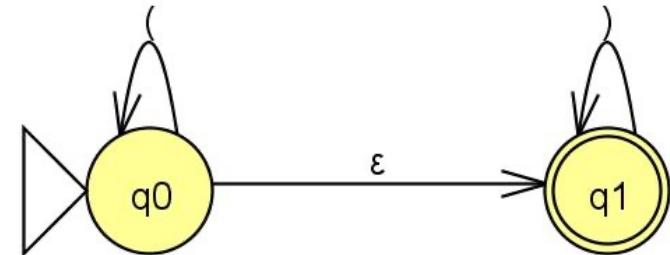
Q : nichtleere endliche Zustandsmenge

Σ : endliches Eingabealphabet

δ : $Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow P(Q)$ Zustandsüberführungs-funktion (Übergangsfunktion)

$q_0 \in Q$: Startzustand (Anfangszustand)

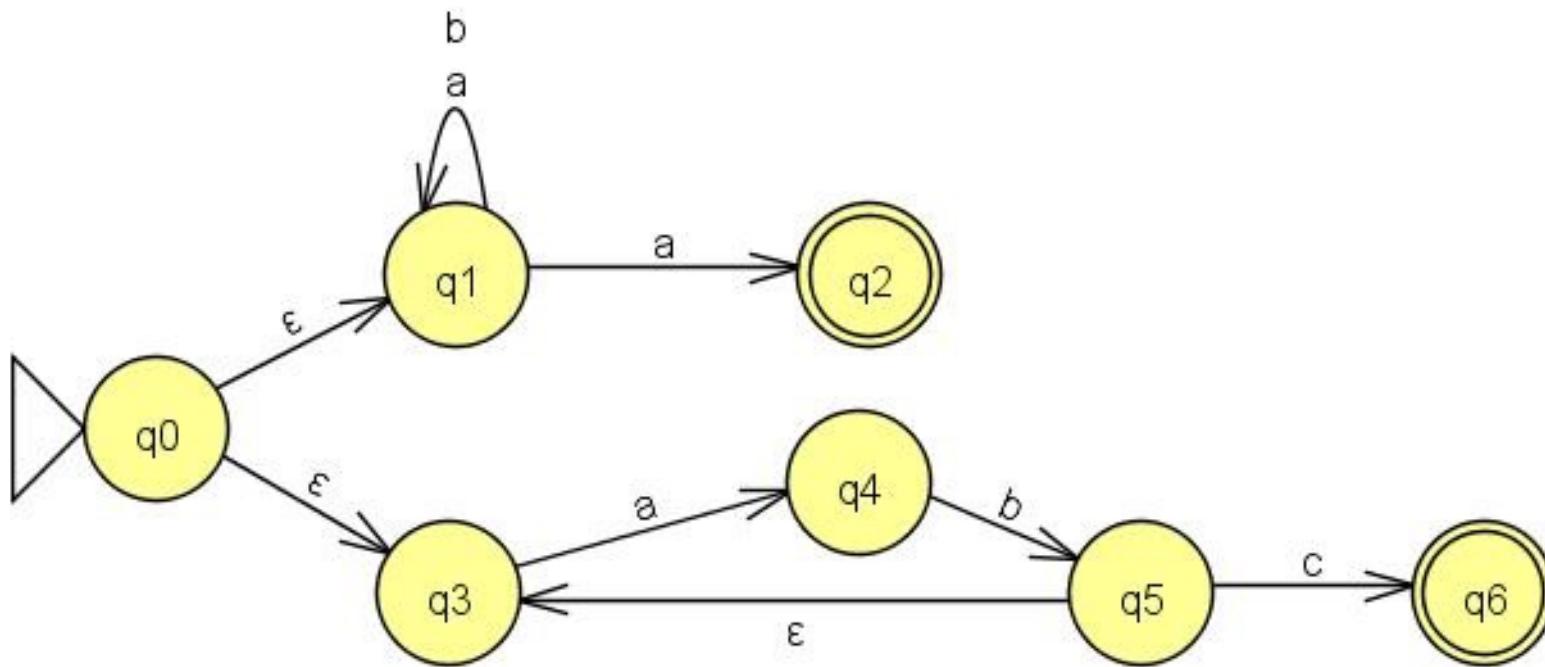
$F \subseteq Q$: Menge von akzeptierenden Zuständen (Endzuständen)



ε-NEA	()	ε
$\rightarrow q_0$	$\{q_0\}$	$\{\}$	$\{q_1\}$
$* q_1$	$\{\}$	$\{q_1\}$	$\{\}$

Zusätzlich zum NEA ist das leere Wort als weiterer Übergang möglich.

Am 5-Tupel kann man einen NEA nicht von einem ε-NEA unterscheiden!

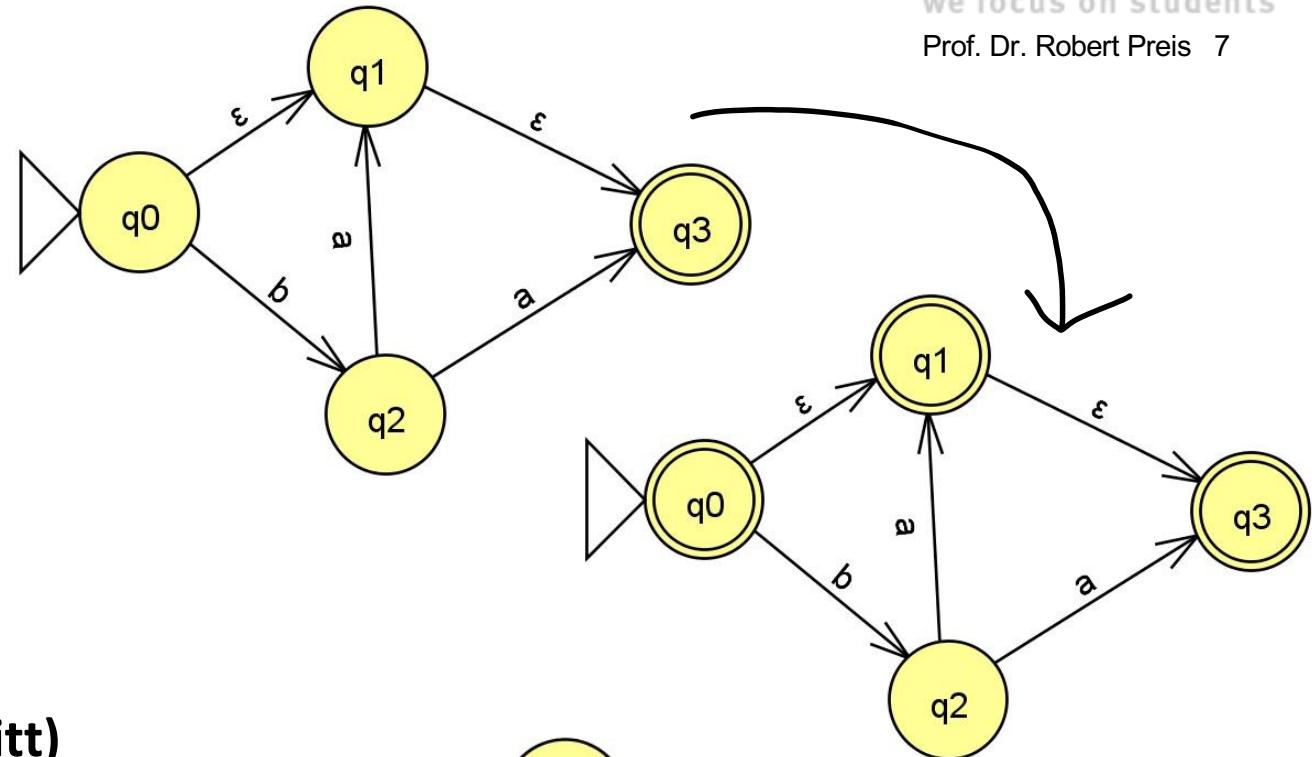


Ableitung von ababc:

Umwandlung ϵ -NEA in NEA: Schritte 1 und 2

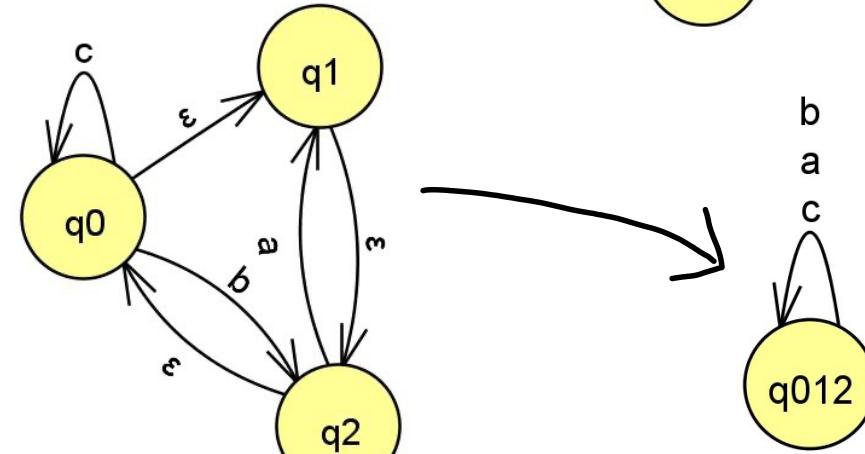
Schritt 1: Endzustände

Alle Zustände, von denen man mit (einem oder mehreren) ϵ -Übergängen zu einem Endzustand kommen kann, werden selbst Endzustände.



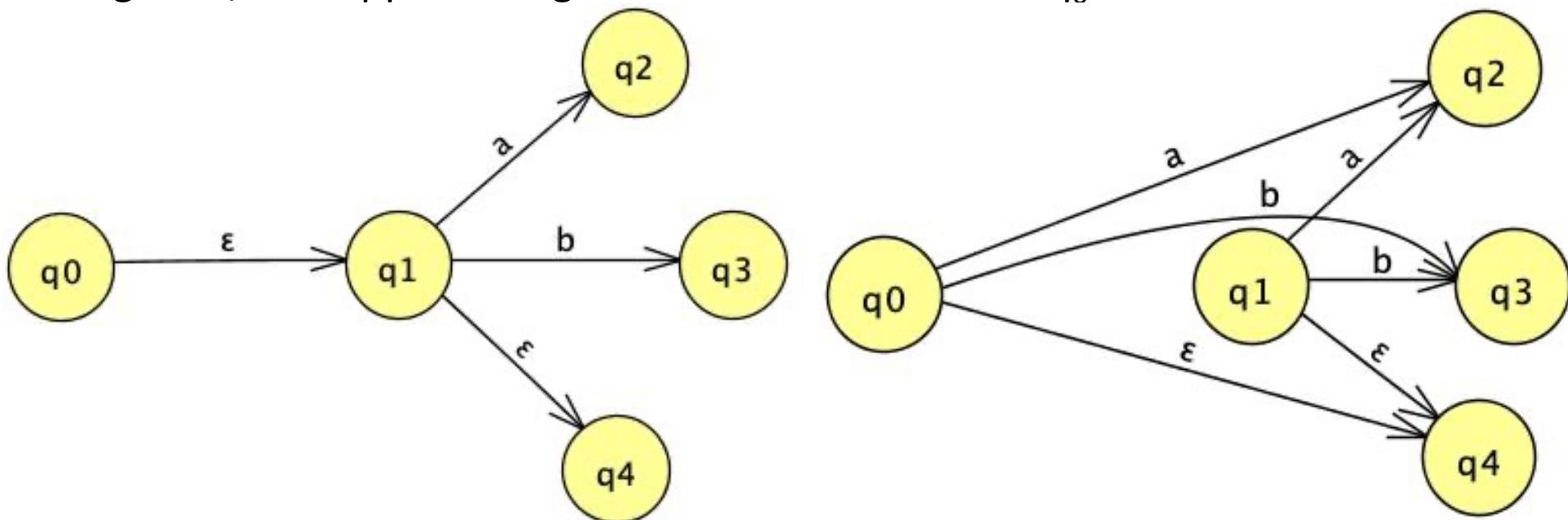
Schritt 2: ϵ -Zyklen (Ausschnitt)

Alle Zyklen von ϵ -Übergängen (d.h. man kann von jedem Zustand in dem Zyklus jeden anderen Zustand durch ϵ -Übergänge erreichen) werden zu einem Zustand geschrumpft. Weitere Übergänge werden zu Schleifen.



Umwandlung ϵ -NEA in NEA: Schritt 3

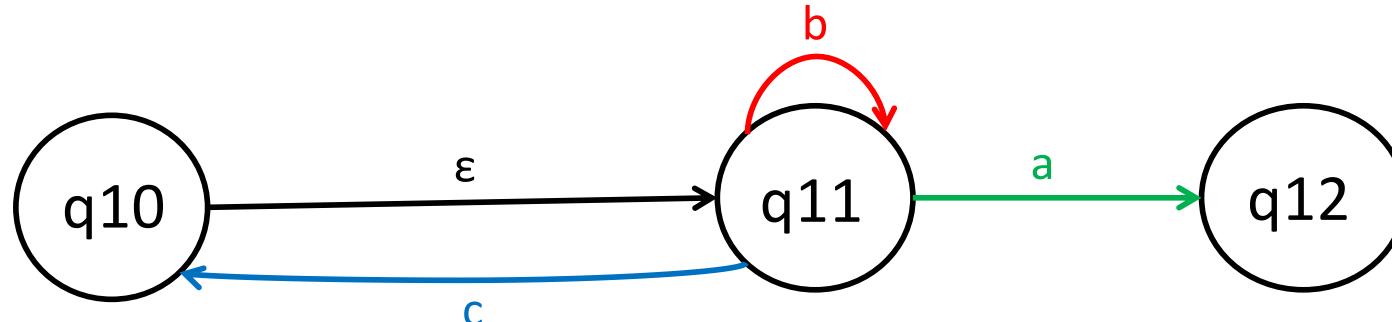
Schritt 3: ϵ –Übergänge werden nacheinander (in beliebiger Reihenfolge) ersetzt
 ϵ -Übergang q_0 nach q_1 wird gelöscht. Dafür werden alle Übergänge, die aus q_1 rausgehen, verdoppelt und gehen zusätzlich auch aus q_0 .



Beispiel: Für den Übergang **a** von **q1** nach **q2** kommt ein
Übergang **a** von **q0** nach **q2** zusätzlich hinzu

Achtung: es können neue ϵ -Übergänge entstehen! ... Aber keine Sorge...😊

Umwandlung ε -NEA in NEA: Sonderfälle in Schritt 3

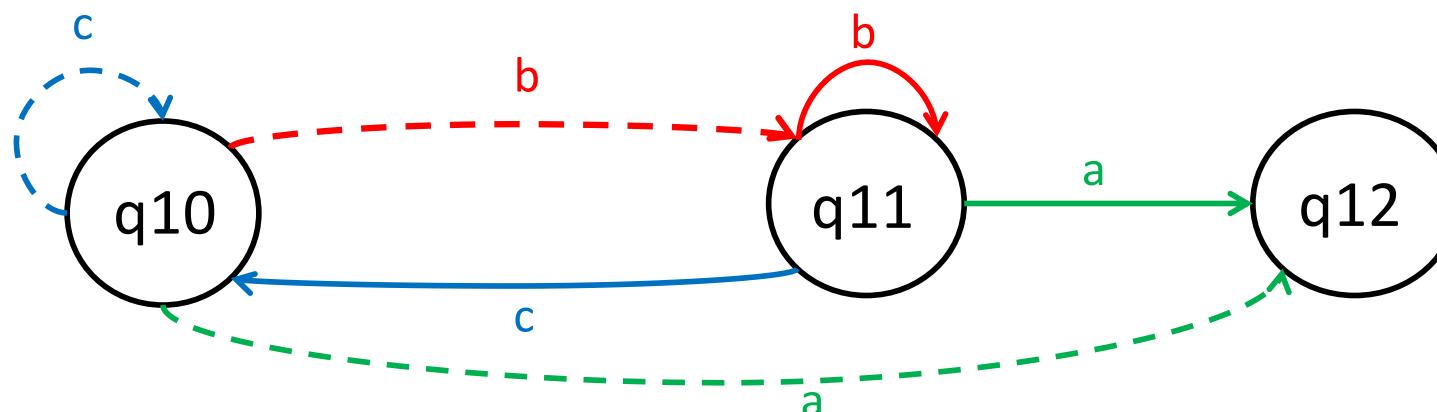


Regelfall:

- Von q_{11} geht ein a -Übergang zu einem weiteren Zustand

Sonderfälle:

- Von q_{11} geht eine b -Schleife nach q_{11}
- Von q_{11} geht ein c -Übergang zurück nach q_{10}

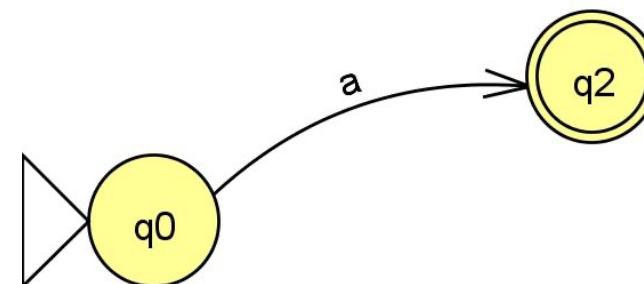
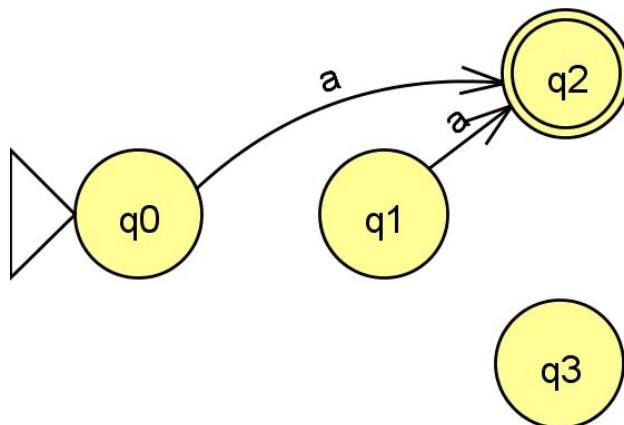


Die gestrichelten Übergänge kommen neu hinzu!

Umwandlung ε -NEA in NEA: Schritt 4

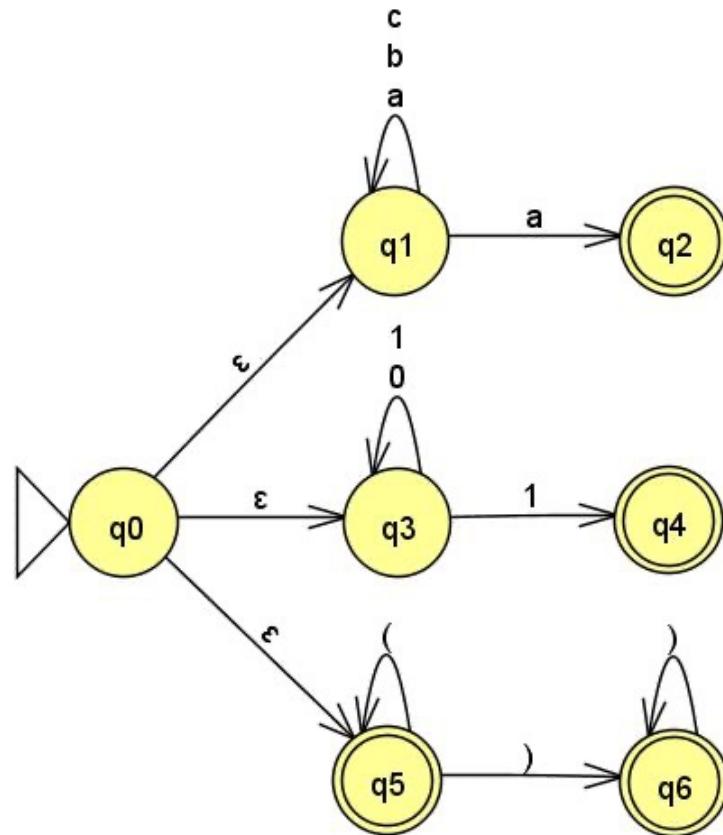
Schritt 4: Lösche alle nicht erreichbaren Zustände

Diese können jederzeit gelöscht werden!

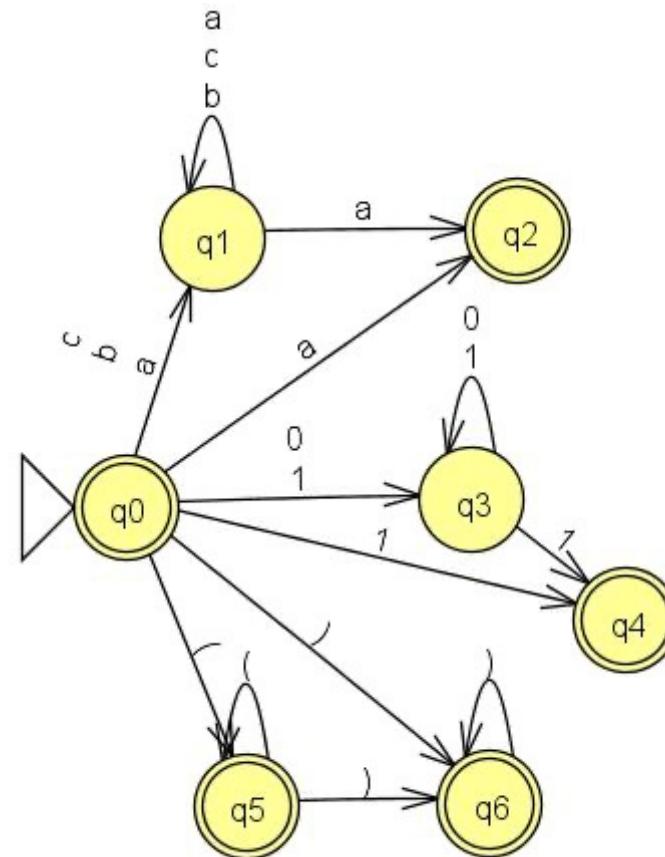


Umwandlung ε -NEA in NEA: Elimination der ε -Übergänge

ε -NEA



Umwandlung in NEA



...sehr bequemes Hilfsmittel beim
Automatenentwurf!

vgl. Folie 2, unsere „Vereinigung“

Zusammenfassung

- ϵ -NEAs sind wie NEAs, aber zusätzlich ist eine Zustandsüberführung bei leerer Eingabe möglich.
- Sie erleichtern die Programmierung.
- Sie sind schwer durch Software zu simulieren.
- Man kann einen ϵ -NEA in einen NEA umwandeln, der die selbe Sprache akzeptiert.
- Automaten (DEA, NEA, ϵ -NEA) akzeptieren die selben Sprachen, die von TYP-3 Grammatiken erzeugt werden.
- Die Automaten akzeptieren genau die regulären Sprachen, d.h. die TYP3-Sprachen.