universität freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

CoNe Freiburg

Theoretische Informatik

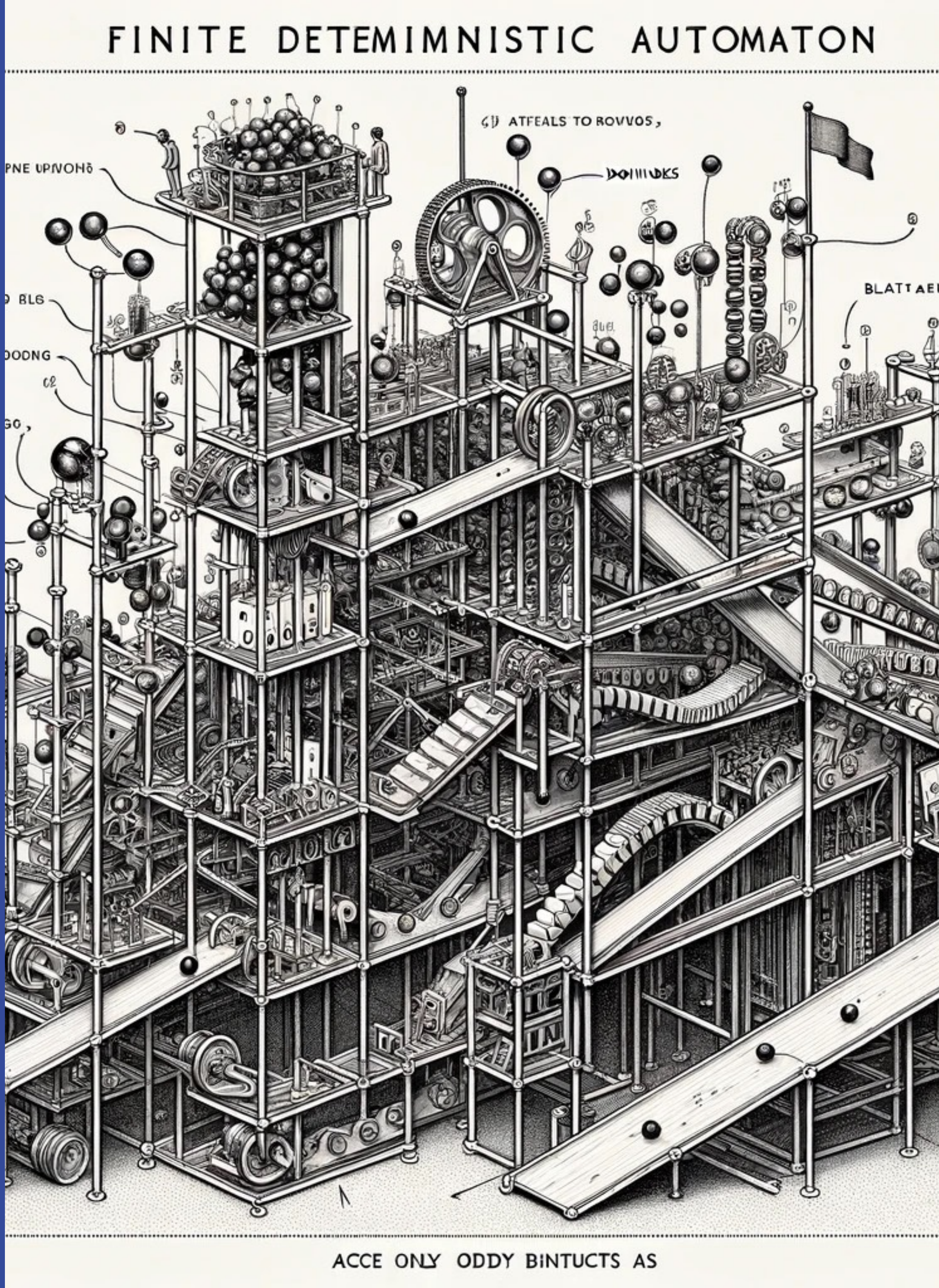
2. Endliche Automaten, Reguläre Sprachen und Reguläre Ausdrücke

Version 15.04.2024
Sommersemester 2024

Technische Fakultät
Christian Schindelhauer
Sneha Mohanty

universität freiburg

Ein einfacher Automat

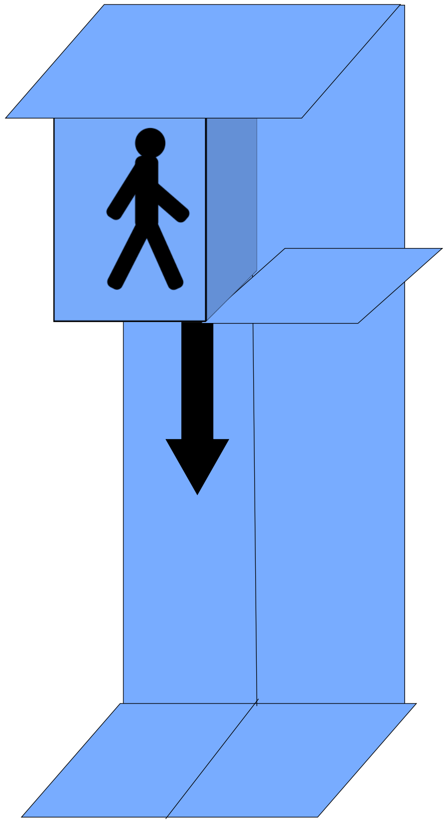


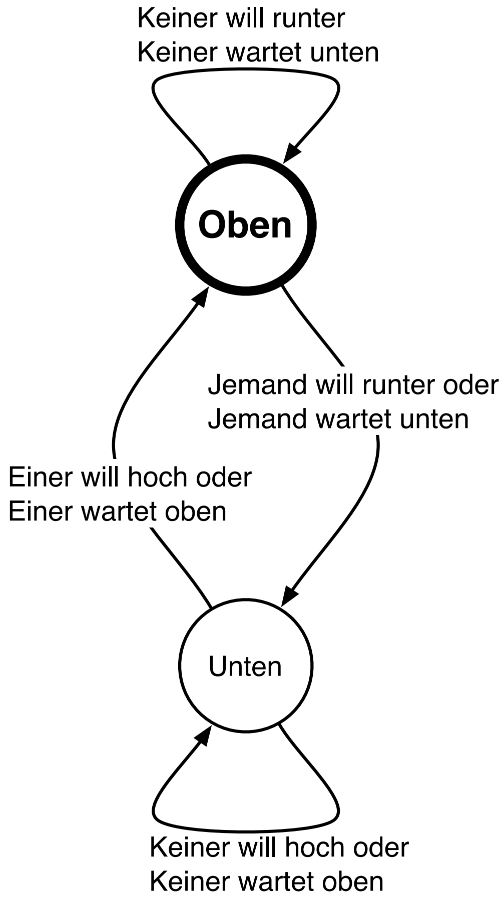
Formale Sprachen und
Endliche Automaten

Ein einfacher Automat
2-Stock-Lift-Automat

Lift mit zwei Stockwerken

- Eingaben:
  - Im Lift will jemand nach oben
  - Im Lift will jemand nach unten
  - Unten Bedarf
  - Oben Bedarf
- Zwei Zustände:
  - **Lift oben**
  - Lift unten

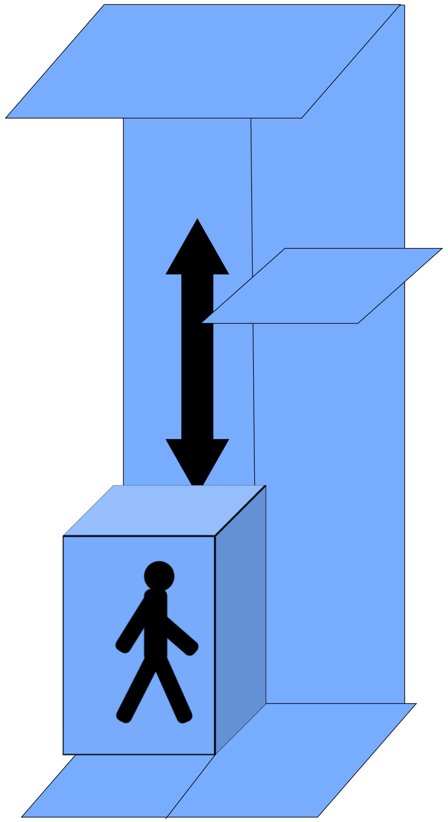


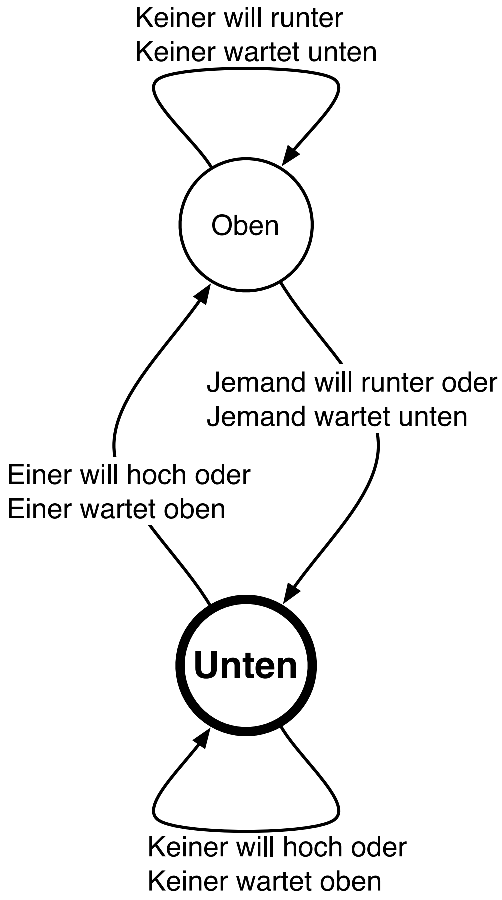


Ein einfacher Automat
2-Stock-Lift-Automat

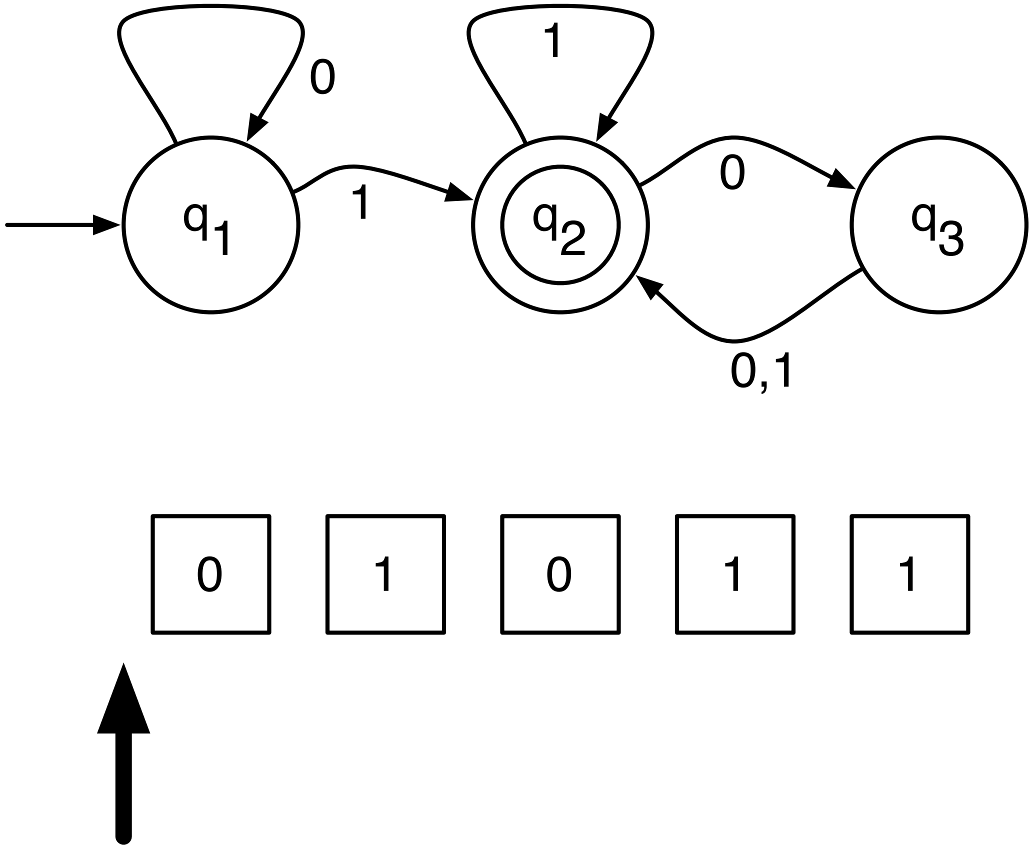
Lift mit zwei Stockwerken

- Eingaben:
  - Im Lift will jemand nach oben
  - Im Lift will jemand nach unten
  - Unten Bedarf
  - Oben Bedarf
- Zwei Zustände:
  - Lift oben
  - **Lift unten**



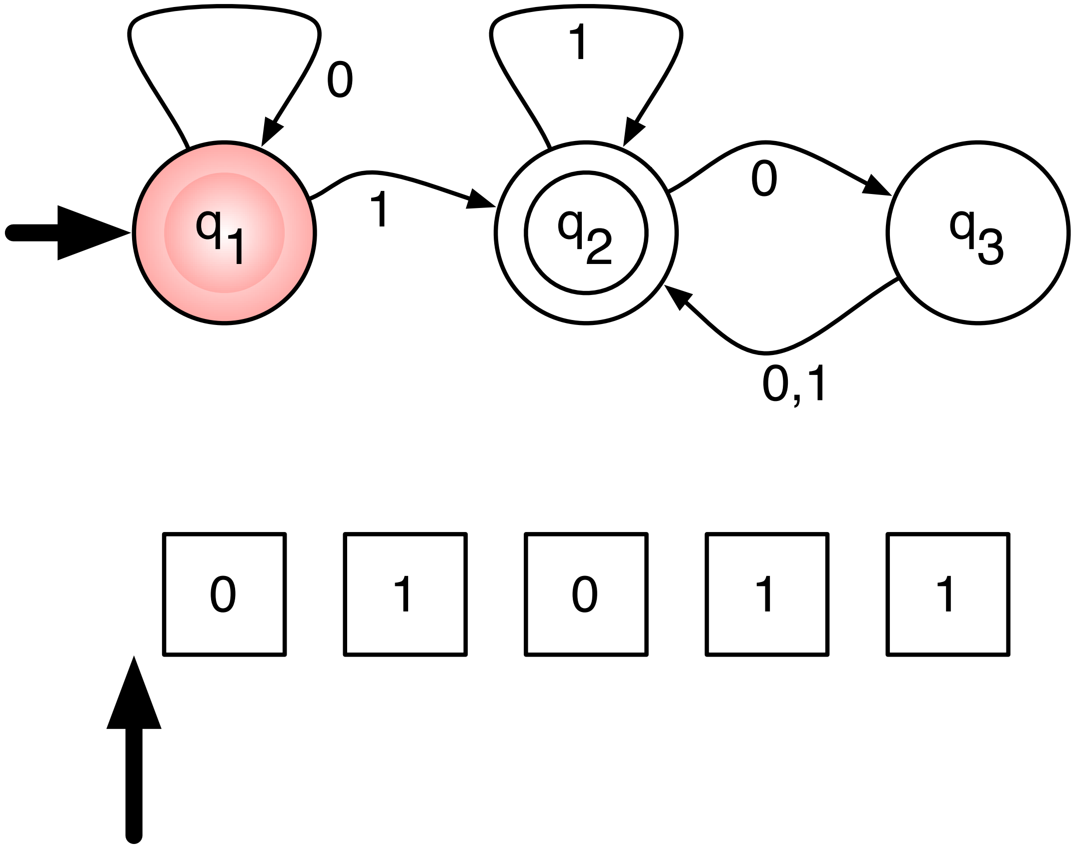


DFA
Ein Beispiel



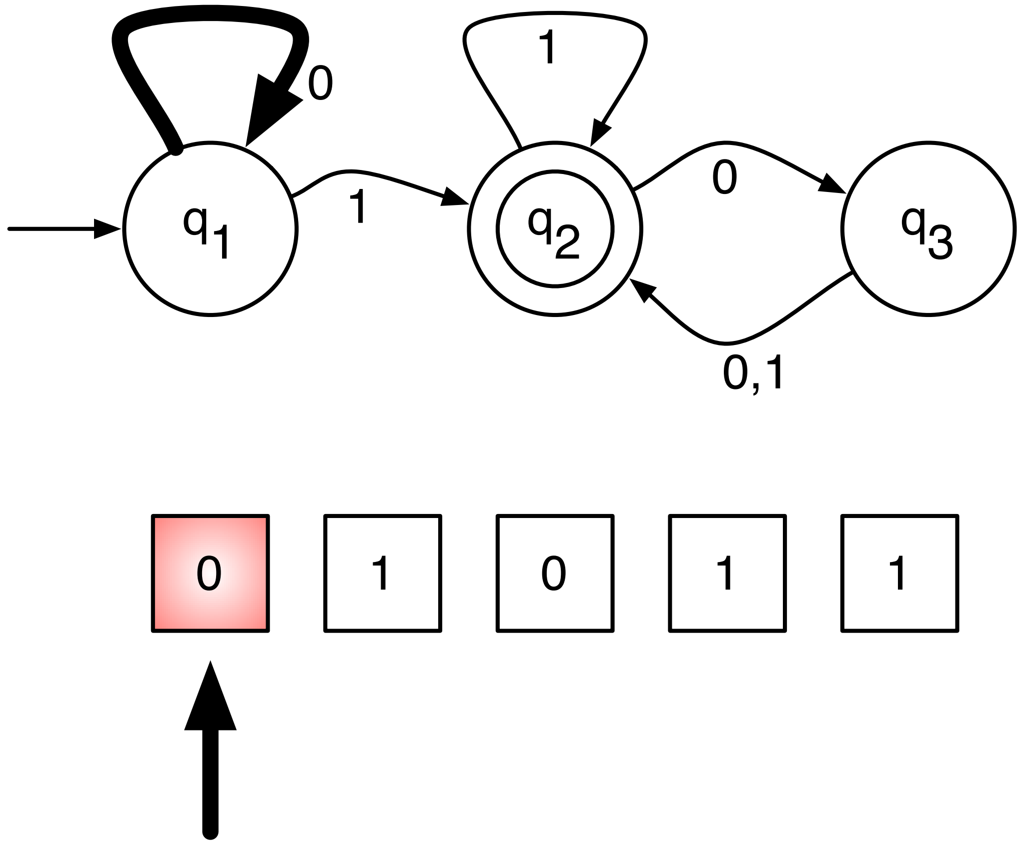
DFA: Ein Beispiel

Startzustand q_1



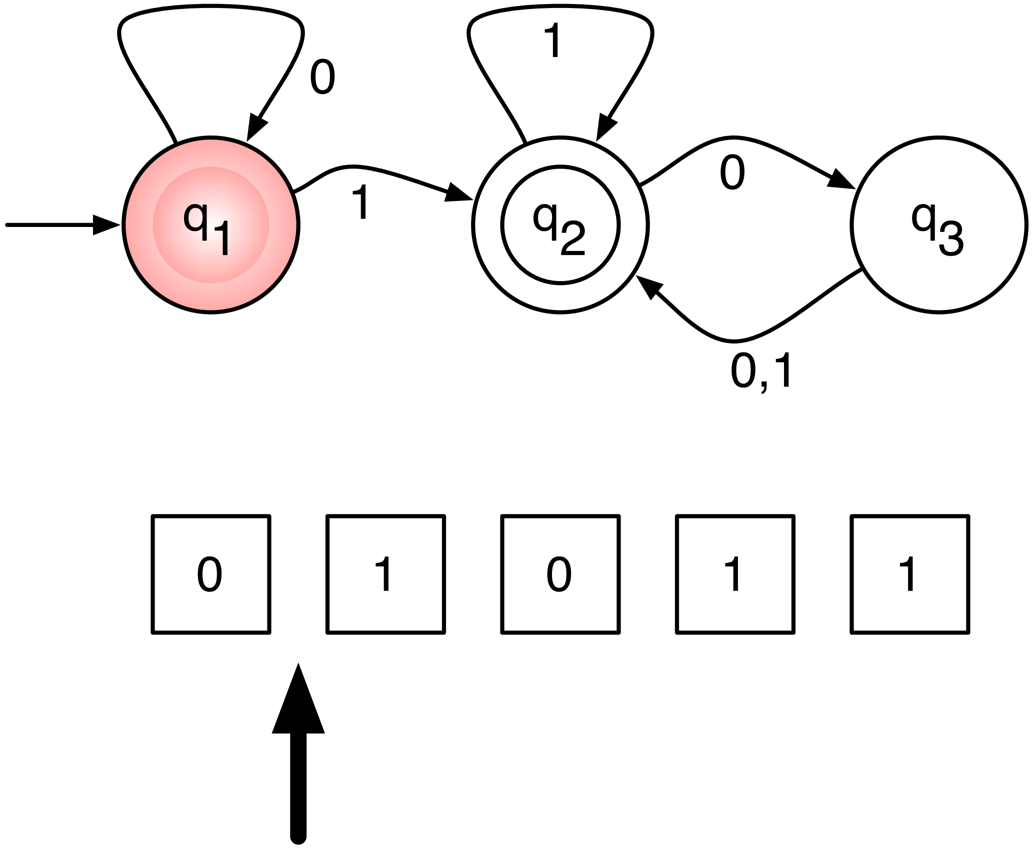
DFA: Ein Beispiel

Lesen des ersten Zeichens: 0

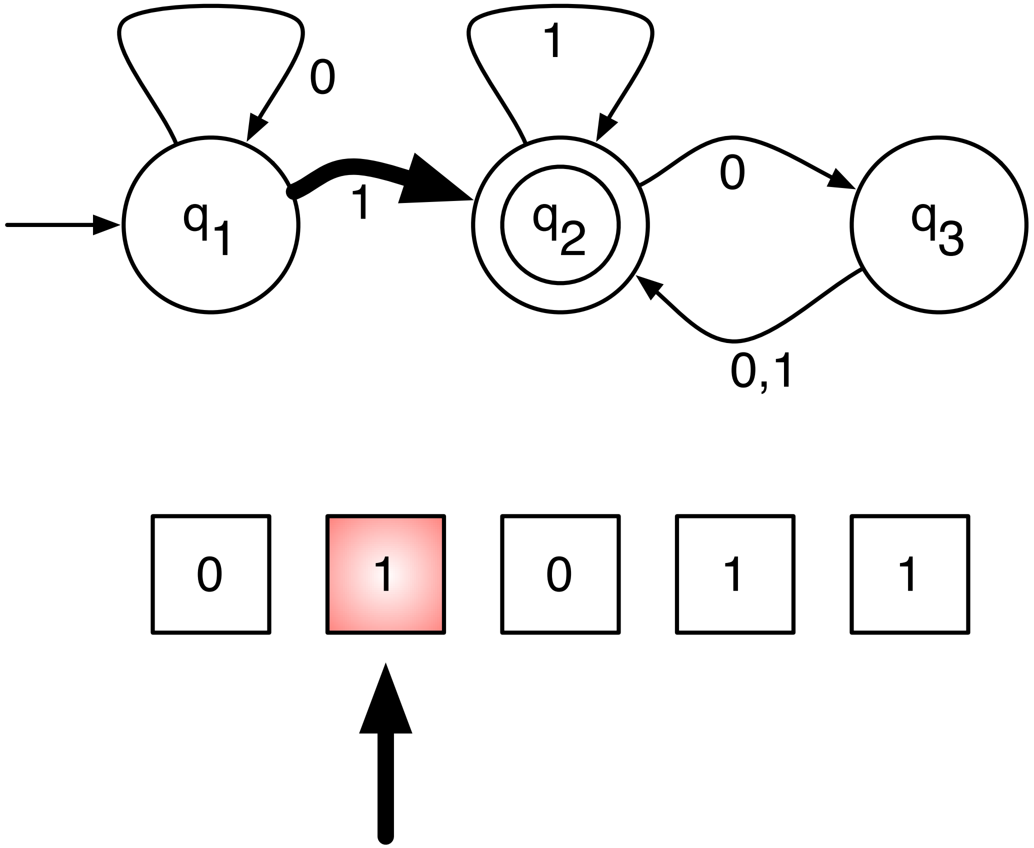


DFA: Ein Beispiel

Zustandsübergang zu q_1

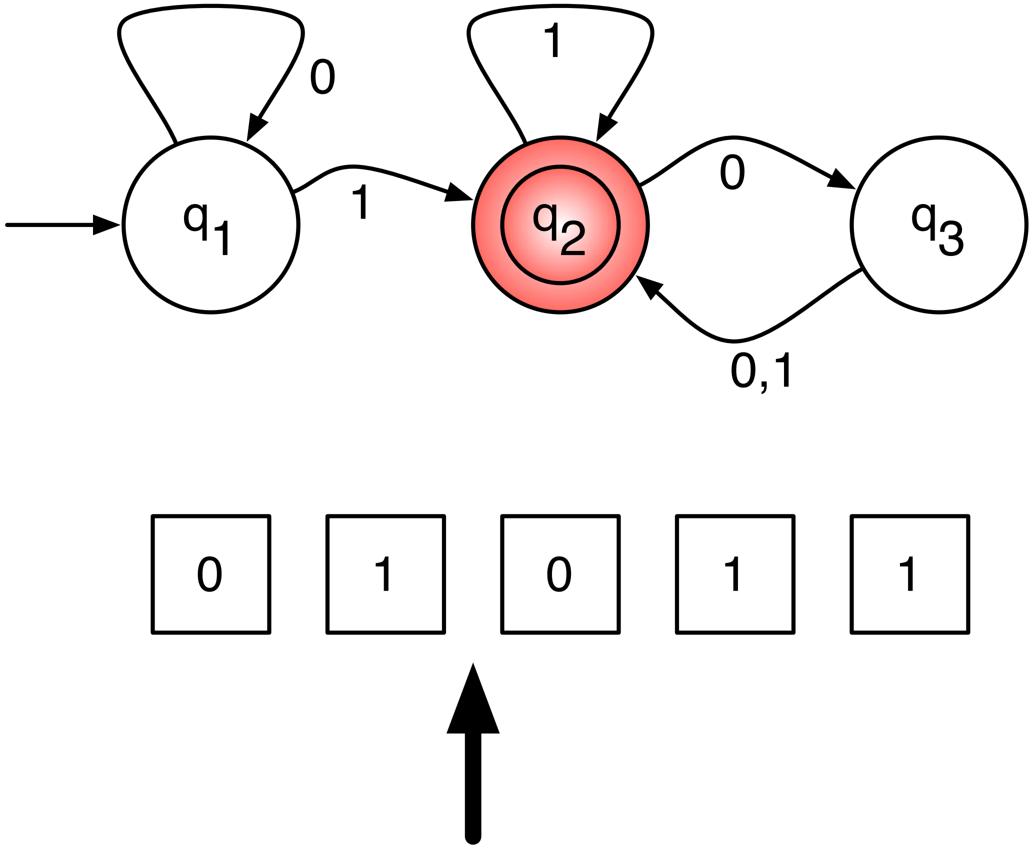


DFA: Ein Beispiel
Lesen des 2. Zeichens: 1



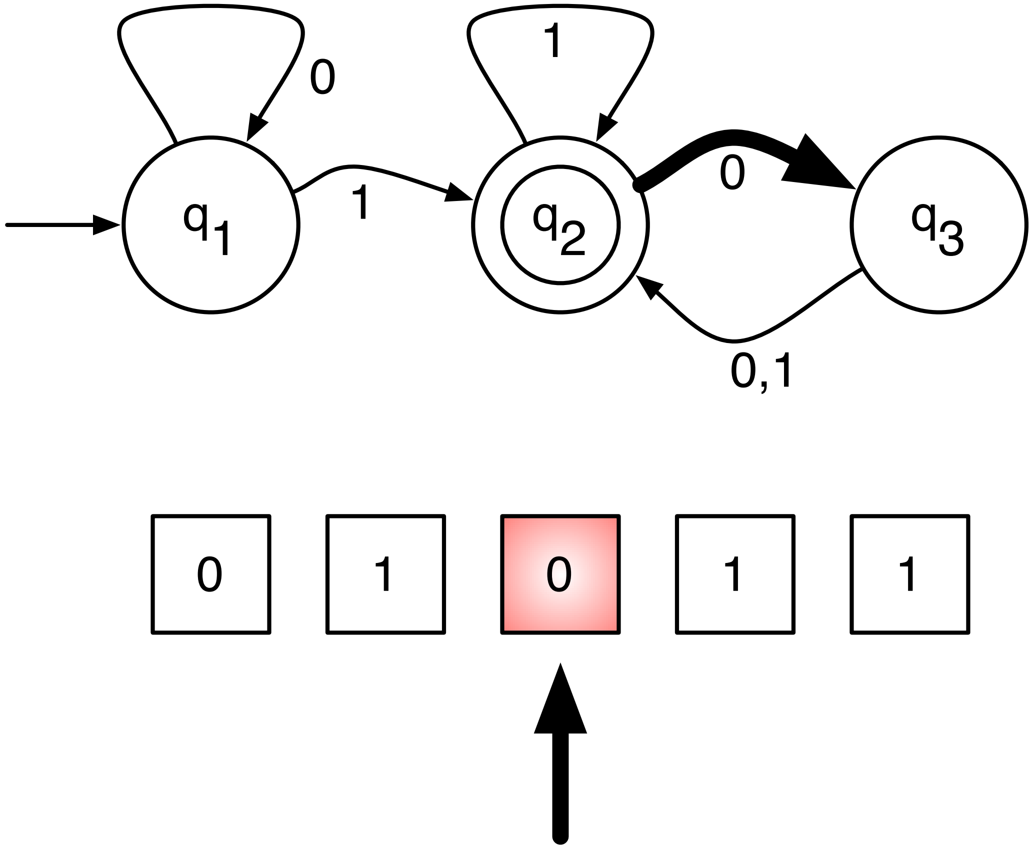
DFA: Ein Beispiel

Neuer Zustand: q_2



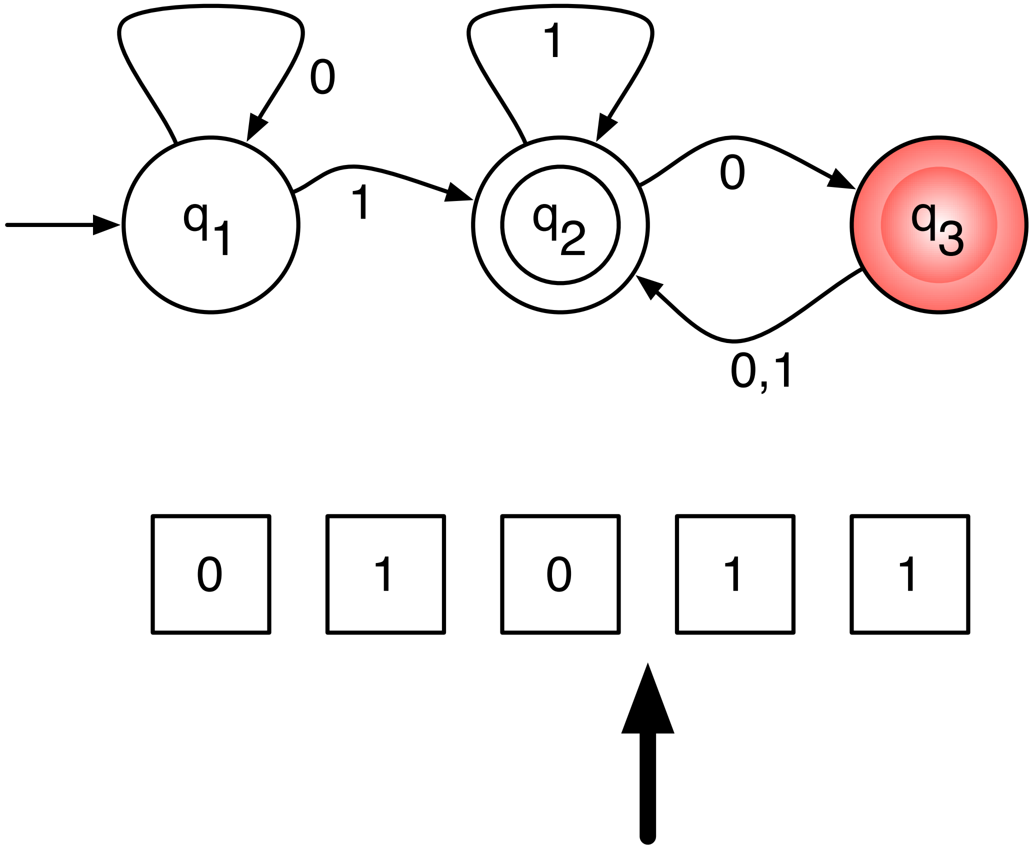
DFA: Ein Beispiel

3. Zeichen: 0



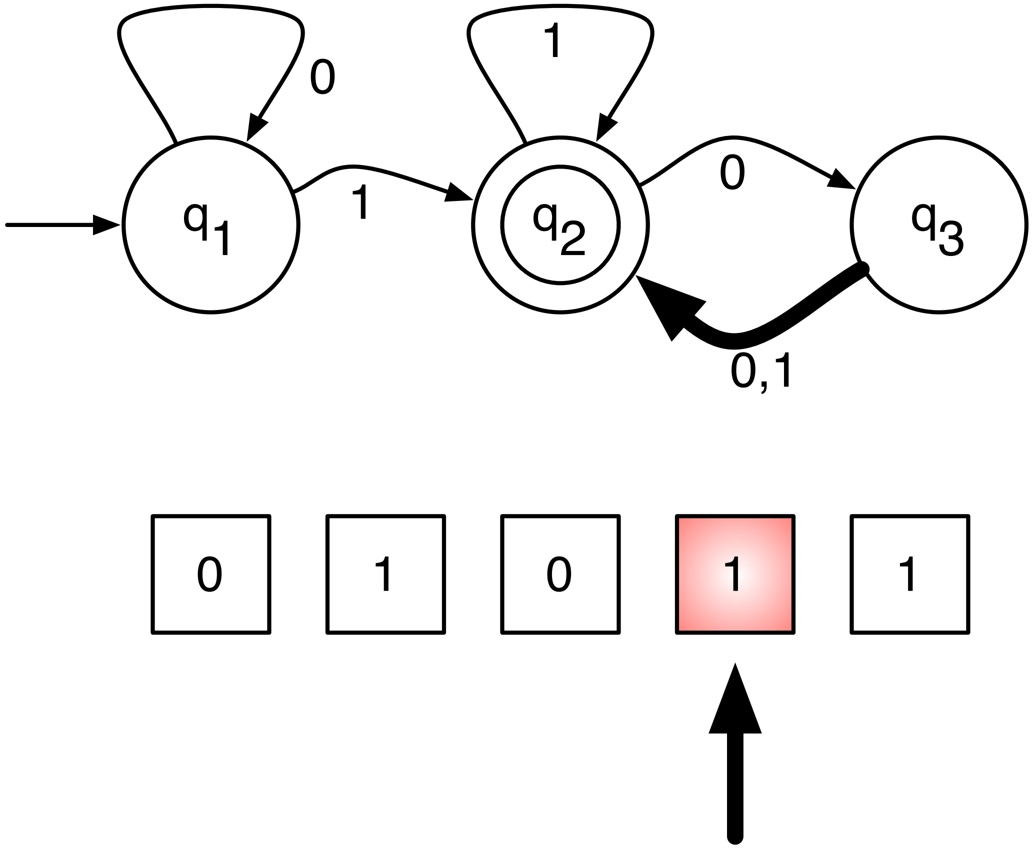
DFA: Ein Beispiel

Neuer Zustand: q_3



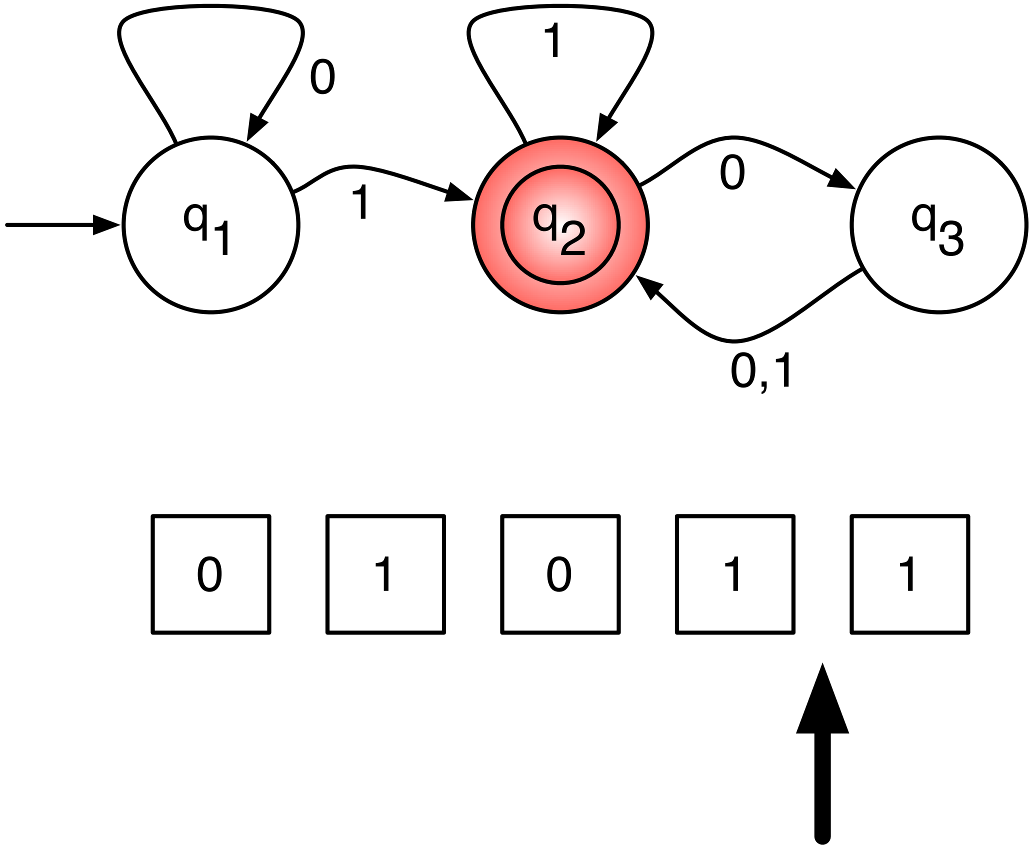
DFA: Ein Beispiel

Lesen von „1“



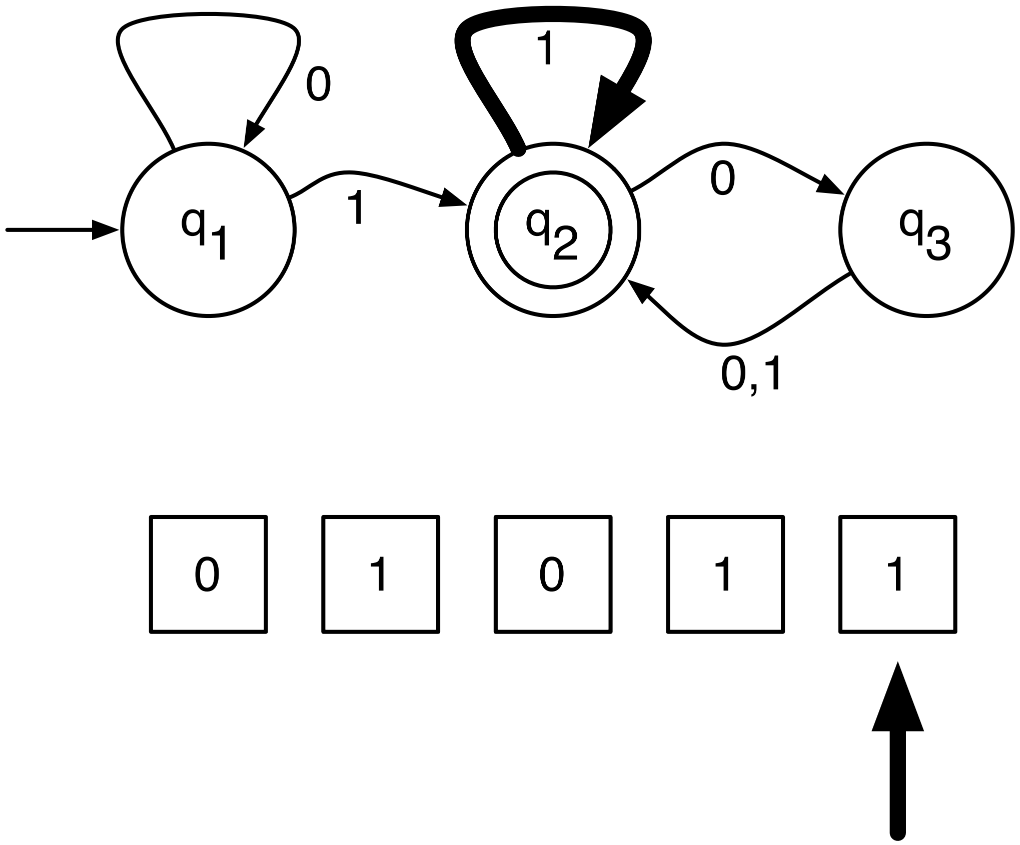
DFA: Ein Beispiel

Zustandsübergang zu q_2

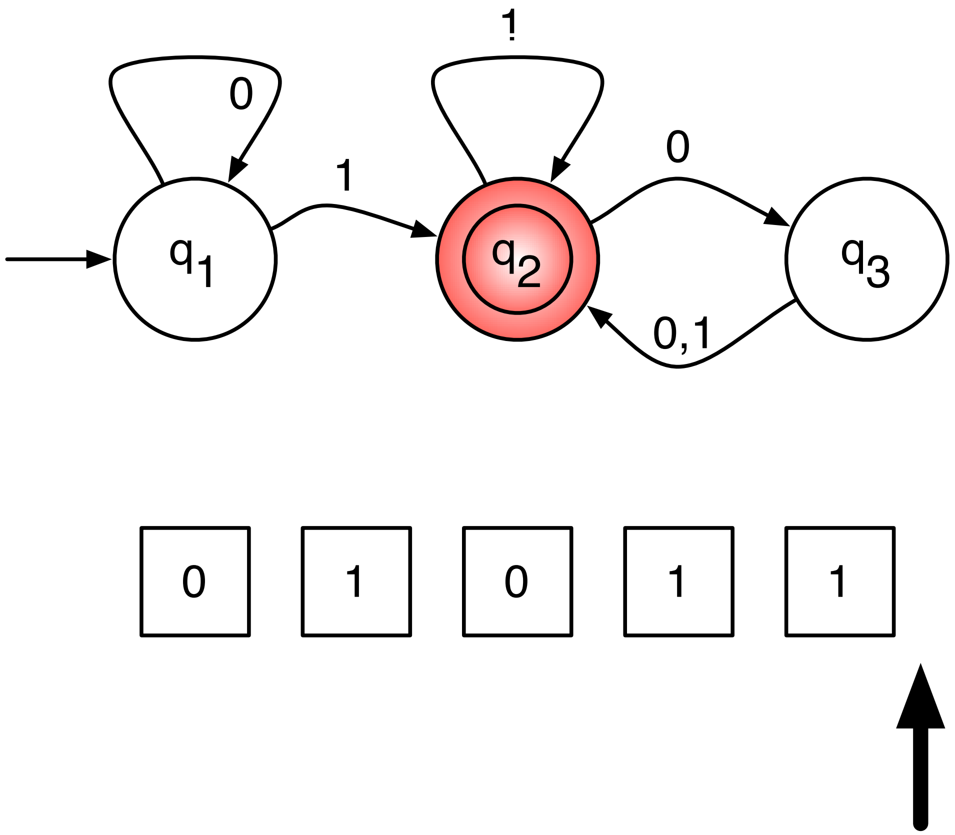


DFA: Ein Beispiel

Letztes Zeichen: „1“



DFA: Ein Beispiel
Wort „01011“ wird akzeptiert



universität freiburg

Worte & Sprachen

Formale Sprachen und
Endliche Automaten

Automaten akzeptieren Sprachen
Alphabete und Wörter

Alphabet

- Ein **Alphabet** Σ besteht aus einer endlichen Menge von Zeichen, z.B.
  - $\Sigma_1 = \{a, b, c\}$
  - $\Sigma_2 = \{0,1\}$
  - $\Gamma = \{0,1,x,y,z\}$

Wort

- Eine **Zeichenkette** (String/Wort) ist eine endliche Folge (Tupel) von Zeichen, z.B
  - w = abba
    - Notation: w_1=a, w_2=b, w_3=b, w_4=a
    - Die Länge eines Worts wird mit |w| beschrieben:
      - z.B. |w| = 4

Konkatenation

- Das Wort $w = x \circ y$ der Länge $|w| = |x| + |y|$
  - beschreibt die Konkatenation der Wörter x und y, d.h.
  - $w_i = x_i$ für $i = \{1,\ldots,|x|\}$
  - $w_{|x|+i} = y_i$ für $i = \{1,\ldots,|y|\}$
- Manchmal lassen wir $\circ$ weg, alternative Notation: $\|$

Menge aller Zeichenketten

- Σ^* bezeichnet die **Menge aller Zeichenketten** über Alphabet Σ
  - z.B.: „abba“ $\in$ {a,b}*
  - Die **leere Zeichenkette** wird mit ε bezeichnet.
    - Es gilt: $|\varepsilon| = 0$
- Eine Menge von Zeichenketten wird als **Sprache** bezeichnet

Mengen von Wörtern sind Sprachen
Beispiele

Wörter

- “010010”, “otto”, “abba”

Sprachen

- {abba, bab, aa}
- $\{w \in \{0,1\}^* \mid (\forall i \in \{1,\ldots,|w|\} : w_i = 0) \vee (\forall i \in \{1,\ldots,|w|\} : w_i = 1)\}$
  - = {ε, 0, 1, 00, 11, 000, 111, 0000, 1111, ...}
- Palindrom $= \{w \in \{a,b\}^* \mid \forall i \in \{1,\ldots,|w|\} : w_i = w_{|w|-i+1}\}$
  - $= \{\epsilon, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, \ldots\}$

ε & ∅

Kein Buchstabe versus kein Wort

Leere Sprache

- Sprache ohne Wörter = leere Menge: ∅
  - ∅ ist kein Wort, sondern eine Sprache

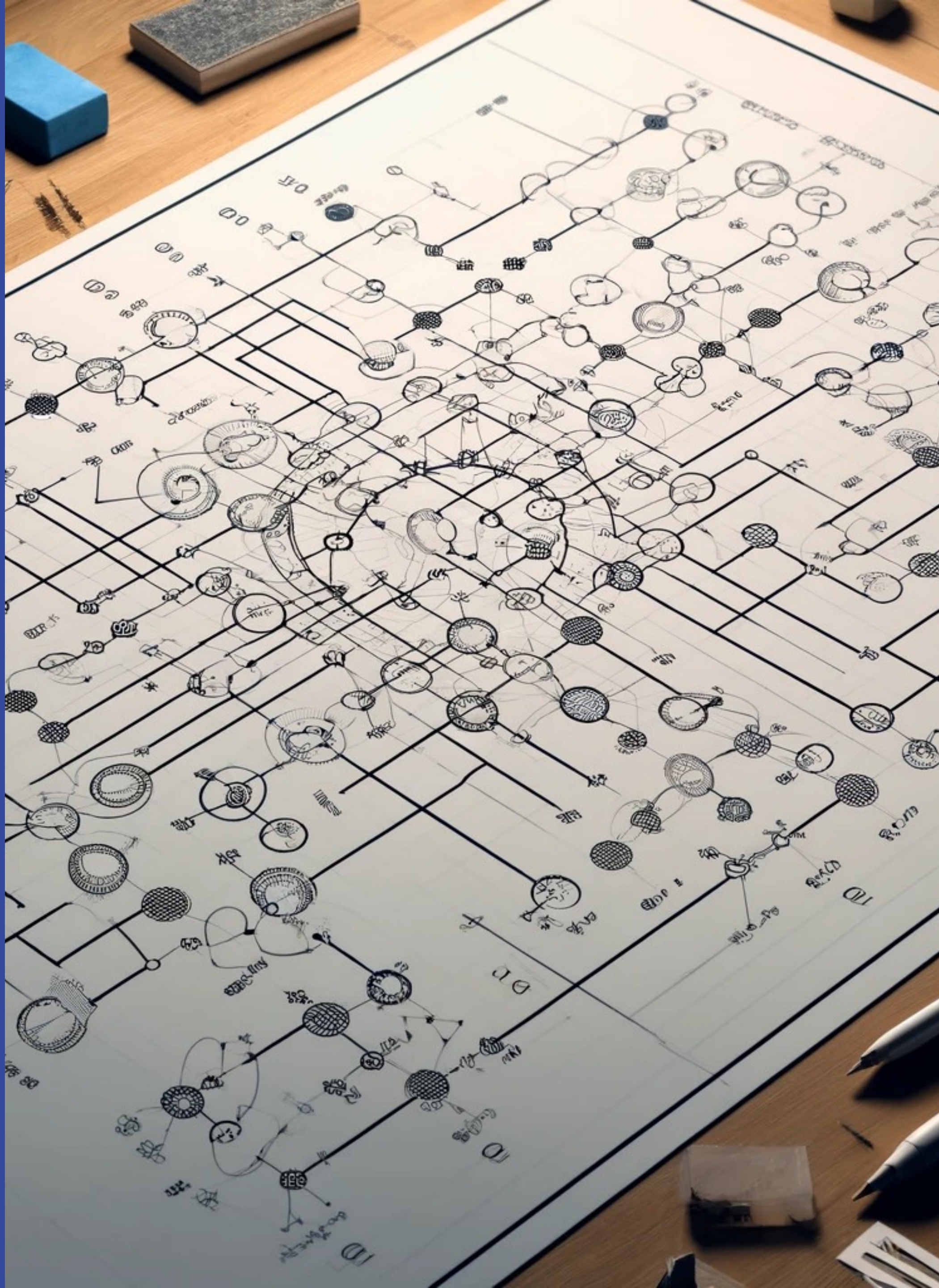
Leeres Wort

- Leeres Wort ε ist eine Zeichenfolge der Länge 0
  - ε ist **keine** Sprache
  - ε ist **kein** Zeichen
- Es gibt Sprachen,
  - die ε beinhalten;
  - die ε nicht beinhalten.

universität freiburg

DFA

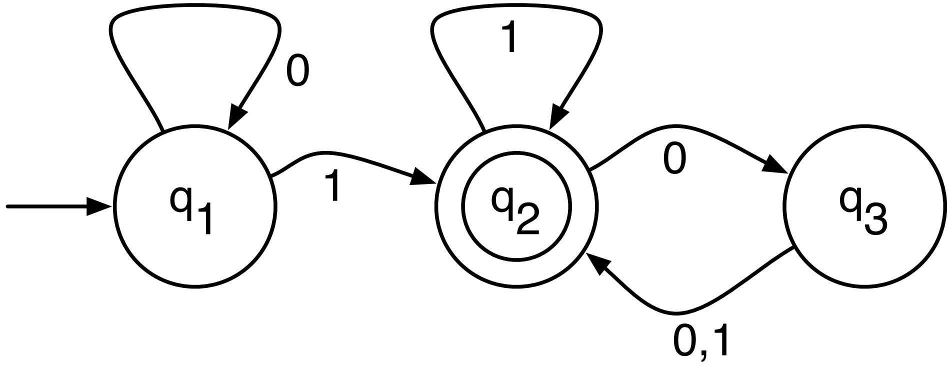
Formale Sprachen und
Endliche Automaten

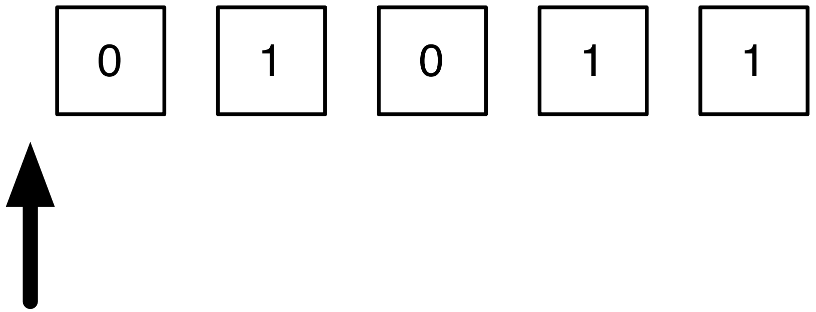


Komponenten des Beispielautomaten
Formale Beschreibung

- **Zustände** $Q = \{q_1, q_2, q_3\}$
- **Alphabet** $\Sigma = \{0,1\}$
- **Übergangsfunktion** δ:
  - $\delta(q_1,0) = q_1$
  - $\delta(q_1,1) = q_2$
  - $\delta(q_2,0) = q_3$
  - $\delta(q_2,1) = q_2$
  - $\delta(q_3,0) = q_2$
  - $\delta(q_3,1) = q_2$
- **Startzustand** $q_0 = q_1$
- **Akzeptierende Zustandsmenge** : $F = \{q_2\}$

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2





universität freiburg

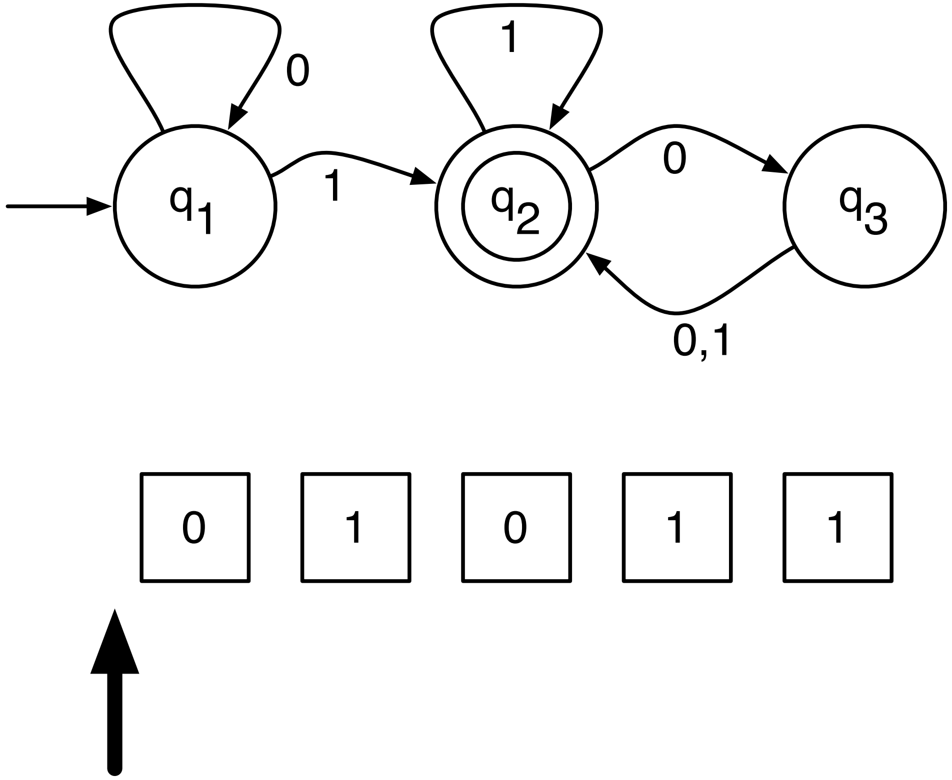
Komponenten des DFA
Formale Beschreibung

DFA

- Deterministischer Endlicher Automat = Deterministic Finite Automaton (DFA)

Definition

- Ein **deterministischer endlicher Automat M** wird durch die fünf Komponenten $Q, \Sigma, \delta, q_0, F$ beschrieben
  - Endliche **Zustandsmenge** Q
  - Endliches **Alphabet** Σ
  - **Übergangsfunktion** $\delta : Q \times \Sigma \to Q$
  - **Startzustand** $q_0 \in Q$
  - F ⊆ Q: ist die Menge der **akzeptierenden Zustände**



Die Berechnung eines endlichen Automaten (DFA)
Formale Definition

Definition

- Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein **endlicher Automat** und $w = w_1 w_2 \ldots w_n$ eine Zeichenkette über dem Alphabet Σ,
- dann **akzeptiert M das Wort w** genau dann, falls es ein Folge $r_0, r_1, \ldots, r_n \in Q$ von Zuständen gibt mit
  - $r_0 = q_0$
  - $\delta(r_i, w_{i+1}) = r_{i+1}$, für alle $i = 0, \ldots, n-1$
  - $r_n \in F$
- **M akzeptiert die Sprache A** falls
  - $A = \{w \in \Sigma^* \mid M \text{ akzeptiert } w\}$

universität freiburg

Reguläre Sprachen und Operationen

$$A \cup B = \{wadda, hadda, du, da\}$$

$$A \circ B = \{waddadu, waddada, had$$

$$B^* = \{\epsilon, du, da, dudu, duda, dadu,$$

Formale Sprachen und
Endliche Automaten

Reguläre Sprachen
Formale Definition

Definition

- Eine **Sprache** heißt **regulär**, falls ein endlicher Automat sie akzeptiert.

- Wir schreiben auch **REG** für die Klasse (Menge) aller regulären Sprachen

Reguläre Operationen
Formale Definition

Definition

- Die regulären Operationen **Vereinigung**, **Konkatenation** und **Stern** werden wie folgt definiert
- **Vereinigung**:
  - $A \cup B = \{w \mid w \in A \text{ oder } w \in B\}$
- **Konkatenation**
  - $A \circ B = \{x \circ y \mid x \in A \text{ und } y \in B\}$
- **Stern (Kleene-Operator)**
  - $A^* = \{x_1 \circ x_2 \circ \ldots \circ x_k \mid k \geq 0 \text{ und } \forall i \in \{1,2,\ldots,k\} : x_i \in A\}$

Frage

- Wie ist der Zusammenhang zwischen regulären Operationen und regulären Sprachen?
  - Wird im Laufe der Veranstaltung geklärt.

Reguläre Operationen
Beispiel

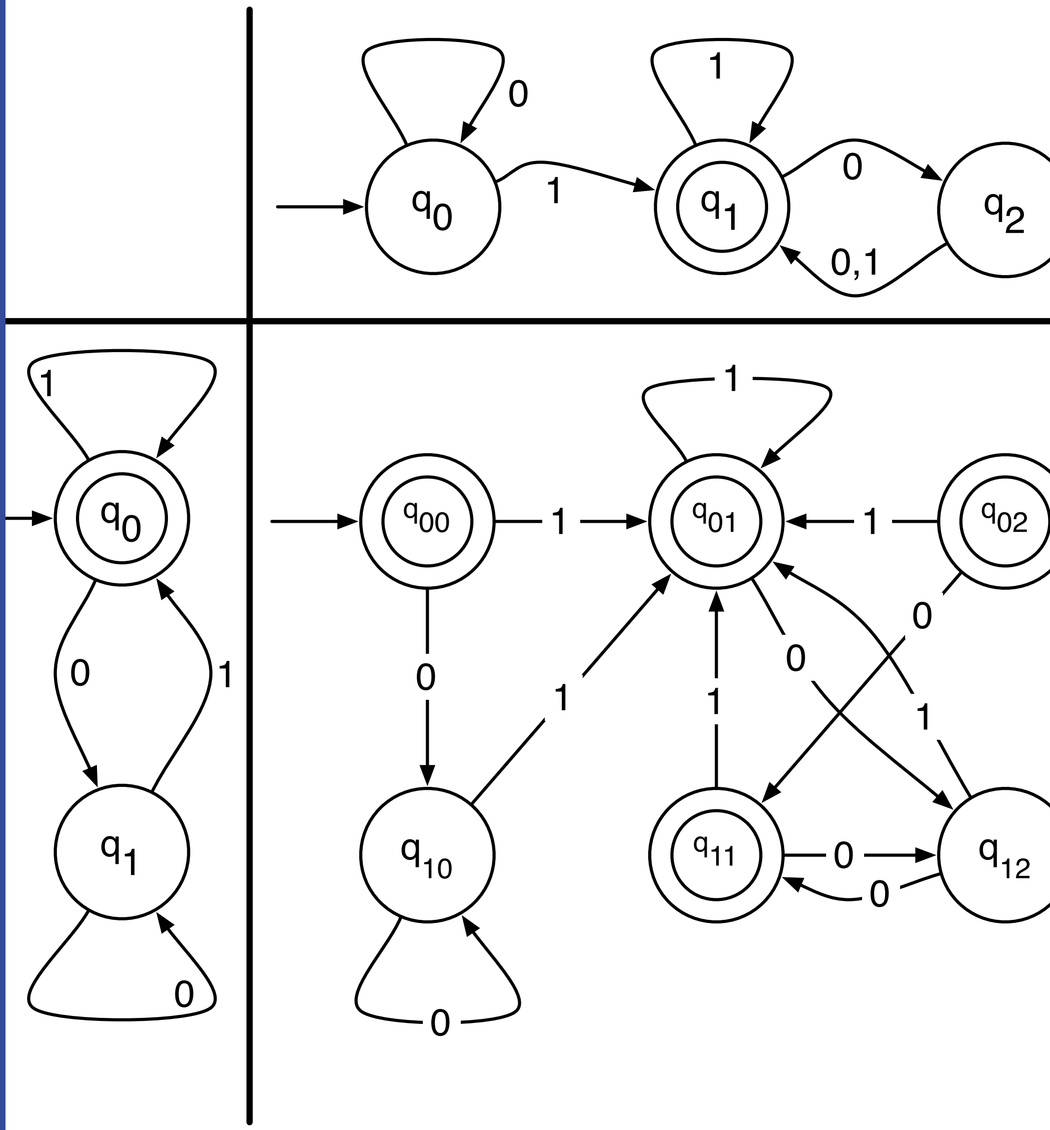
Ein Alphabet und zwei endliche Mengen

- Sei $\Sigma = \{a, b, c, d, \ldots, z\}$
- $A = \{wadda, hadda\}$
- $B = \{du, da\}$

- Vereinigung
  - $A \cup B = \{wadda, hadda, du, da\}$
- Konkatenation
  - $A \circ B = \{waddadu, waddada, haddadu, haddada\}$
- Stern
  - $B* = \{\epsilon, da, du, dada, dadu, duda, dudu, dadada, \ldots\}$

universität freiburg

Abschluss von REG unter Vereinigung



Formale Sprachen und Endliche Automaten

Reguläre Operationen
Abschluss unter Vereinigung

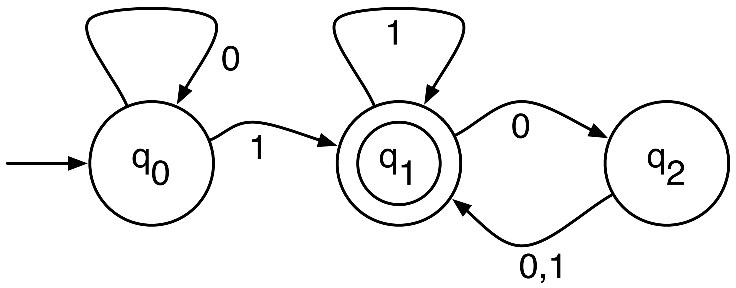
Theorem

- Die Klasse der regulären Sprachen ist **abgeschlossen unter der Vereinigung**:
- Falls A und B regulär ist, dann ist auch $A \cup B$ regulär

D.h.

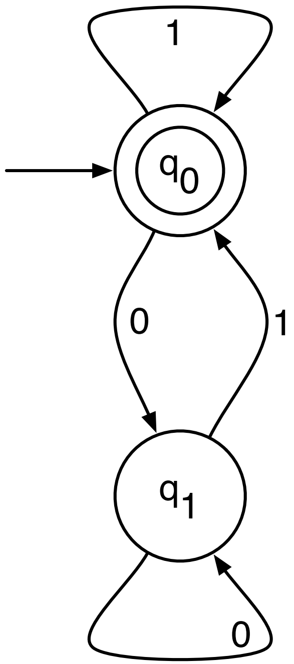
- Gegeben ein endlicher Automat für A und
- gegeben ein endlicher Automat für B,
  - dann gibt es auch einen endlichen Automaten für $A \cup B$

Abschluss unter Vereinigung
Beweisidee



Gegeben

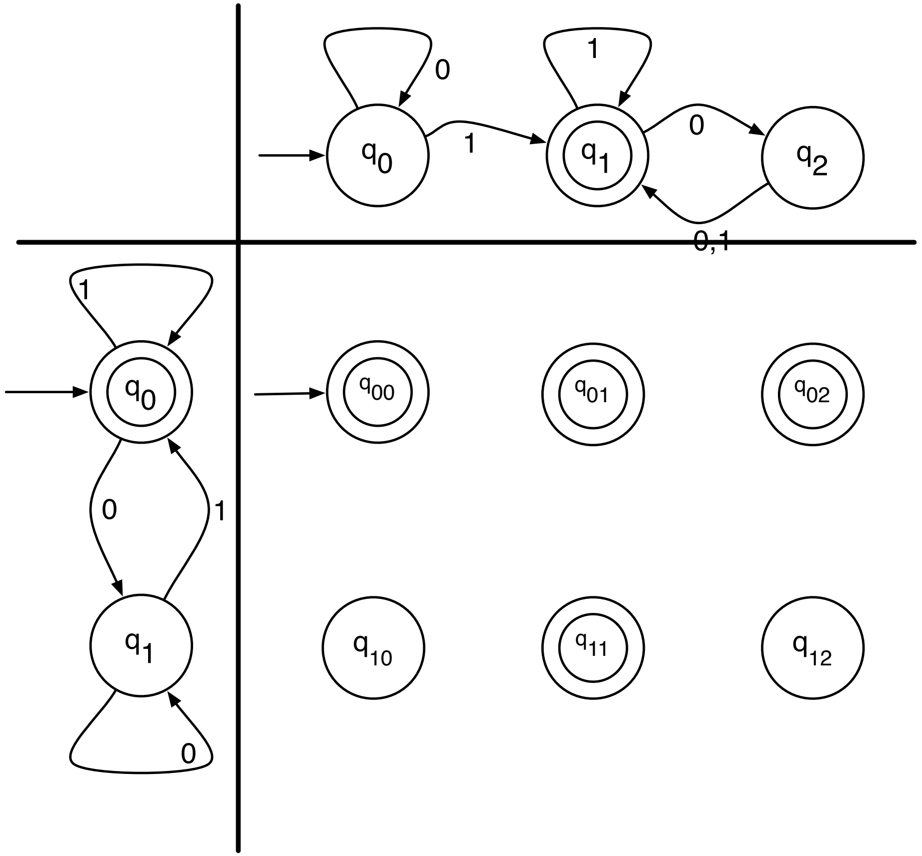
- zwei Automaten und die Eingabe **0110**
- Simuliere mit **zwei Fingern** die Automaten gleichzeitig
- Falls **einer der „Finger“ akzeptiert, akzeptiere**



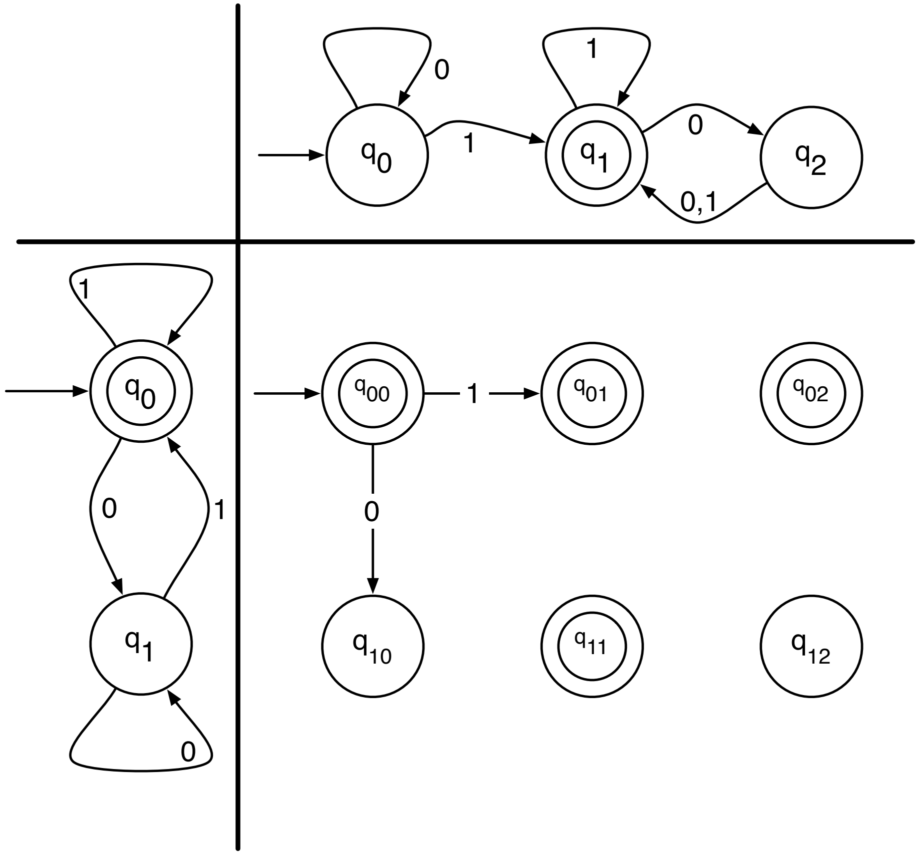
Idee

- Ersetze zwei Finger durch mehr Zustände

Die Konstruktion des kartesischen Produkts
für den Abschluss unter Vereinigung

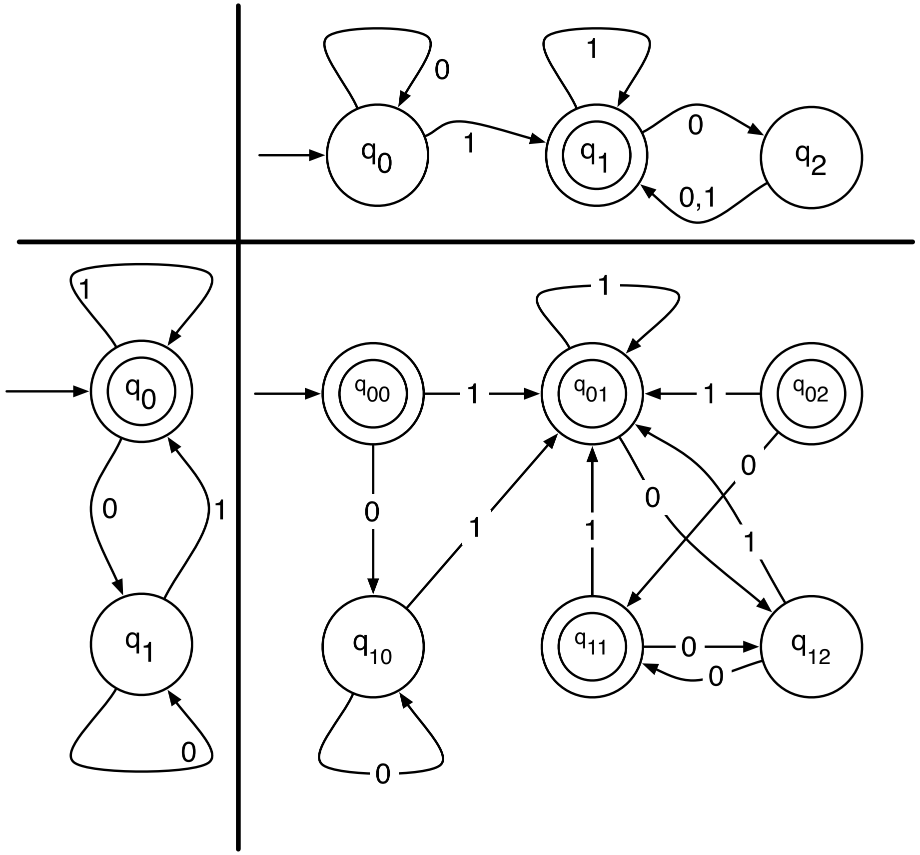


Ein Zustandsübergang
des kartesischen Produktautomaten



Automat für die vereinigte Sprache

Kartesischer Produktautomat



Die formale Konstruktion
Kartesischer Produktautomat

Theorem

- Gegeben sei ein endlicher Automat $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ für A_1 und
- gegeben sei ein endlicher Automat $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ für A_2
- dann gibt es auch einen endlichen Automaten für $A_1 \cup A_2$

Beweis

- Wir konstruieren einen Automaten $M = (Q, \Sigma, \delta, q_0, F)$ der $A_1 \cup A_2$ erkennt mit
  1. **Zustandsmenge**
     $Q = \{(r_1, r_2) \,|\, r_1 \in Q_1 \text{ und } r_2 \in Q_2\}$,
     Q ist das kartesische Produkt aus Q1 und Q2
  2. **Alphabet** Σ bleibt gleich
     - Wir erlauben uns diese Vereinfachung, dass die Ausgangsalphabete gleich sind.Eine kleine Modifikation würde den anderen Fall auch beweisen
  3. **Übergangsfunktion** δ
     Für alle $r_1 \in Q_1$ und $r_2 \in Q_2$ und $a \in \Sigma$ gelte
     $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$
  4. **Anfangszustand**: $q_0 = (q_1, q_2)$
  5. **Akzeptierende Zustände**:
     $F = \{(r_1, r_2) \;|\; r_1 \in F_1 \text{ oder } r_2 \in F_2\}$

Der formale Beweis
Abschluss unter Vereinigung

Theorem

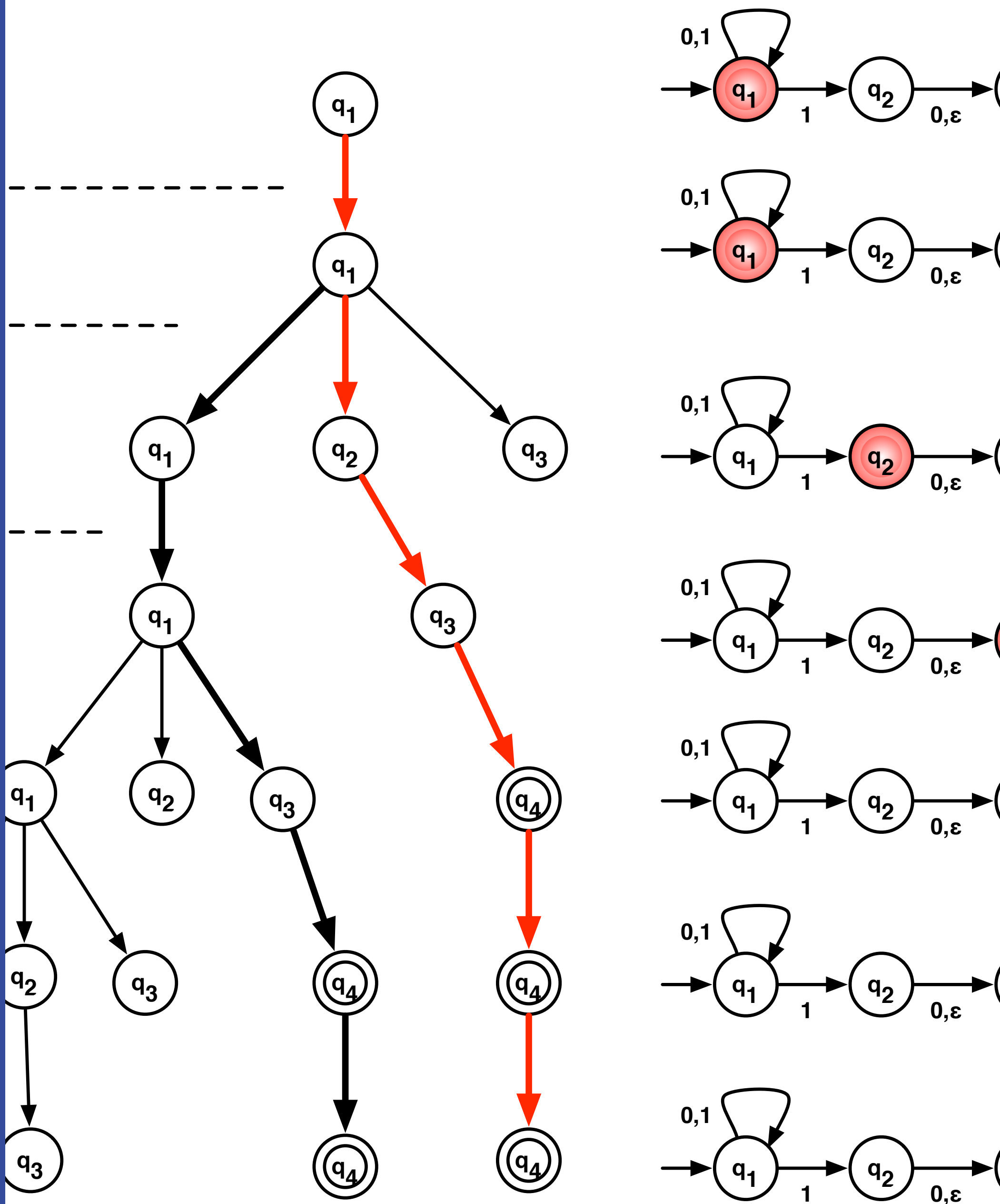
- Gegeben sei ein endlicher Automat $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ für A_1 und
- gegeben sei ein endlicher Automat $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ für A_2
- dann gibt es auch einen endlichen Automaten für $A_1 \cup A_2$

Beweis (Fortsetzung)

- Der Rest des Beweises erfolgt durch eine Induktion über die Länge n der Wörter
- **Induktionsanfang**: Maschine ist in $q_0 = (q_1, q_2)$ für ε
- **Induktionsannahme**: Maschine ist in Zustand $(\delta_1(q_1, w), \delta_2(q_2, w))$ nach Abarbeiten von w
    - Erweiterung der Definition der Übergangsfunktion: $\delta_1(q, wa) = \delta_1(\delta_1(q, w), a)$ und ebenso für δ_2
- **Induktionsschluss**: Maschine ist in Zustand $(\delta_1(q_1, wa), \delta_2(q_2, wa))$ für w a, wobei $a \in \Sigma$ und $w \in \Sigma^n$
    - Induktionsschritt: Führe eine Transition aus
- Der **Beweis folgt dann aus**: $F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ oder } r_2 \in F_2\}$
    - d.h. die Maschine akzeptiert, wenn das Wort w in A_1 oder A_2 war
    - weil die beiden Maschinenzustände in den Paaren des kartesischen Produkts immer korrekt abgebildet werden.

universität freiburg

NFA
Non Deterministic Finite Automaton
Nicht-deterministischer endlicher Automat



Formale Sprachen und
Endliche Automaten

Die Konkatenation
Abschluss

Frage

- Sind endliche Automaten auch unter Konkatenation abgeschlossen?

1. Idee

- Automat M_1 akzeptiert Sprache A_1
- Automat M_2 akzeptiert Sprache A_2
  - Konstruiere Automat M, der aus den Zuständen von M_1 und M_2 besteht ... ? ....
- Nach längeren Probieren:
  - Idee funktioniert nicht mit DFA!!!!

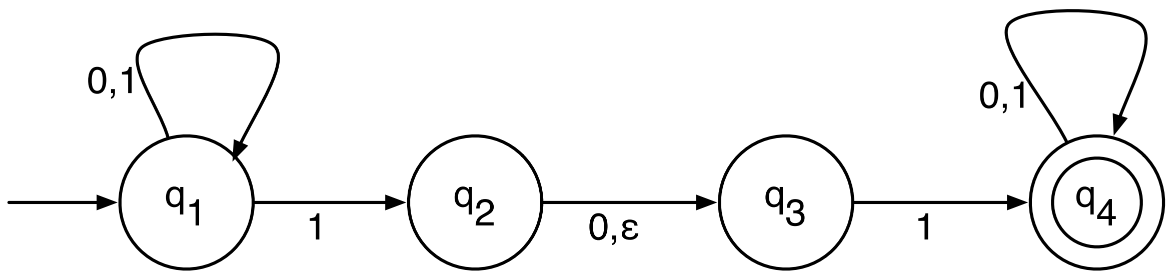
Lösung:

- Betrachte NFA

Ein nichtdeterministischer Automat
Beispiel eines NFA

Was ist anders als beim DFA?

- Von q_1 gehen zwei Übergänge mit 1 aus
- Von q_2 geht ein ε-Übergang nach q_3
- Von q_2 fehlt der Übergang bei Zeichen 1
- Von q_3 fehlt der Übergang bei Zeichen 0

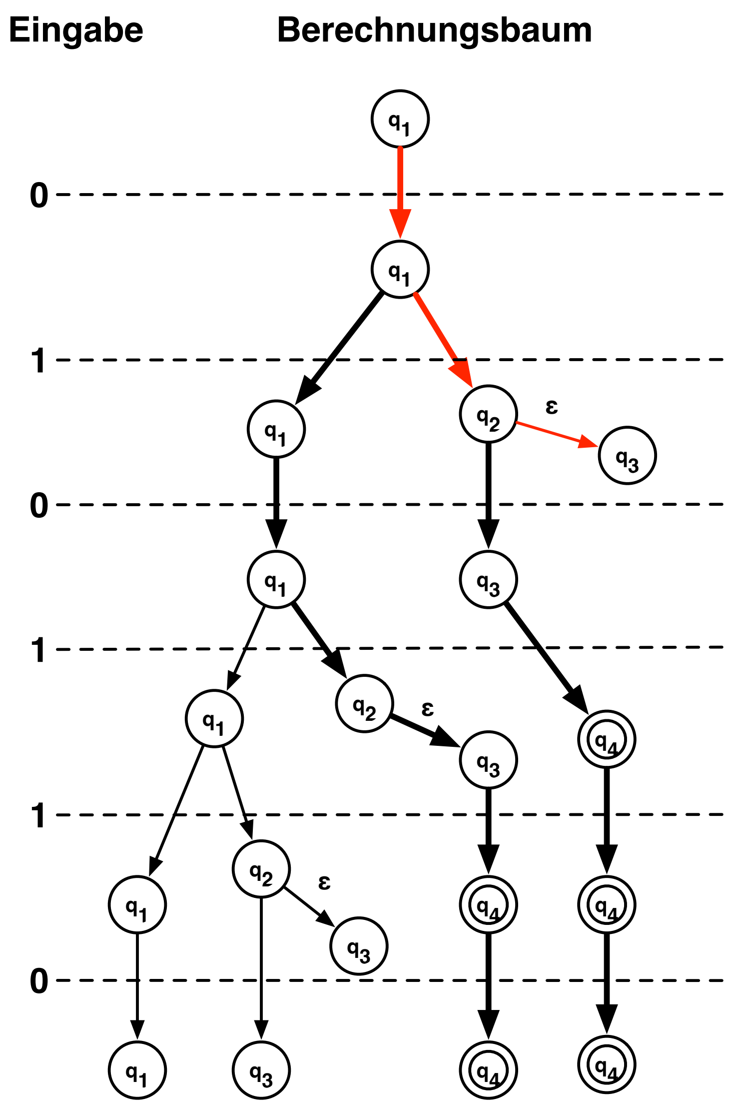


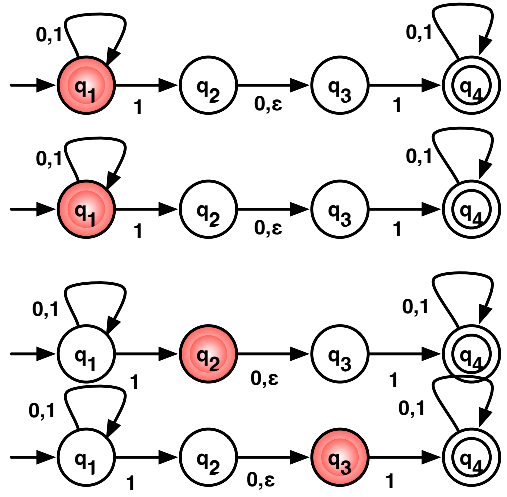
Frage

- Was und wann akzeptiert so ein Automat?

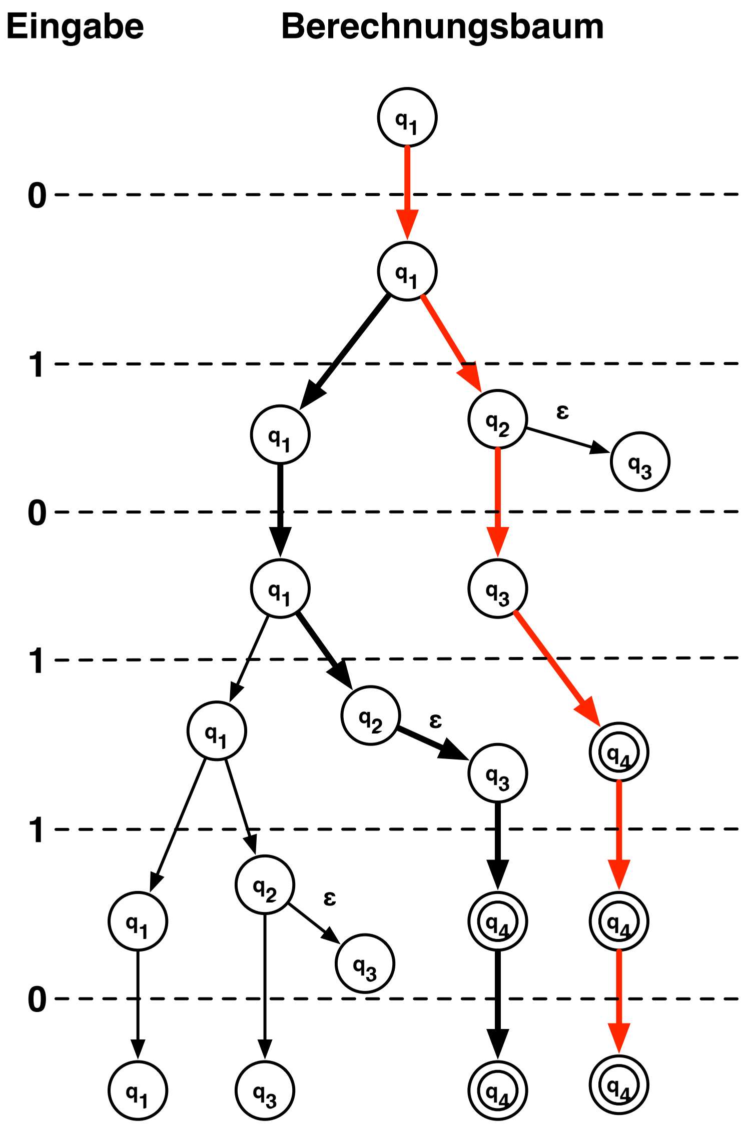
Nicht-akzeptierende Berechnung eines NFA

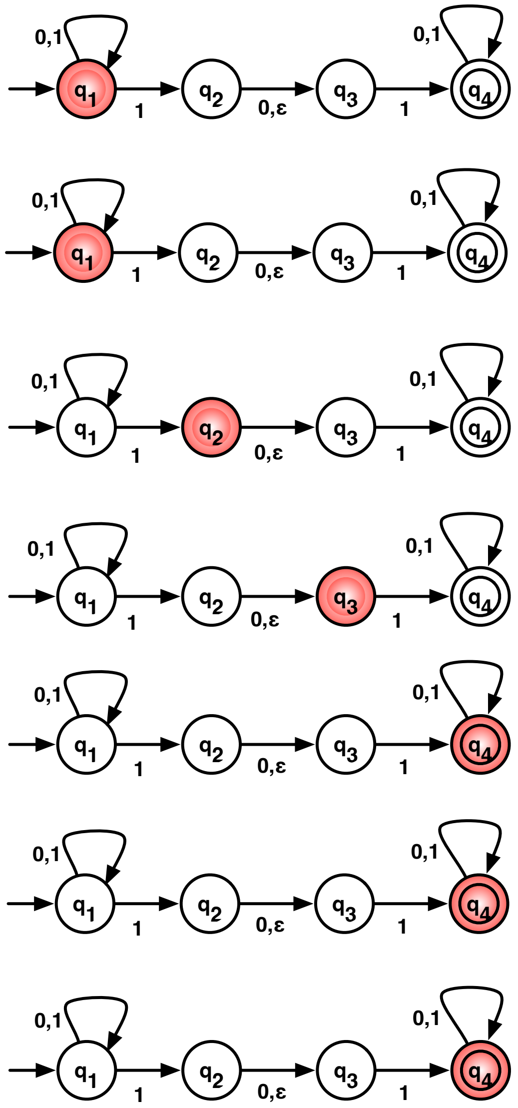
Beispiel eines NFA



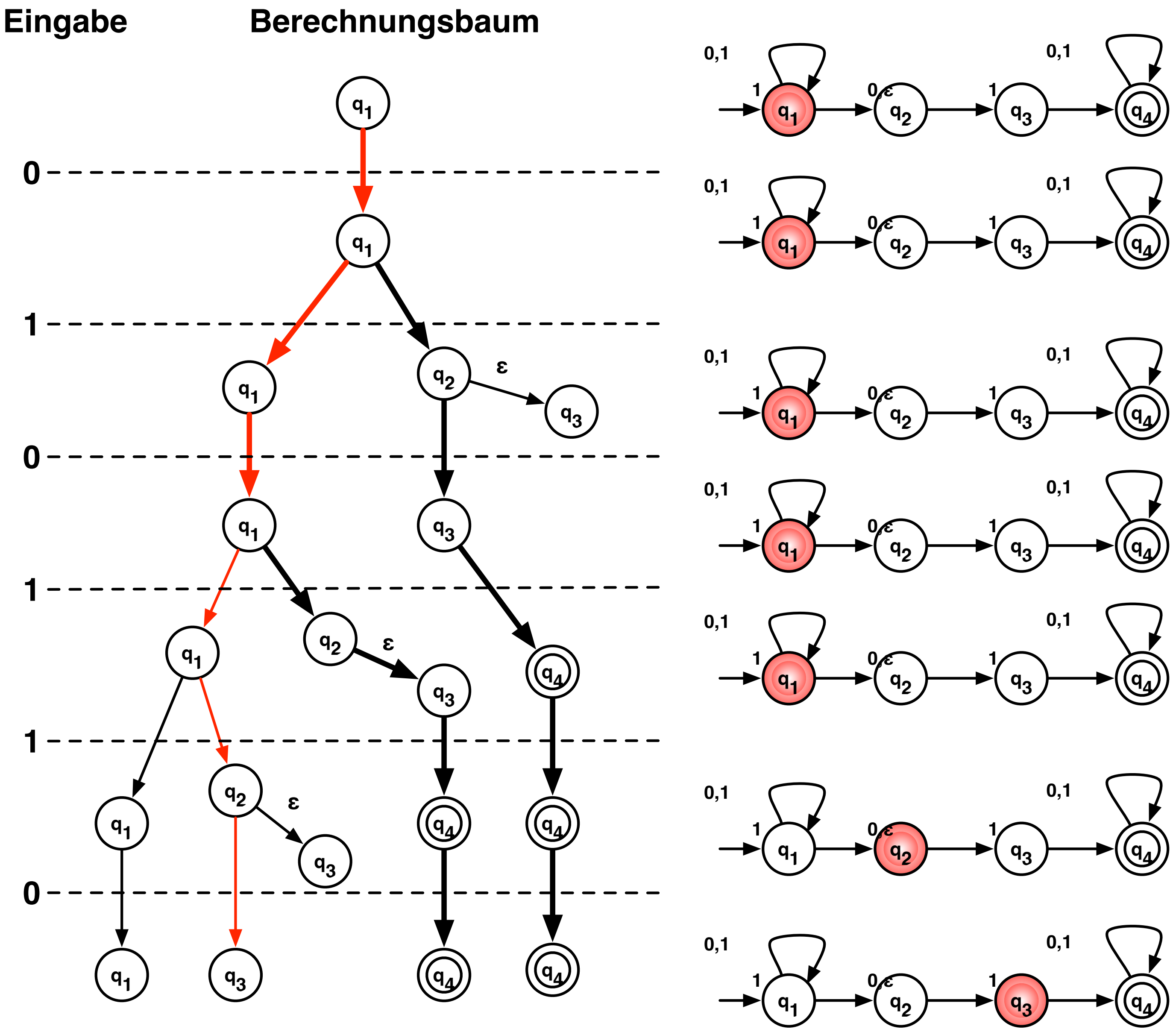


Akzeptierende Berechnung eines NFA
Beispiel eines NFA



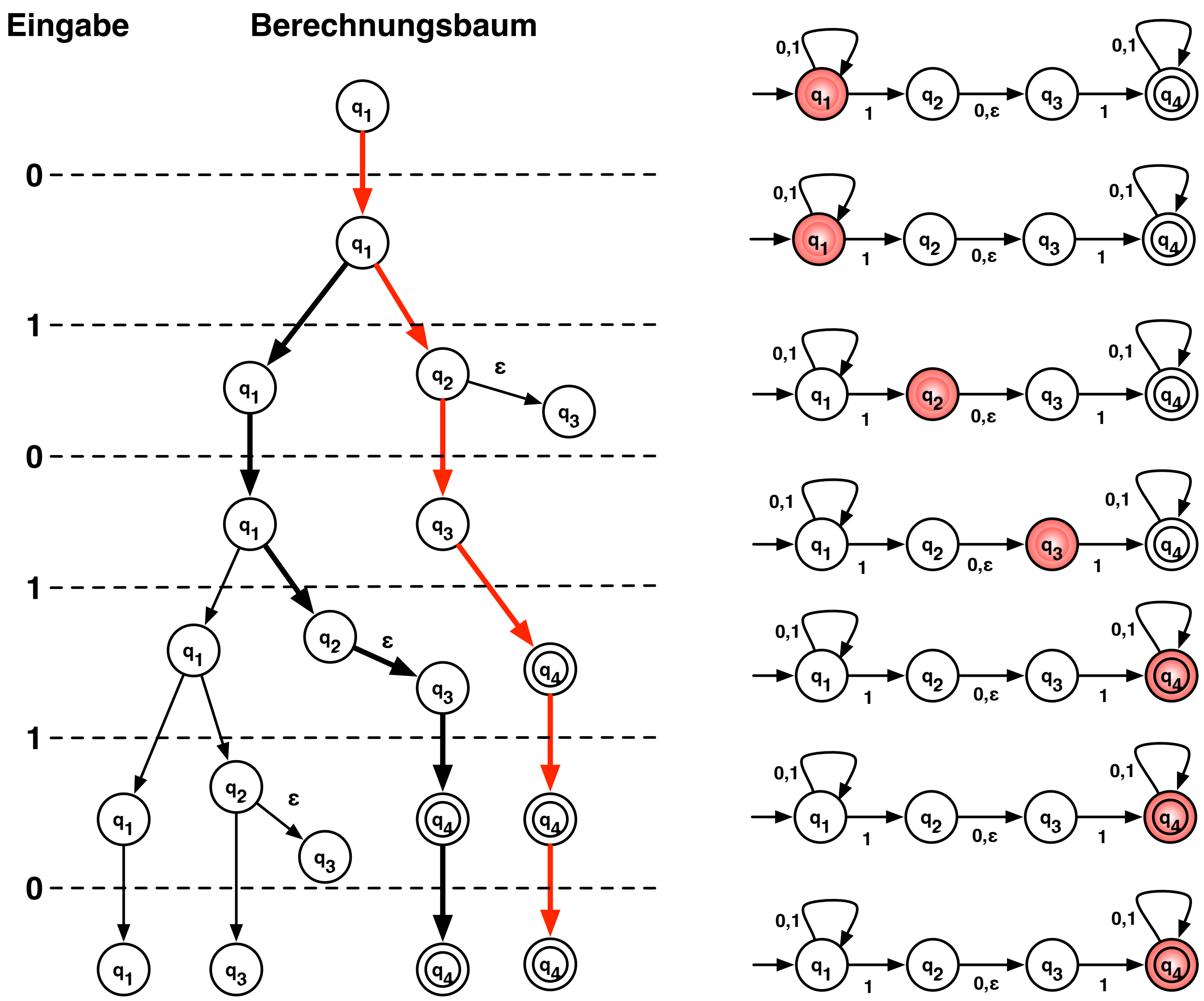


Nicht akzeptierende Berechnung
Beispiel eines NFA



Wann akzeptiert ein NFA?

Berechnungsbaum



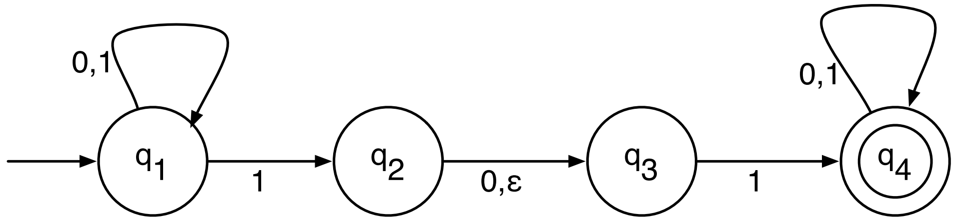
Idee

- Ein nicht-deterministischer endlicher Automat (NFA - Nondeterministic Finite Automaton)
- **akzeptiert ein Wort,**
- falls es (unter allen möglichen) **mindestens eine akzeptierende Berechnung** gibt.

Komponenten eines NFA
Beispiel

1. **Zustände** $Q = \{q_1, q_2, q_3, q_4\}$
2. **Alphabet** $\Sigma = \{0,1\}$
3. **Übergangsfunktion** δ
   - $\delta(q_1,0) = \{q_1\}$
   - $\delta(q_1,1) = \{q_1, q_2\}$
   - $\delta(q_1, \varepsilon) = \emptyset$
   - $\delta(q_2,0) = \{q_3\}$
   - $\delta(q_2,1) = \emptyset$
   - $\delta(q_2, \varepsilon) = \{q_3\}$
   - $\delta(q_3,0) = \emptyset$
   - $\delta(q_3,1) = \{q_4\}$
   - $\delta(q_3, \varepsilon) = \emptyset$
   - $\delta(q_4,0) = \{q_4\}$
   - $\delta(q_4,1) = \{q_4\}$
   - $\delta(q_4, \varepsilon) = \emptyset$
4. **Startzustand** $q_0 = q_1$
5. **Akzeptierende Zustandsmenge**
   $F = \{q_4\}$

δ	0	1	ε
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	$\emptyset$



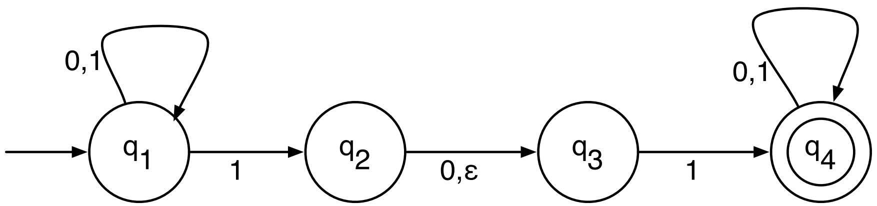
Komponenten eines NFA
Formale Definition

Definition

- Ein **nichtdeterministischer endlicher Automat (NFA)** wird durch das 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ beschrieben
  - Q: Eine endliche Menge von **Zuständen**
  - Σ: ist das **Alphabet**
  - $\delta : Q \times \Sigma_\varepsilon \to P(M)$ ist die **Übergangsfunktion**
  - $q_0 \in Q$: ist der **Startzustand**
  - F ⊆ Q: ist die Menge der **akzeptierenden Zustände**

Notation

- Sei $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$
- $P(M) = 2^M$ ist die **Potenzmenge von M**
  - z.B. $P(\{a,b\}) = \{\emptyset, \{a\}, \{b\}, \{a,b\}\}$
  - d.h. die Menge aller möglichen Teilmengen von M: $P(M) := \{L \mid L \subseteq M\}$



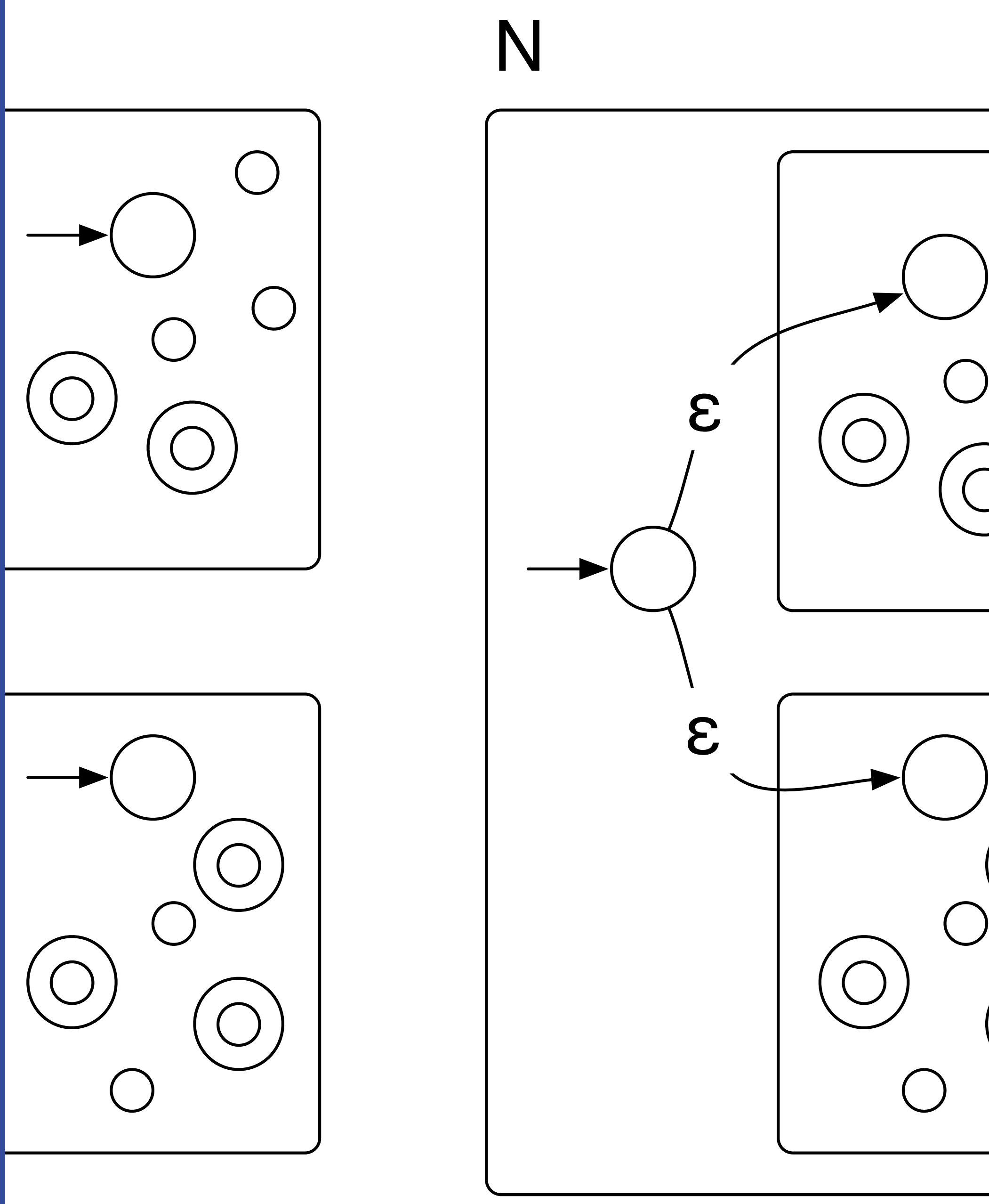
Berechnung eines NFA
Akzeptanz eines Worts

Definition

- Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein NFA. Falls es eine
  - Darstellung der Eingabe $w = v_1 v_2 \ldots v_m$ aus Σ_ε und
  - eine Folge $r_0, r_1, \ldots, r_m$ von Zuständen aus Q gibt, wobei
    - $r_0 = q_0$,
    - $r_{i+1} \in \delta(r_i, v_{i+1})$ für alle $i \in \{0,...,m-1\}$, und
    - $r_m \in F$
- dann **akzeptiert M das Wort.**

universität freiburg

L(NFA)=L(DFA)



Formale Sprachen und
Endliche Automaten

$L(DFA) \subseteq L(NFA)$
Der einfache Fall

Theorem

- Jeder deterministische endliche Automat hat einen äquivalenten nichtdeterministischen Automaten.

Beweis

- Jeder deterministische endliche Automat ist *(praktisch)* ein nichtdeterministischer endlicher Automat.

L(NFA) ⊆ L(DFA)
Beweisanfang

Theorem

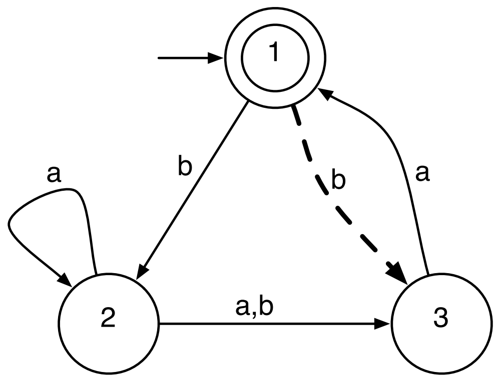
- Jeder nichtdeterministischer endliche Automat hat einen äquivalenten deterministischen Automat.

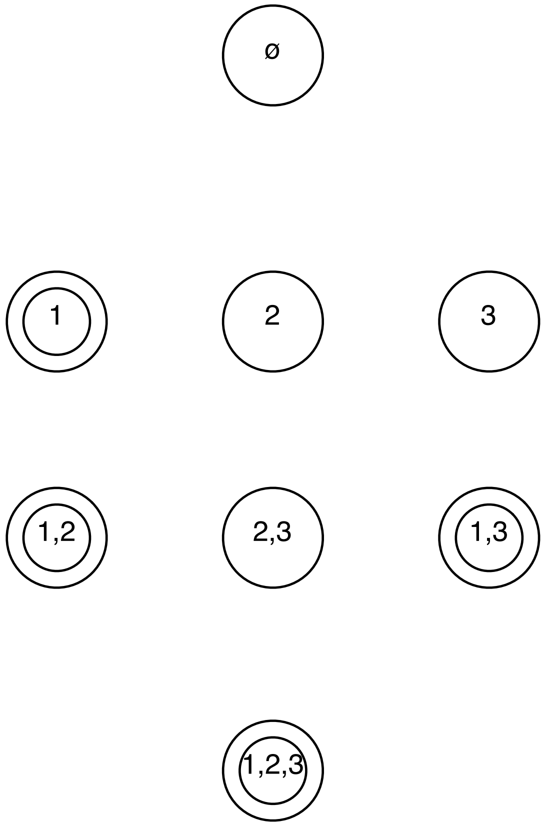
Beweis:

- Gegeben sei ein nichtdeterministischer endlicher Automat $N = (Q, \Sigma, \delta, q_0, F)$
- dann konstruieren wir den DFA $M = (Q', \Sigma, \delta', q_0', F')$ wie folgt:
- **1. Fall:** kein ε-Übergang in δ
  - **Zustandsmenge**: $Q' = P(Q)$, Q' ist die Potenzmenge von Q
- Alphabet bleibt gleich
- **Übergangsfunktion**:
  Für alle R ∈ Q‘ und a ∈ Σ gelte
  $\delta'(R, a) = \{q \in Q \mid \exists r \in R : q \in \delta(r, a)\}$
  - andere Notation: $\delta'(R, a) = \bigcup_{r \in R} \{\delta(r, a)\}$
- **Anfangszustand**: $q_0 = \{q_0\}$
- **Akzeptierende Zustände**
  - Zustand R akzeptiert, falls ein akzeptierender Zustand von F in R ist
  - $F' = \{R \in Q' \mid \exists r \in R : r \in R\}$

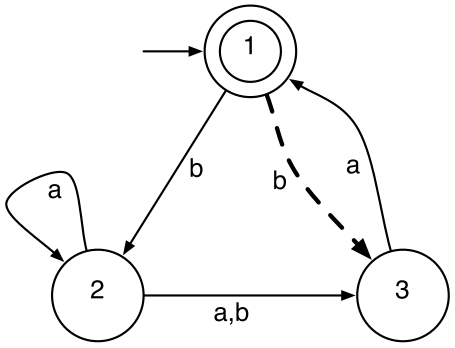
Potenzmenge der Zustände
Ein Beispiel

- **Zustandsmenge**: Q' = P(Q)
  - Q' ist die Potenzmenge von Q

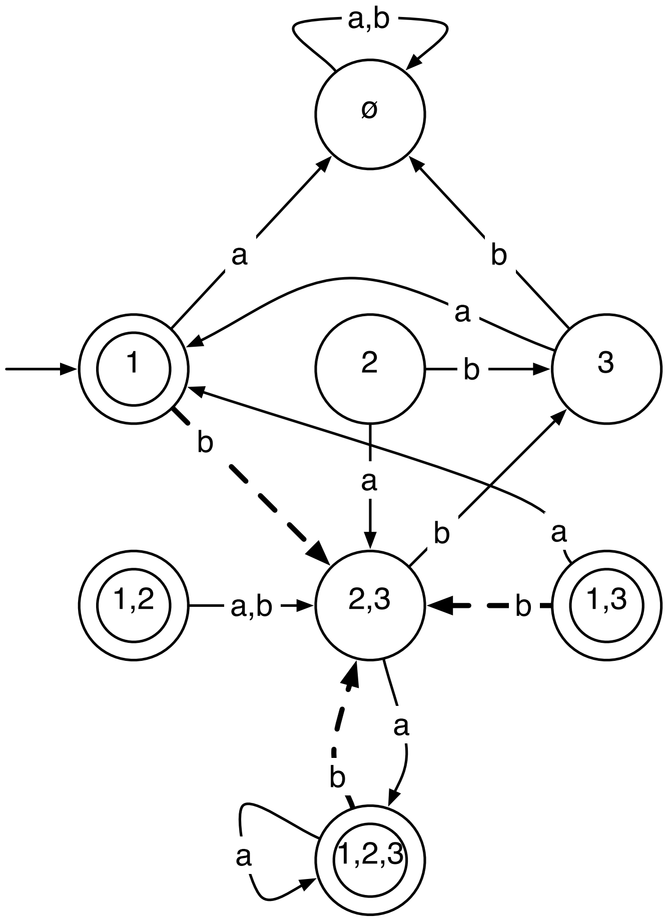




Ohne ε-Übergang
Potenzautomat



- **Übergangsfunktion**:
  Für alle R ∈ Q‘ und a ∈ Σ gelte
  $\delta'(R, a) = \{q \in Q \mid \exists r \in R : q \in \delta(r, a)\}$
  - andere Notation: $\delta'(R, a) = \bigcup_{r \in R} \{\delta(r, a)\}$
- **Anfangszustand**: $q_0 = \{q_0\}$
- **Akzeptierende Zustände**
  - Zustand R akzeptiert, falls ein akzeptierender Zustand von F in R ist
  - $F' = \{R \in Q' \mid \exists r \in R : r \in R\}$

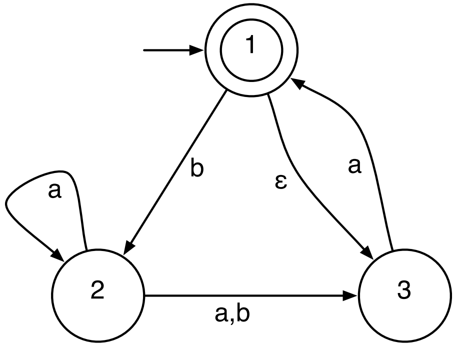


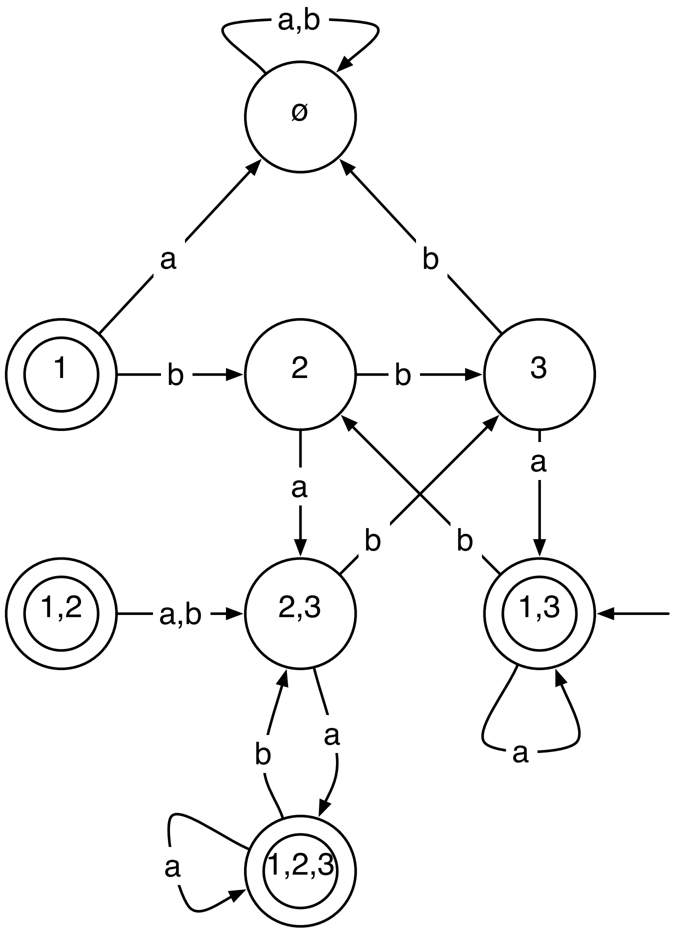
L(NFA) ⊆ L(DFA)
Mit ε-Übergang

- Gegeben sei ein nichtdeterministischer endlicher Automat $N = (Q, \Sigma, \delta, q_0, F)$
- dann konstruieren wir den DFA $M = (Q', \Sigma, \delta', q_0', F')$ wie folgt:
- Notation:
  $E(R) = \{q \mid q \text{ wird von } R \text{ durch } \epsilon\text{-Übergänge erreicht}\}$
- **2. Fall: Mit ε-Übergang in δ**
  - **Zustandsmenge**: Q' = **P**(Q),
    - Q' ist die Potenzmenge von Q
- **Übergangsfunktion** δ'
  - Für alle $R \in Q'$ und $a \in \Sigma$ gelte
    $\delta'(R, a) = \{q \in Q \mid \exists r \in R : q \in E(\delta(r, a))\}$
    andere Notation:
  - $$\delta'(R, a) = \bigcup_{r \in R} E(\{\delta(r, a)\})$$
- **Anfangszustand**: $q_0 = E(\{q_0\})$
- **Akzeptierende Zustände:**
  - Zustand R akzeptiert, falls ein akzeptierender Zustand von F in R ist
  - $F' = \{R \in Q' \mid \exists r \in R : r \in R\}$

L(NFA) ⊆ L(DFA)
Mit ε-Übergang

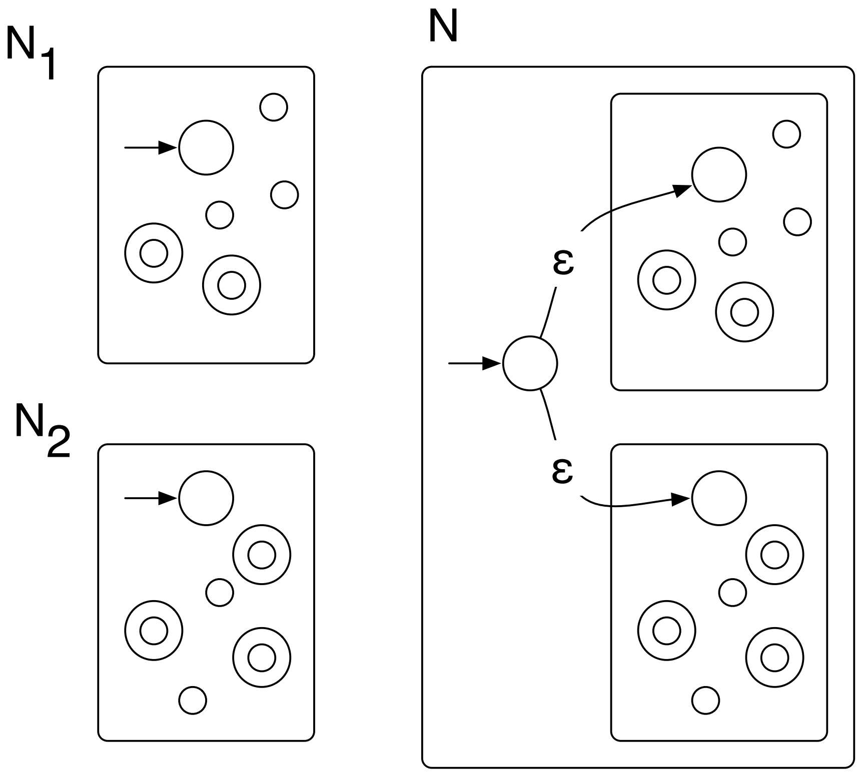
- **Übergangsfunktion** δ'
- Für alle $R \in Q'$ und $a \in \Sigma$ gelte
  $\delta'(R, a) = \{q \in Q \mid \exists r \in R : q \in E(\delta(r, a))\}$
  andere Notation:
  - $\delta'(R, a) = \bigcup_{r \in R} E(\{\delta(r, a)\})$
- **Anfangszustand**: $q_0 = E(\{q_0\})$
- **Akzeptierende Zustände F'**:
  - Zustand R akzeptiert, falls ein akzeptierender Zustand von F in R ist
  - $F' = \{R \in Q' \mid \exists r \in R : r \in R\}$





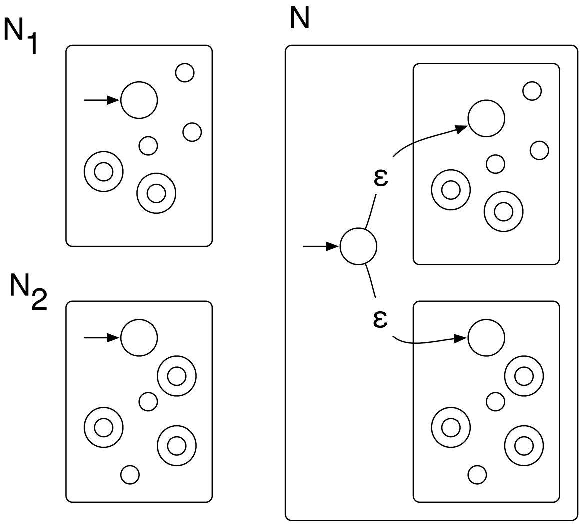
Abschluss über der Vereinigung
Ein alternativer Beweis

- **Beweisskizze**:
  - Betrachte NFAs N_1 und N_2
  - Konstruiere N mit neuem Startzustand und ε-Übergangen zu den Startzuständen von N_1 und N_2
  - NFA N akzeptiert $L(N_1) \cup L(N_2)$



Abschluss unter Vereinigung
Alternativer Beweis

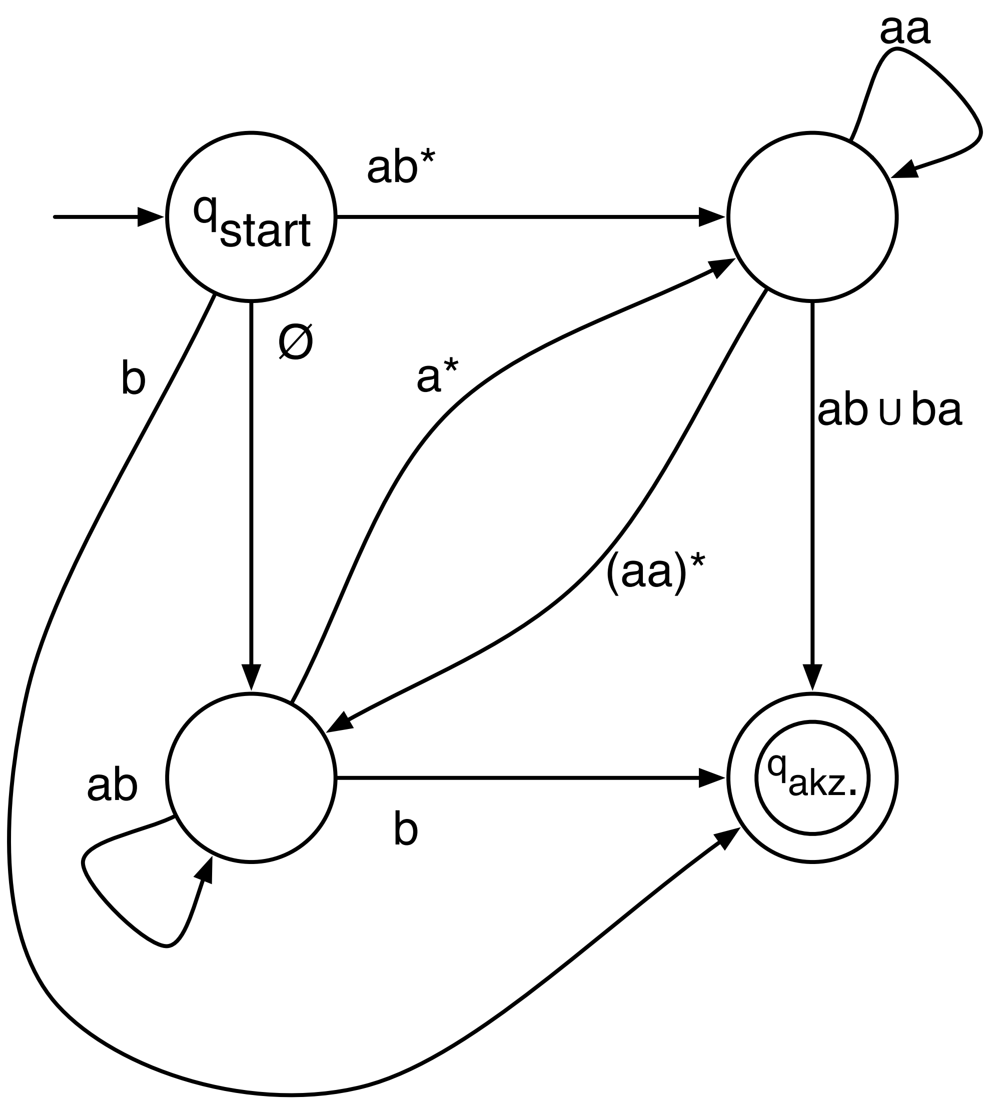
- Gegeben seien nichtdeterministische endliche Automaten
  - $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ und
  - $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$



- Konstruktion von $N = (Q, \Sigma, \delta, q_0, F)$ so dass $L(N_1) \cup L(N_2)$
  - **Zustandsmenge**: $Q = \{q_0\} \cup Q_1 \cup Q_2$
  - **Anfangszustand**: q_0
  - **Akzeptierende Zustände**: $F = F_1 \cup F_2$
  - **Übergangsfunktion**
  - $$\delta(q, a) = \begin{cases} \delta_1(q, a), & q \in Q_1 \\ \delta_2(q, a), & q \in Q_2 \\ \{q_1, q_2\}, & q = q_0 \text{ und } a = \epsilon \\ \emptyset, & q = q_0 \text{ und } a \neq \epsilon \end{cases}$$

universität freiburg

REG und reguläre Ausdrücke

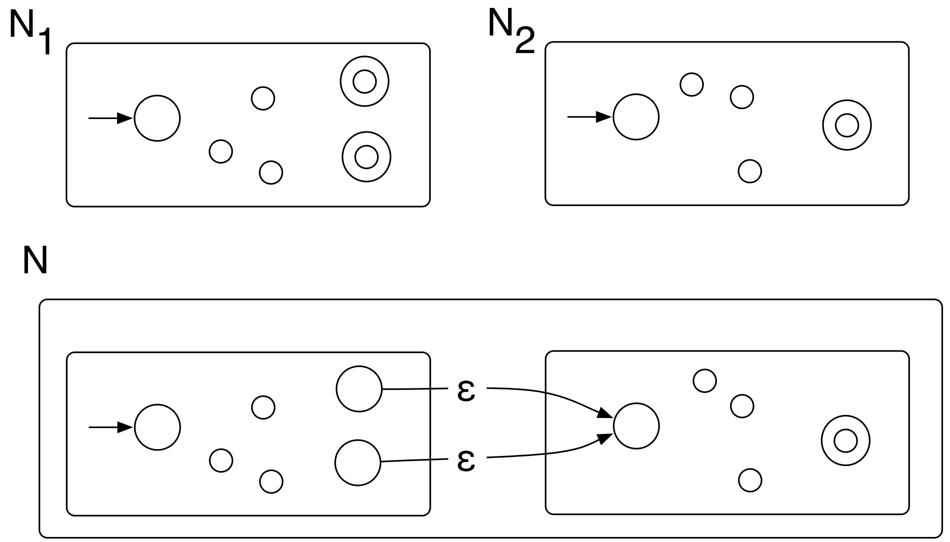


Formale Sprachen und
Endliche Automaten

Reguläre Sprachen
Abschluss gegenüber der Konkatenation

Beweisskizze

- Betrachte NFA N_1 und N_2
- **Konstruiere NFA N mit**
  - **ε-Übergangen** von **allen akzeptierenden Zuständen** von NFA N_1
  - **zu dem Startzustand** von NFA N_2
- **Neuer Startzustand von N** ist Startzustand von N_1
  - Die neuen akzeptierenden Zustände sind die von N_2
  - NFA N akzeptiert $L(N_1) \circ L(N_2)$



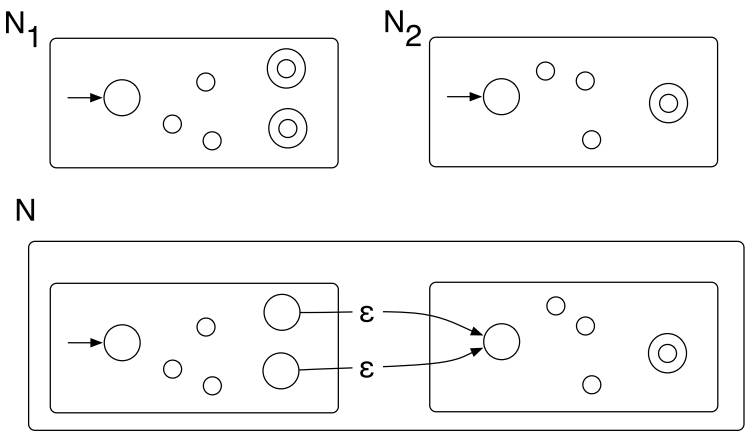
Reguläre Sprachen
Abschluss gegenüber der Konkatenation

Theorem

- Die Menge der regulären Sprachen ist abgeschlossen unter Konkatenation.

Beweis:

- Gegeben seien nichtdeterministische endliche Automaten
  - $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ und
  - $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

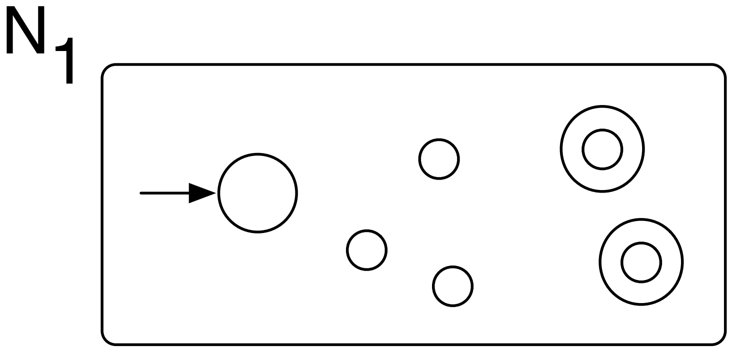


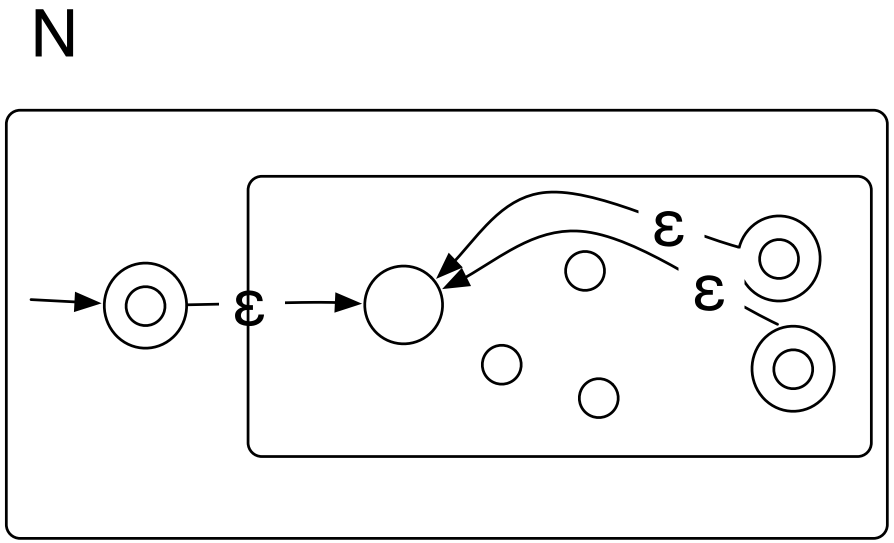
- **Konstruktion von $N = (Q, \Sigma, \delta, q_0, F)$,**
  - sodass $L(N) = L(N_1) \circ L(N_2)$
  - **Zustandsmenge**: $Q' = Q_1 \cup Q_2$
  - **Anfangszustand**: $q_0 = q_1$
  - **Akzeptierende Zustände**: $F = F_2$
- **Übergangsfunktion**
  - $$\delta(q,a) = \begin{cases} \delta_1(q,a), & q \in Q_1 \text{ und } q \notin F_1 \\ \delta_1(q,a), & q \in F_1 \text{ und } a \neq \epsilon \\ \delta_1(q,a) \cup \{q_2\}, & q \in F_1 \text{ und } a = \epsilon \\ \delta_2(q,a), & q \in Q_2 \end{cases}$$
- Korrektheit der Konstruktion zur Übung empfohlen

Reguläre Sprachen
Abschluss unter dem Stern-Operator

Beweisskizze:

- Betrachte NFA N_1
- Konstruiere NFA N mit neuem Startzustand
- ε-Übergang vom neuem Startzustand zum Alten
- ε-Übergangen von allen akzeptierenden Zuständen zum Ex-Startzustand
- Der Rest von V bleibt gleich
- $L(N) = L(N_1)^*$





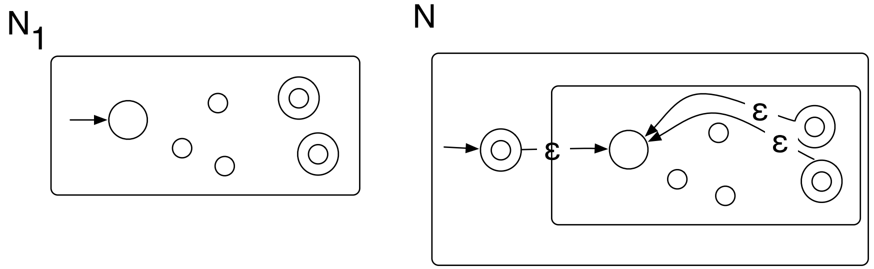
Reguläre Sprachen
Abschluss unter dem Stern-Operator

Theorem

- Die Menge der regulären Sprachen ist abgeschlossen unter der Stern-Operation

Beweis:

- Gegeben sei der nichtdeterministische endliche Automat
  - $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ und



- **Konstruktion von N mit** $L(N) = L(N_1)*$
  - **Zustandsmenge**: $Q' = \{q_0\} \cup Q_1$
  - **Anfangszustand**: q_0
  - **Akzeptierende Zustände:** $F = \{q_0\} \cup F_1$
  - **Übergangsfunktion**
  - $$\delta(q,a) = \begin{cases} \delta_1(q,a), & q \in Q_1 \text{ und } q \notin F_1 \\ \delta_1(q,a), & q \in F_1 \text{ und } a \neq \epsilon \\ \delta_1(q,a) \cup \{q_1\}, & q \in F_1 \text{ und } a = \epsilon \\ \{q_1\}, & q = q_0 \text{ und } a = \epsilon \\ \emptyset, & q = q_0 \text{ und } a \neq \epsilon \end{cases}$$
  - Rest des Beweises zum Üben

Theoretische Informatik
Vielen Dank für die Aufmerksamkeit

Christian Schindelhauer

Rechnernetze und Telematik

Institut für Informatik

Technische Fakultät

schindel(at)tf.uni-freiburg.de

Sneha Mohanty

Rechnernetze und Telematik

Institut für Informatik

Technische Fakultät

mohanty(at)informatik.uni-freiburg.de

Notfolie
für kurze Arme

Gegeben

- zwei Automaten und die Eingabe **0110**
- Simuliere mit **zwei Fingern** die Automaten gleichzeitig
- Falls **einer der „Finger“ akzeptiert, akzeptiere**

Idee

- Ersetze zwei Finger durch mehr Zustände