Informatik III

CoNe Freiburg

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Christian Schindelhauer
Wintersemester 2006/07
2. Vorlesung
27.10.2006
schindel@informatik.uni-freiburg.de

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Reguläre Sprachen

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Reguläre Sprachen

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

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

Reguläre Operationen

- **Definition**
  - Die regulären Operationen **Vereinigung**, **Konkatenation** und **Stern** werden wie folgt definiert

1. **Vereinigung**:

$$A \cup B = \{x \mid x \in A \ \text{ oder } \ x \in B\}$$

2. **Konkatenation**

$$A \circ B = \{xy \mid x \in A \ \text{ und } \ y \in B\}$$

3. **Stern**

$$A^* = \{x_1 x_2 \ldots x_k \mid k \geq 0 \ \text{ und } \ x_i \in A\}$$

Beispiel

- **Sei Σ = {a,b,c,d,..., z}**
  - A = {wadda, hadda}
  - B = {du, da}

- **Vereinigung**

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

- **Konkatenation**

$$A \circ B = \{waddadu, waddada, haddadu, haddada\}$$

- **Stern**

$$B^* = \{\epsilon, du, da, dudu, duda, dadu, dada, dududu, dududa, dudadu, \ldots\}$$

Abschluss unter Vereinigung

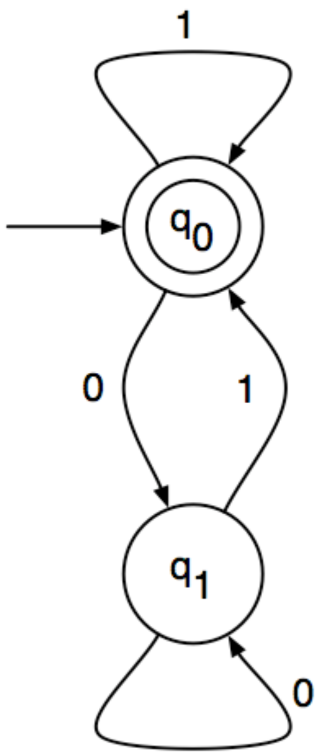
- **Theorem**
  - Die Klasse der regulären Sprachen ist abgeschlossen unter der Vereinigung.

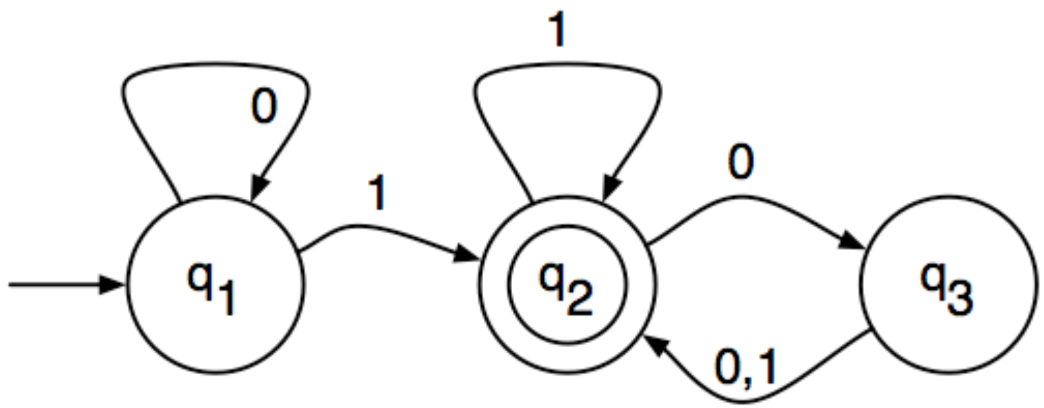
- **D.h. Falls A regulär ist und B regulär ist, dann ist auch A $\cup$ B regulär**
- **D.h.**
  - Gegeben ein endlicher Automat für A und
  - gegeben ein endlicher Automat für B,
    - dann gibt es auch einen für A $\cup$ B

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Beweisidee

- Gegeben 2 Automaten und die Eingabe 0 1 1 0
- Simuliere mit zwei Fingern die Automaten gleichzeitig
- Falls einer der Finger akzeptiert, akzeptiere

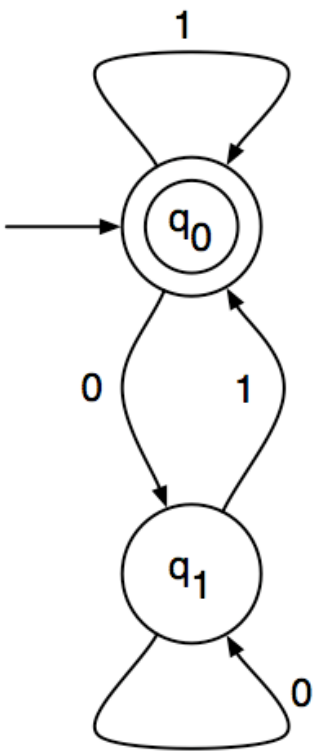


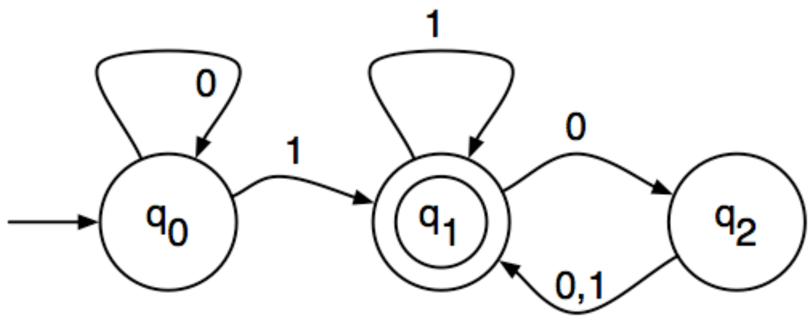


Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Beweisidee

- Gegeben 2 Automaten und die Eingabe 0 1 1 0
- Simuliere mit zwei Fingern die Automaten gleichzeitig
- Falls einer der Finger akzeptiert, akzeptiere

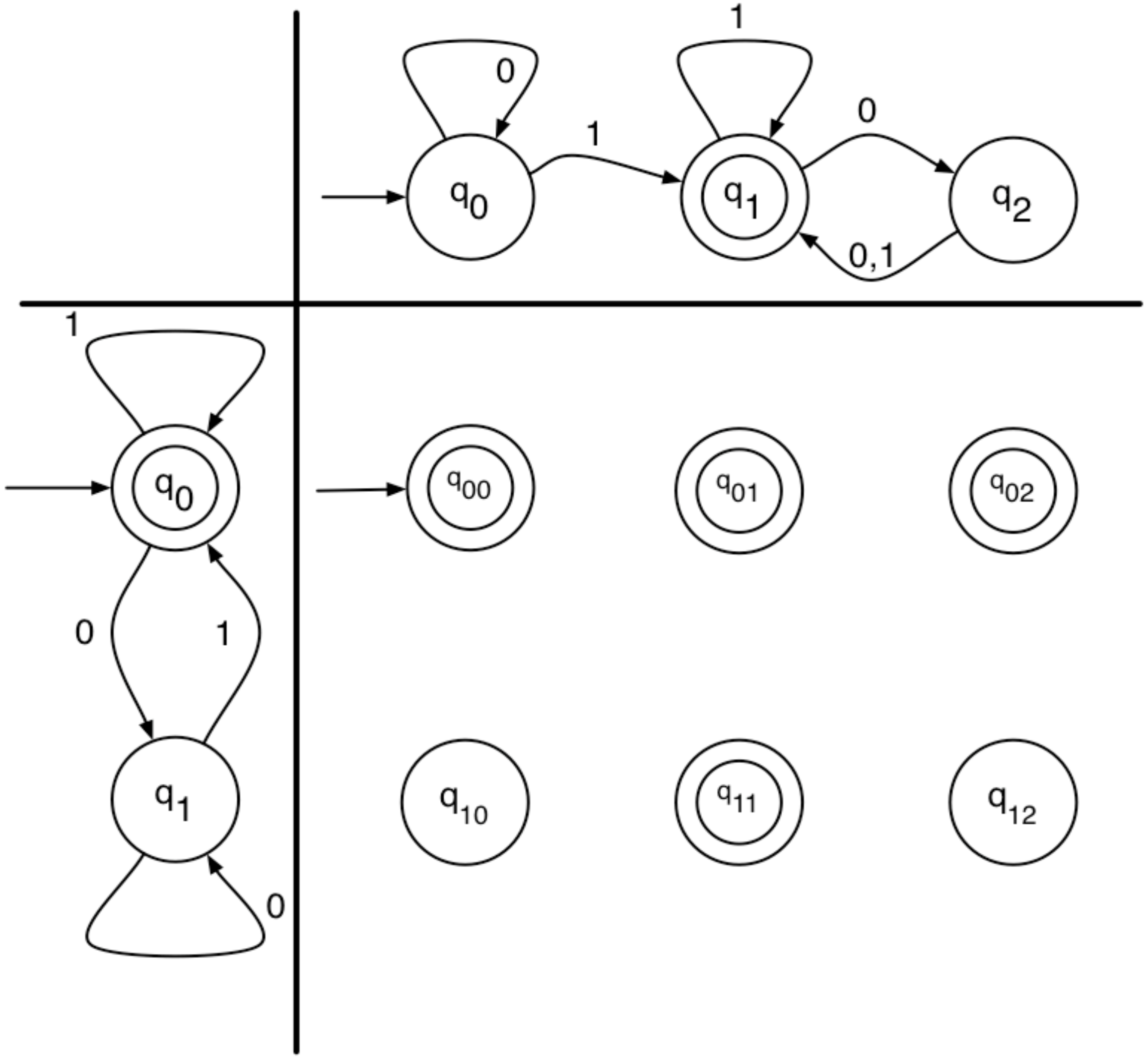




Idee: Ersetze zwei Finger durch mehr Zustände

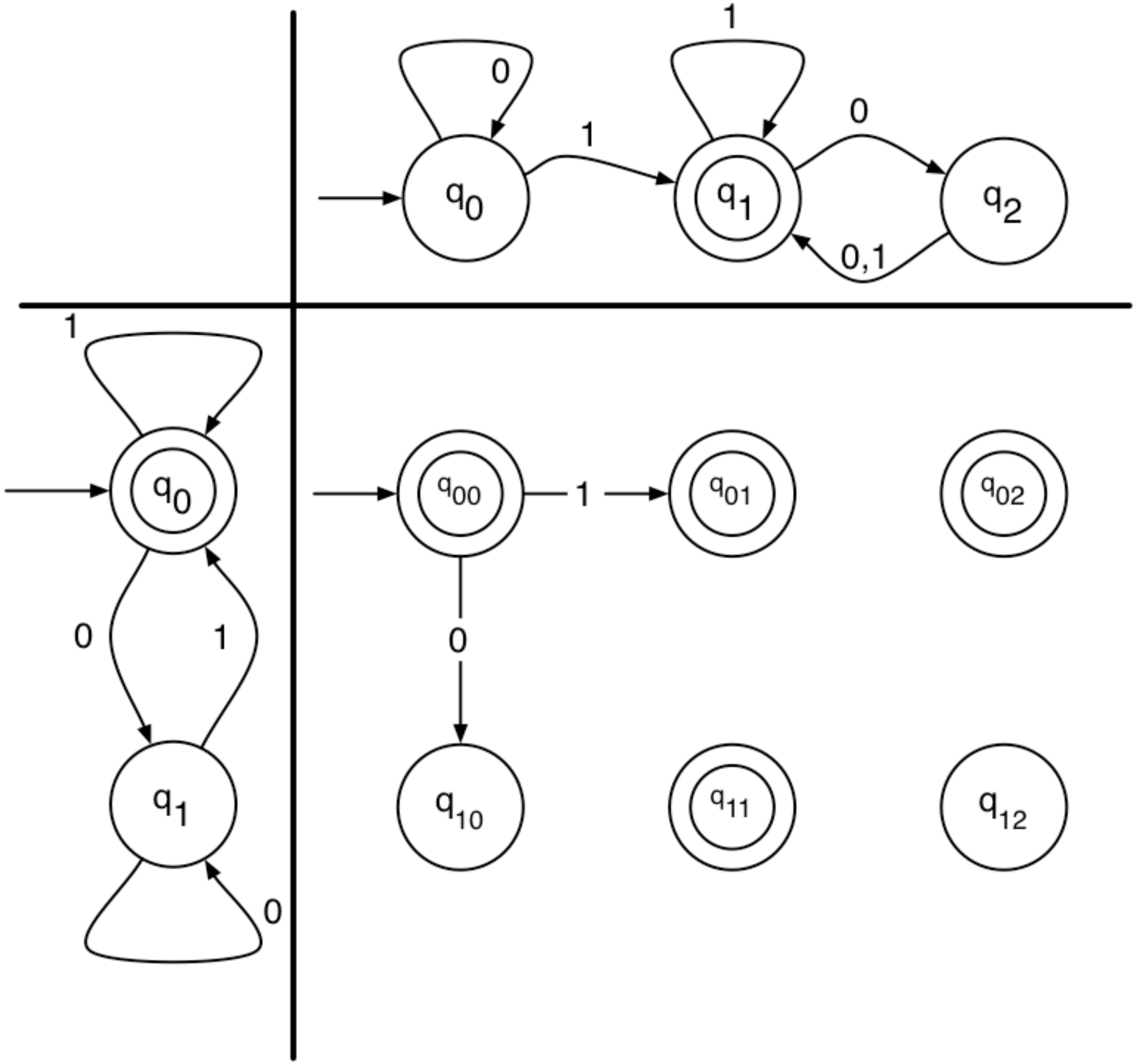
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Die Konstruktion des kartesischen Produkts

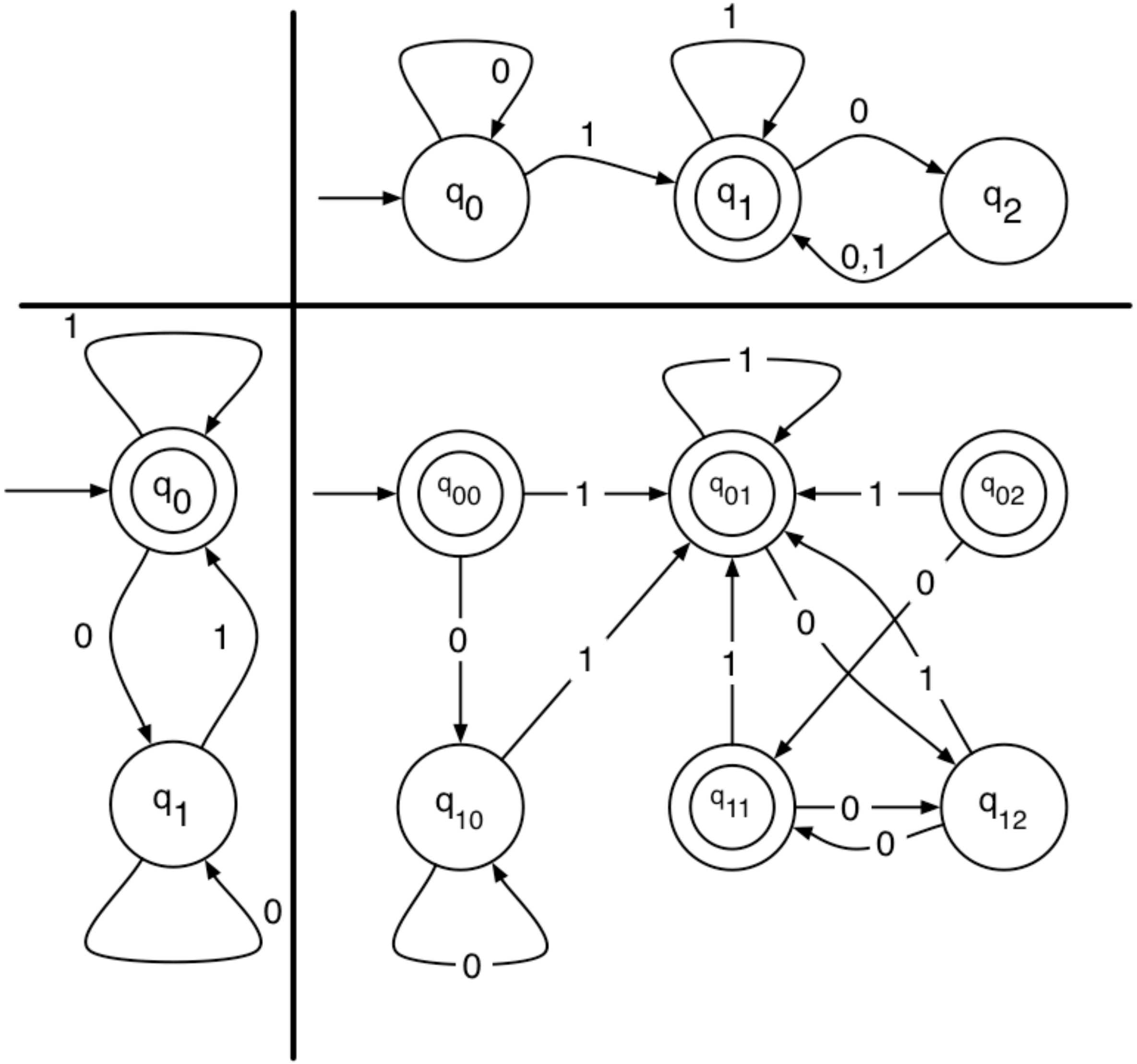


Ein Zustandsübergang

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer



Der Automat für die vereinigte Sprache



Die formale Konstruktion

- **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

1. **Zustandsmenge**
   $Q = \{ (r_1,r_2) \mid r_1 \in Q_1$ und $r_2 \in Q_2\}$,
   Q ist das **kartesische Produkt** aus Q_1 und Q_2
2. **Alphabet bleibt gleich**
   Wir erlauben uns diese Vereinfachung, dass die Ausgangsalphabete gleich sind.
   Eine kleine Modifikation würde den anderen Fall auch beweisen
3. **Übergangsfunktion**
   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:**
   $$F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ oder } r_2 \in F_2\}$$

Der formale Beweis

- **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 konstruierten einen Automaten $M = (Q, \Sigma, \delta, q_0, F)$ der $A_1 \cup A_2$ erkennt
  - Der Rest des Beweises erfolgt durch eine Induktion über die Länge der Worte
    - Induktionsanfang: Maschine ist in q_0 für ε
    - Induktionsannahme: Maschine ist in Zustand $(\delta_1(q_1,w), \delta_2(q_2,w))$ nach Abarbeiten von w
      - Erweiterung der Def. der Übergangsfunktion: $\delta_1(q,wa) = \delta_1(\delta_1(q,w),a)$
    - Induktionsbehauptung: 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: $F = \{(r_1, r_2) \mid r_1 \in F_1 \ \text{ oder } \ r_2 \in F_2\}$

Die Konkatenation

- **Sind endliche Automaten auch unter Konkatenation abgeschlossen?**

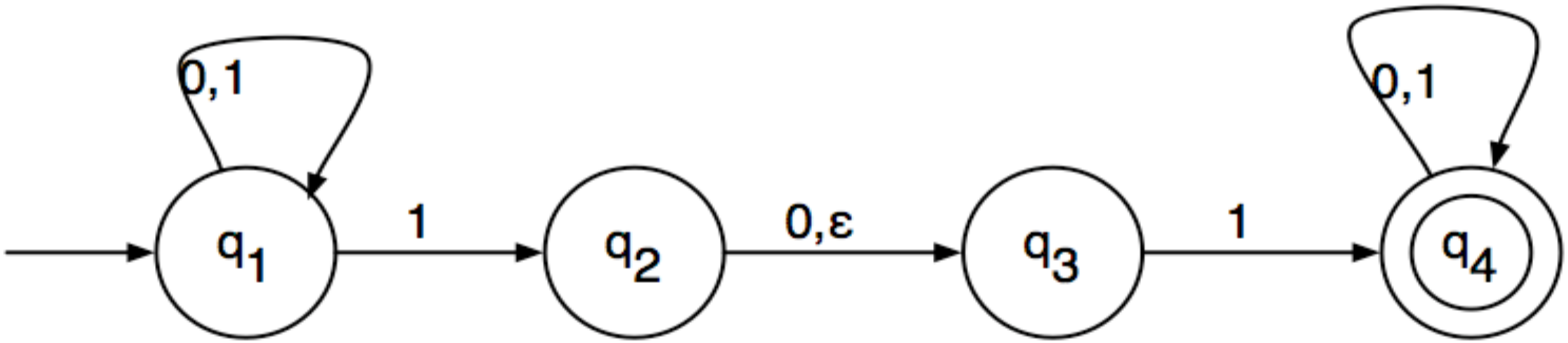
- **1. Idee:**
  - Automat M_1 akzeptiert Sprache A_1
  - Automat M_2 akzeptiert Sprache A_2
  - Konstruiere Automat M, der aus den Zuständen von M_1 und M_2 besteht
  - Die Übergänge müssen dann noch hingebastelt werden...
  - ... ? ....

- **Nach längeren Probieren:**
  - Idee funktioniert nicht mit DFAs!!!!

- **Lösung: Betrachte nicht-deterministische endliche Automaten (NFA)**
  (Non-deterministic Finite Automaton)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Ein nichtdeterministischer Automat



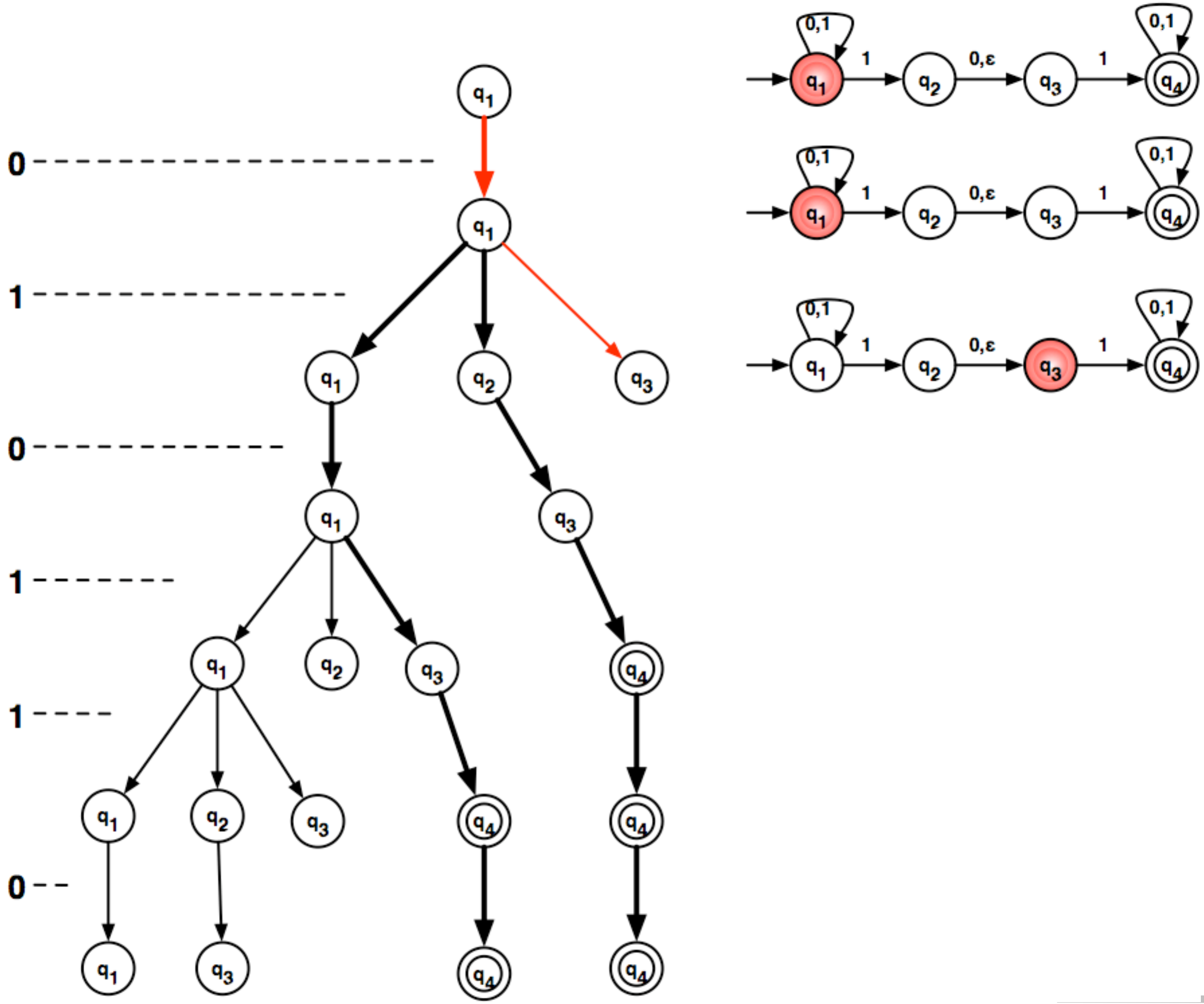
- **Was ist anders:**
  - –Von q_1 gehen zwei Übergänge mit 1 aus
  - –Von q_2 geht ein ε-Übergang nach q_3
  - –Von q_2 fehlt der Übergang bei Zeichen 1
  - –Von q_3 fehlt der Übergang bei Zeichen 0
- **Wann akzeptiert so ein Automat?**

ACHTUNG, ACHTUNG !!!

- **Verwechslungsgefahr:**
  - Die Sprache, die aus keinem Wort besteht, ist die leere Menge: $\emptyset$
    - $\emptyset$ ist kein Wort!
  - Das leere Wort ε ist keine Sprache, sondern eine Folge der Länge 0.
    - ε ist keine Sprache.
    - ε ist kein Zeichen!!!
    - Es gibt Sprachen, die ε beinhalten.
    - Es gibt Sprachen, die ε nicht beinhalten.

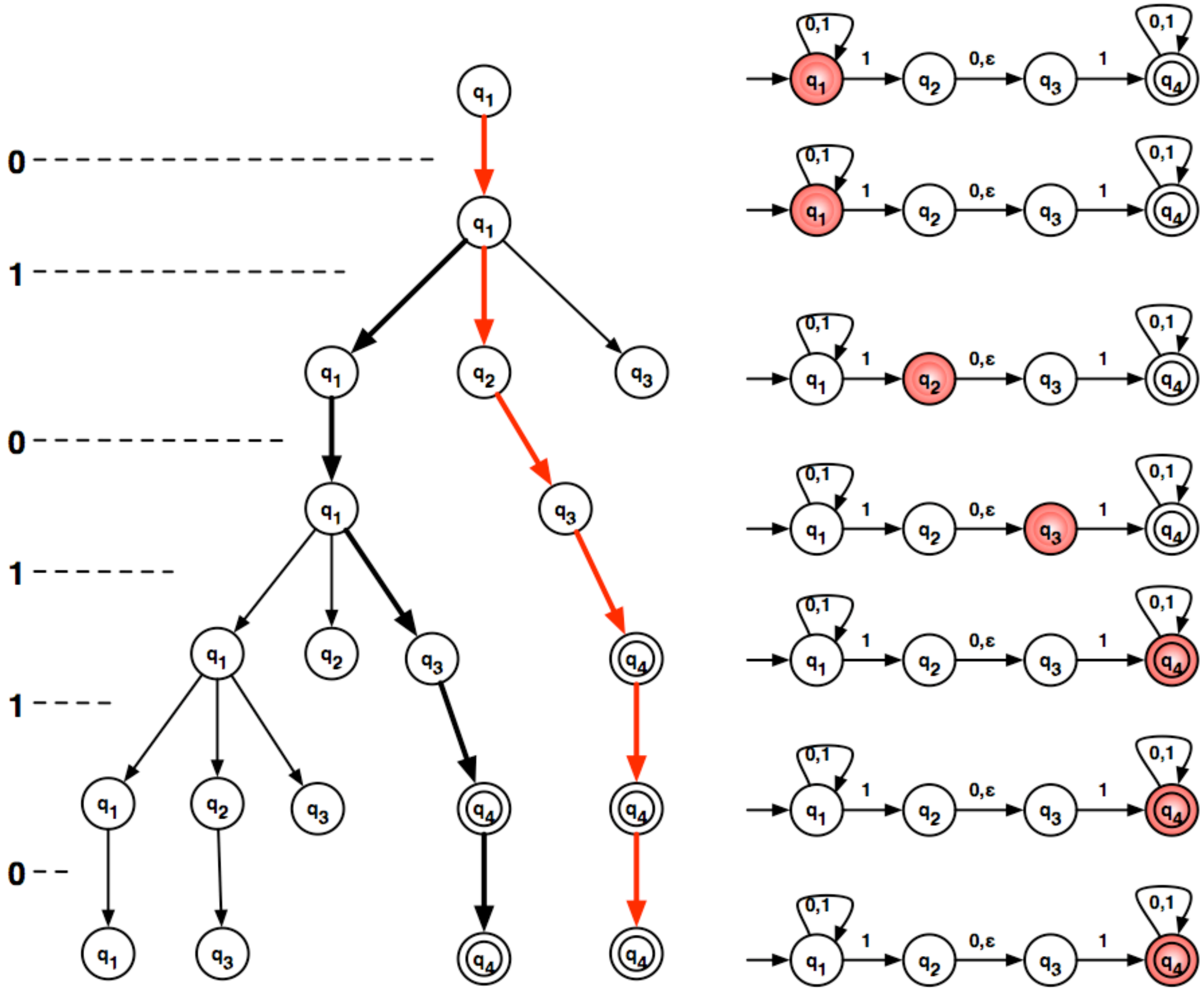
Nichtakzeptierende Berechnung eines NFA

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer



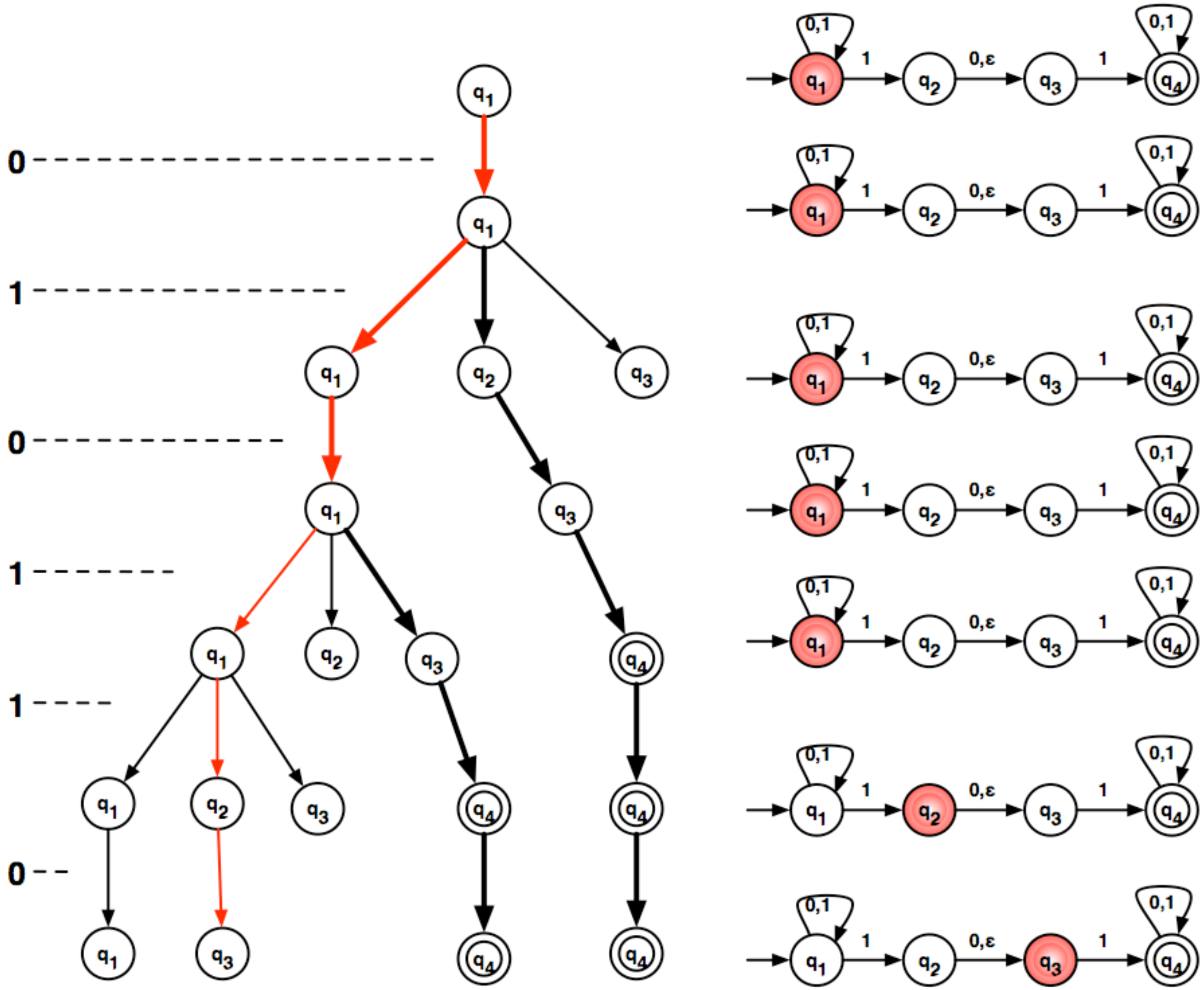
Akzeptierende Berechnung eines NFA

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer



Nicht akzeptierende Berechnung

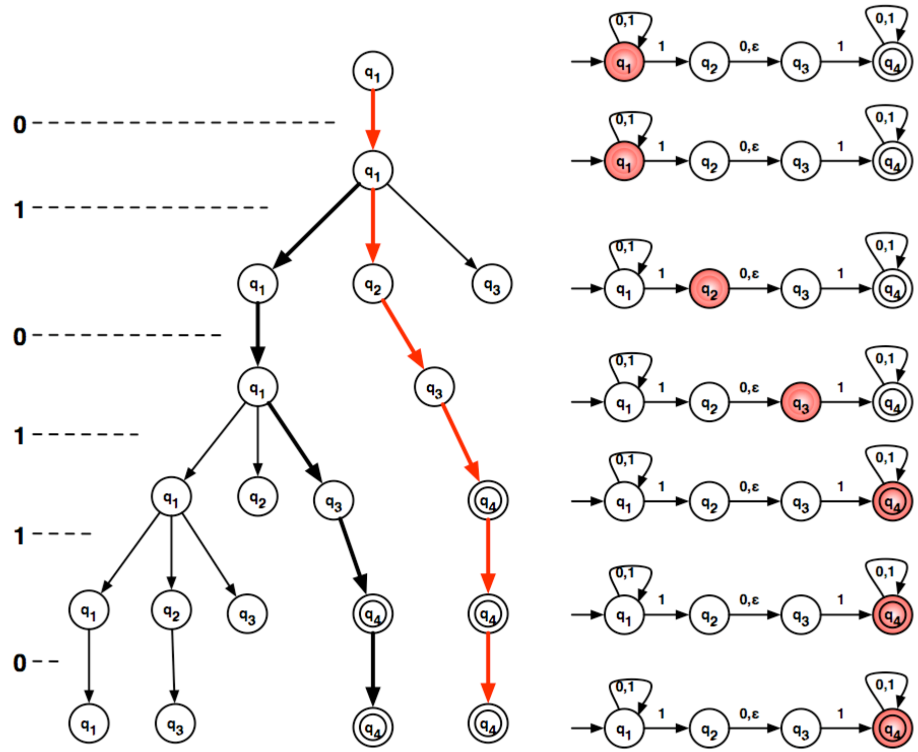
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer



Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Wann akzeptiert ein NFA? (nondeterministic finite automaton)

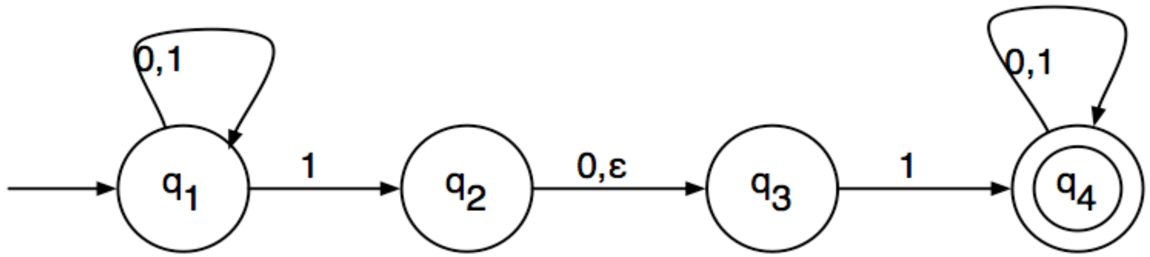
Ein nicht-deterministischer endlicher Automat akzeptiert ein Wort, falls es (unter allen möglichen) mindestens eine akzeptierende Berechnung gibt.



Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Komponenten des Beispielautomaten

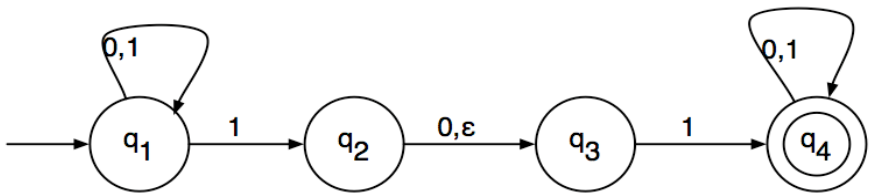
1. **Zustände $Q = \{q_1, q_2, q_3, q_4\}$**
2. **Alphabet $\Sigma = \{0,1\}$**
3. **Übergangsfunktion:**
   - $\delta(q_1,0) = \{q_1\}$
   - $\delta(q_1,1) = \{q_1, q_2\}$
   - $\delta(q_1,\varepsilon) = \emptyset$
   - $\delta(q_2,0) = \{q_3\}$
   - $\delta(q_2,1) = \emptyset$
   - $\delta(q_2,\varepsilon)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



	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

- **Notation:**
  - Sei $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$
  - P(M) die Potenzmenge von M
    - z.B. $P(\{a,b\})=\{\emptyset,\{a\},\{b\},\{a,b\}\}$
- **Definition**
  - Ein **nichtdeterministischer endlicher Automat** wird durch das 5-Tupel $(Q, \Sigma, \delta, q_0, F)$ beschrieben
    1. Q: Eine endliche Menge von **Zuständen**
    2. Σ: ist das **Alphabet**
    3. $\delta: Q \times \Sigma_\varepsilon \rightarrow P(Q)$ ist die **Übergangsfunktion**
    4. $q_0 \in Q$: ist der **Startzustand**
    5. $F \subseteq Q$: ist die Menge der **akzeptierenden Zustände**



Berechnung eines nichtdeterministischen endlichen Automatens (NFA)

- **Definition**
  - Sei M= (Q, Σ, δ, q_0, F) ein nichtdeterministischer endlicher Automat. Falls es eine
    - Darstellung der Eingabe $w = q_1 q_2 \ldots q_m$ über dem Alphabet Σ_ε
    - und eine Folge $r_0 r_1 \ldots r_m$ von Zuständen aus Q gibt, wobei
      1. $r_0 = q_0$
      2. $r_{i+1} \in \delta(r_i, w_{i+1})$, für alle i =0, ..., m-1
      3. $r_m \in F$
  - dann **akzeptiert** M das Wort.

L(DFA) $\subseteq$ L(NFA)

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Theorem**
  - Jeder deterministische endliche Automat hat einen äquivalenten nichtdeterminstischen Automaten.

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

L(NFA) $\subseteq$ L(DFA)

- **Theorem**
  - Jeder nichtdeterministischer endliche Automat hat einen äquivalenten determinstischen Automat.
- **Beweis:**
  - Gegeben sei ein nichtdeterministischer endlicher Automat N = (Q, Σ, δ, q_0, F)
  - dann konstruieren wir den DFA M = (Q', Σ, δ', q_0', F') wie folgt:
  - 1. Fall: kein ε-Übergang in δ

1. **Zustandsmenge: Q' = P(Q),**
   Q' ist die **Potenzmenge von Q**
2. **Alphabet bleibt gleich**
3. **Übergangsfunktion**
   Für alle r ∈ Q_1 und a ∈ Σ gelte

$$\delta'(R,a) = \{q \in Q \mid q \in \delta(r,a) \text{ für ein } r \in R\}$$

andere Notation:

$$\delta'(R,a) = \bigcup_{r \in R} \{\delta(r,a)\}$$

4. **Anfangszustand: q_0 = {q_0}**
5. **Akzeptierende Zustände:**

$$F' = \{R \in Q' \mid \exists r \in F \ : \ r \in R\}$$

Zustand R akzeptiert, falls ein akzeptierender Zustand von F in R ist

Ende der 2. Vorlesung

Albert-Ludwigs-Universität Freiburg
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

Informatik III

Christian Schindelhauer
schindel@informatik.uni-freiburg.de