Informatik III

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

Christian Schindelhauer

Wintersemester 2006/07

3. Vorlesung

02.11.2006

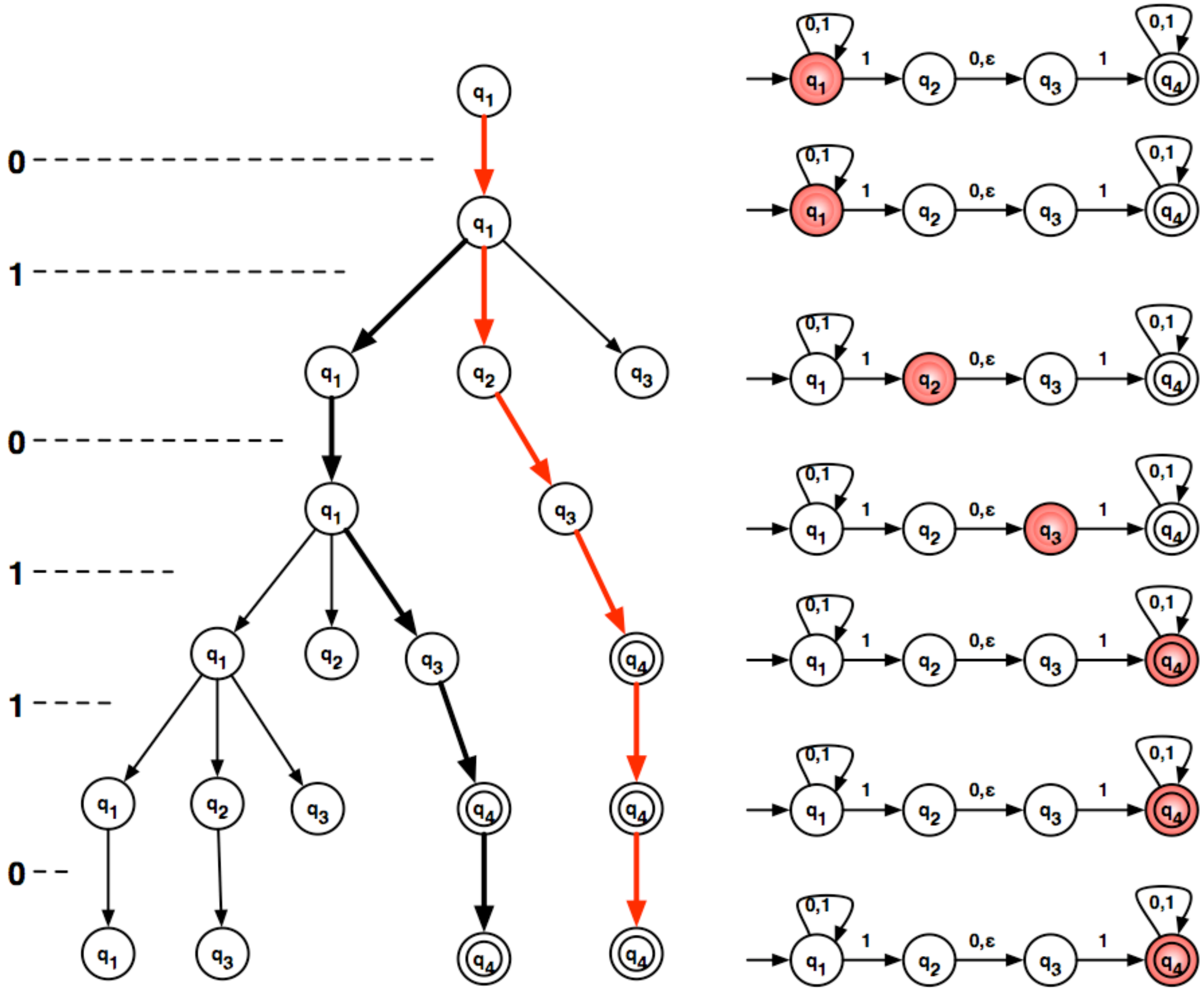
schindel@informatik.uni-freiburg.de

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

Reguläre Sprachen und Ausdrücke

Akzeptierende Berechnung eines NFA

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



L(NFA) ⊆ L(DFA)

- **Theorem**
  - Jeder nichtdeterministischer endliche Automat hat einen äquivalenten determinstischen 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

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

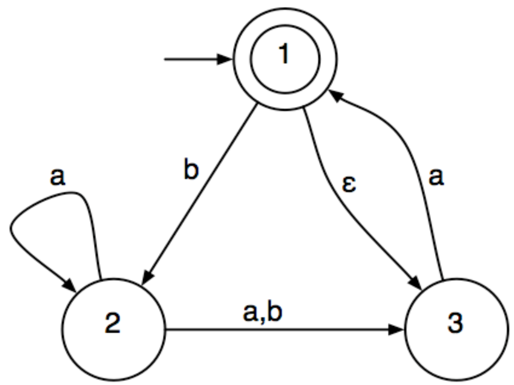
$$F' = \{R \in Q' \mid \exists r \in F \ : \ r \in R\}$$

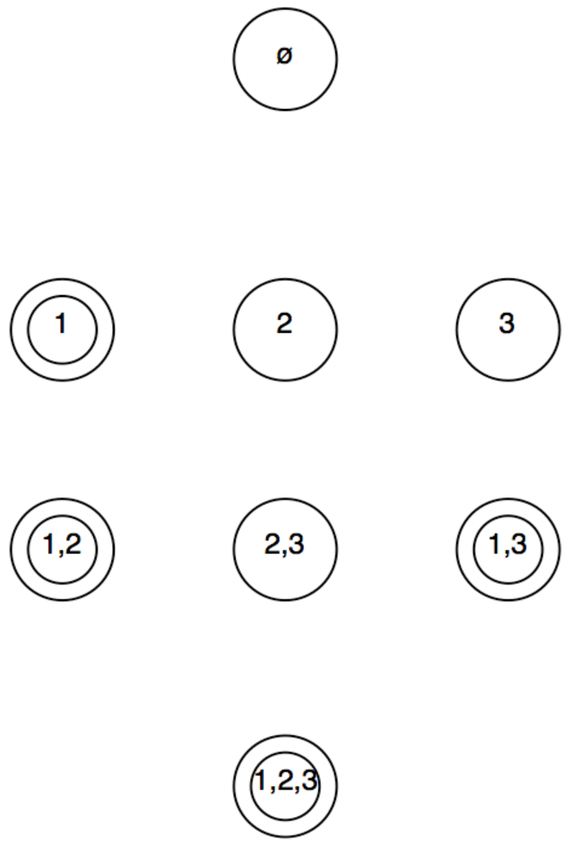
Zustand R akzeptiert, falls ein akzeptierender Zustand von F in R ist

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

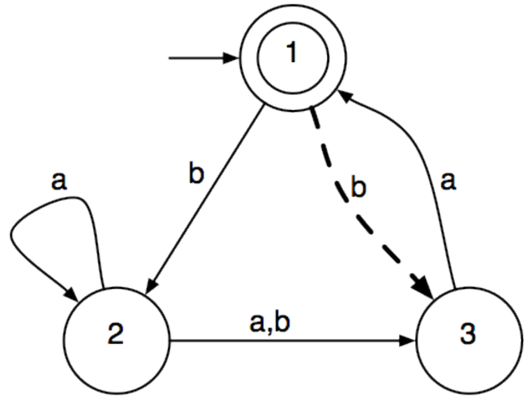
Potenzmenge

1. **Zustandsmenge: Q' = P(Q),**
   Q' ist die **Potenzmenge von Q**





Ohne ε-Übergang



1. ..
2. **Übergangsfunktion**
   Für alle r ∈ Q_1
   und a ∈ Σ gelte

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

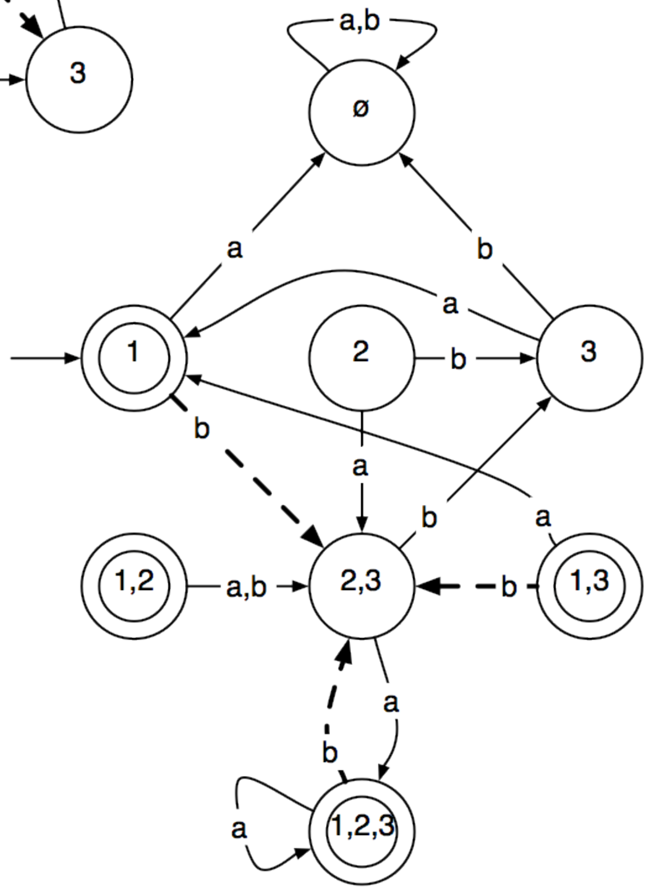
   andere Notation:

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

3. **Anfangszustand: q_0 = {q_0}**
4. **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



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

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

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:

2. Fall: Mit ε-Übergang in δ

Notation:

$$E(R) = \{q \mid q \text{ wird von R durch keinen, einen oder mehr } \epsilon\text{-Übergänge erreicht}\}$$

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

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

andere Notation:

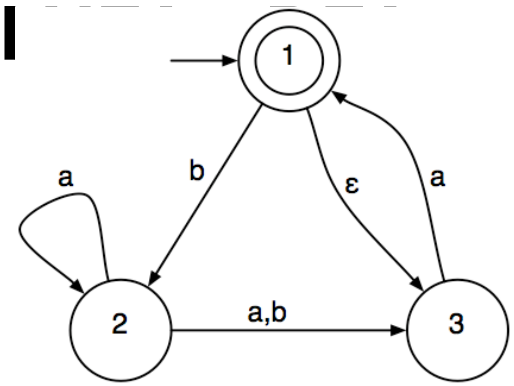
$$\delta'(R,a) = \bigcup_{r \in R} E(\{\delta(r,a)\})$$

4. **Anfangszustand: $q_0 = E(\{q_0\})$**
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

Beispiel



1. ..
2. ..

3. **Übergangsfunktion**
   Für alle $r \in Q_1$ und $a \in \Sigma$ gelte

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

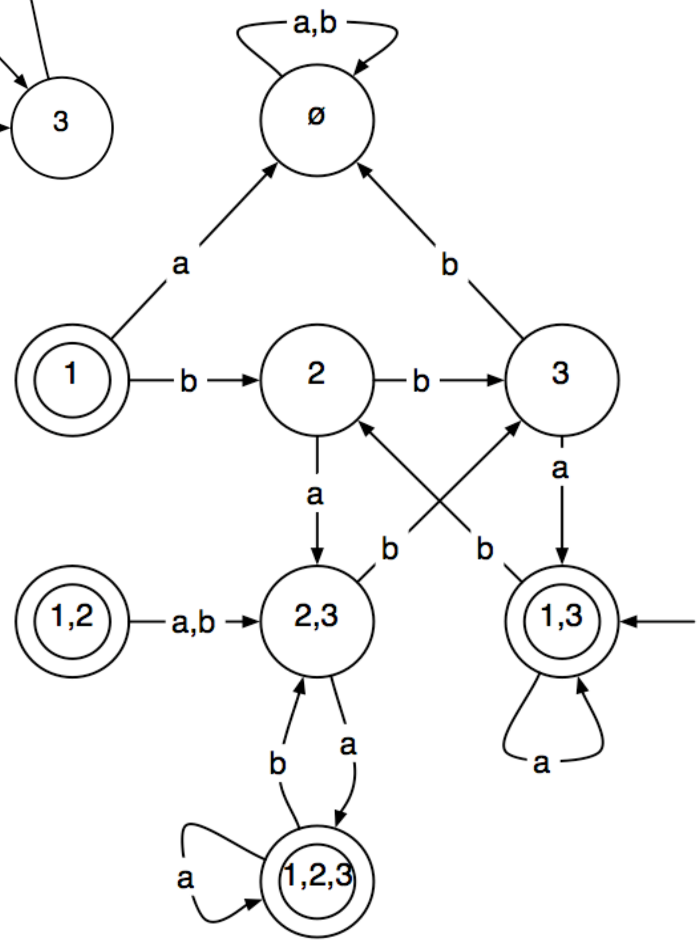
andere Notation:

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

4. **Anfangszustand: $q_0 = E(\{q_0\})$**
5. **Akzeptierende Zustände:**

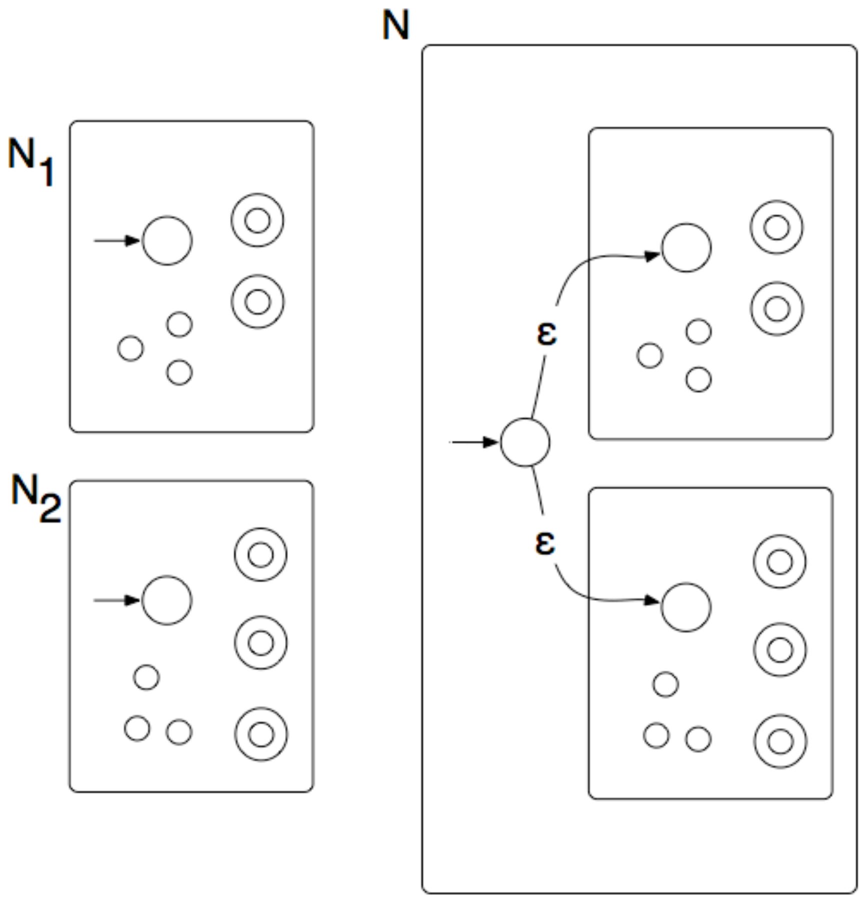
$$F' = \{R \in Q' \mid \exists r \in F \ : \ r \in R\}$$

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



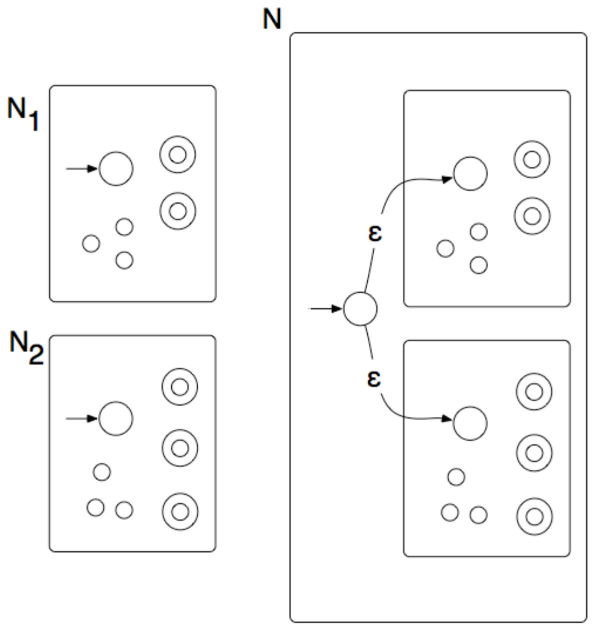
Ein alternativer Beweis für den Abschluß über der Vereinigung

- **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 $N = (Q, \Sigma, \delta, q_0, F)$ mit $L(N)= L(N_1) \cup L(N_2)$

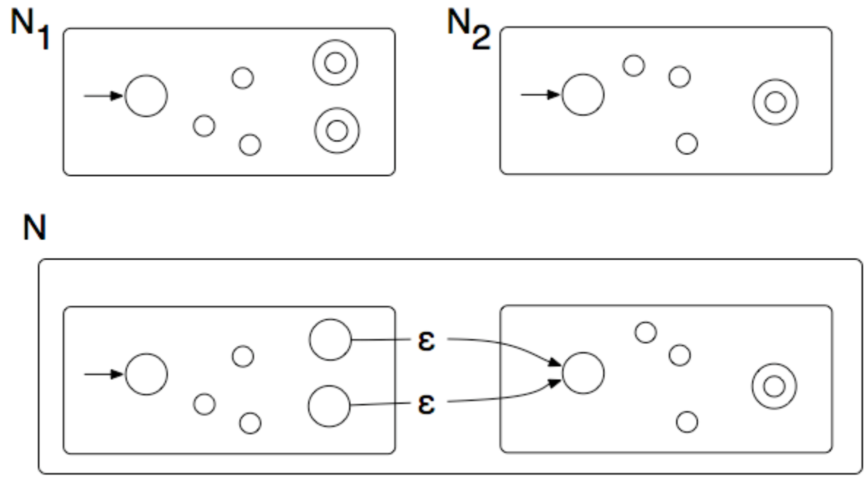
1. Zustandsmenge: $Q = \{q_0\} \cup Q_1 \cup Q_2$
2. Anfangszustand: q_0
3. Akzeptierende Zustände: $F = F_1 \cup F_2$
4. Übergangsfunktion

$$\delta(q,a) = \begin{cases} \delta_1(q,a)\ , & q \in Q_1 \\ \delta_2(q,a)\ , & q \in Q_2 \\ \{q_1, q_2\}\ , & q = q_0 \text{ und } a = \epsilon \\ \emptyset\ , & q = q_0 \text{ und } a \neq \epsilon \end{cases}$$

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

Die regulären Sprachen sind gegenüber der Konkatenation abgeschlossen

- **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 $L(N_1)L(N_2)$

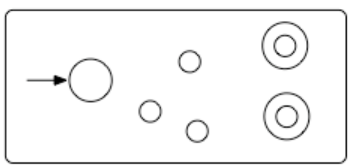


Abschluss unter Konkatenation

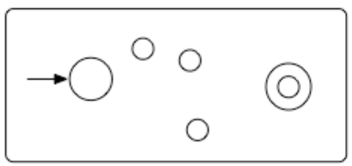
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **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
    - $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

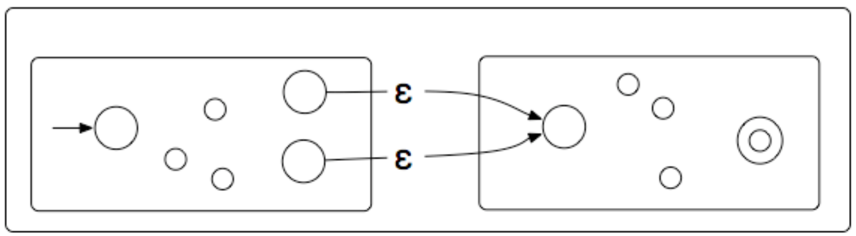
N_1



N_2



N



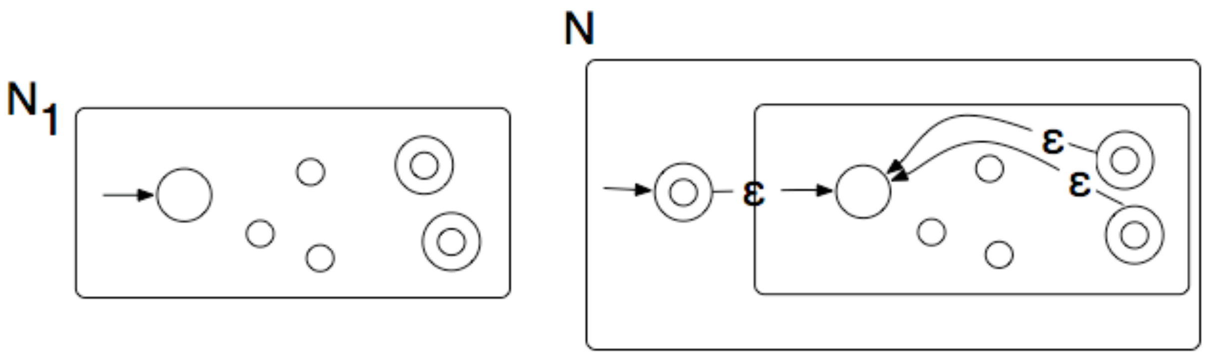
Konstruktion von N mit $L(N)= L(N_1)\, L(N_2)$

1. Zustandsmenge: $Q' = Q_1 \cup Q_2$
2. Anfangszustand: $q_0 = q_1$
3. Akzeptierende Zustände: $F = F_2$
4. Übergangsfunktion

$$\delta(q,a) = \begin{cases} \delta_1(q,a)\ , & q \in Q_1 \text{ und } q \notin F_1 \\ \delta_1(q,a)\ , & q \in F_1 \text{ und } a \neq \epsilon \\ \delta_1(q,a) \cup \{q_2\}\ , & q \in F_1 \text{ und } a = \epsilon \\ \delta_2(q,a)\ , & q \in Q_2 \end{cases}$$

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

- **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 Exstartzustand
  - Rest bleibt gleich in N
  - $L(N) = L(N_1)^*$



Abschluss unter der Stern-Operation

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

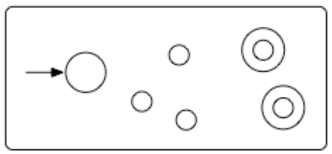
- **Theorem**
  - Die Menge der regulären Sprachen ist abgeschlossen unter der Stern-Operation
- **Beweis:**
  - Gegeben sei der nichtdeterministische endliche Automat
    - $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ und

Konstruktion von N mit
$L(N) = L(N_1)^*$

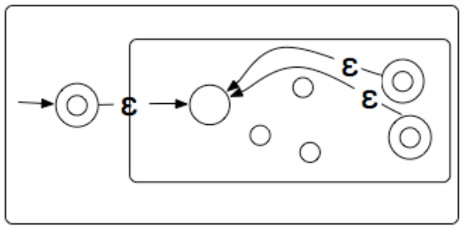
1. Zustandsmenge:
   $Q' = \{q_0\} \cup Q_1$
2. Anfangszustand: q_0
3. Akzeptierende Zustände:
   $F = \{q_0\} \cup F_1$
4. Übergangsfunktion

$$\delta(q,a) = \begin{cases} \delta_1(q,a)\ , & q \in Q_1 \text{ und } q \notin F_1 \\ \delta_1(q,a)\ , & q \in F_1 \text{ und } a \neq \epsilon \\ \delta_1(q,a) \cup \{q_1\}\ , & q \in F_1 \text{ und } a = \epsilon \\ \{q_1\}\ , & q = q_0 \text{ und } a = \epsilon \\ \emptyset\ , & q = q_0 \text{ und } a \neq \epsilon \end{cases}$$

N_1



N



Reguläre Ausdrücke

- **Definition**
  - R ist ein regulärer Ausdruck, wenn R eines der Darstellungen besitzt
    - a, für ein Zeichen $a \in \Sigma$
    - ε
    - $\emptyset$
    - $(R_1 \cup R_2)$, wenn R_1 und R_2 schon reguläre Ausdrücke sind
    - $(R_1 \circ R_2)$, wenn R_1 und R_2 reguläre Ausdrücke sind
    - $(R_1)^*$, wenn R_1 ein regulärer Ausdruck ist

- **Notation**
  - statt $\mathbf{R_1 \circ R_2}$ schreiben wir $\mathbf{R_1\, R_2}$
  - Bindung:
    - zuerst Stern, dann Konkatenation, dann Vereinigung
    - Also: $\mathbf{a \circ b \cup i^* \circ a^* \cup c^* \circ k = (a \circ b) \cup ((i^*) \circ a^*) \cup (c^* \circ k)}$
    - Schöner: $= \mathbf{ab \;\cup\; i^*a^* \;\cup\; c^*k}$

Beispiele

- **Zum Warmwerden: Was ist das?**
  - `flick` ∪ `flack = fl (i` ∪ `a) ck`
  - `fidera(la)*la`
  - `0*10*`
  - `(0∪1∪2∪3∪4∪5∪6∪7∪8∪9)*(0∪5)`
- **Kniffliger:**
  - `otto` $\cup \emptyset$
  - `otto` $\circ\ \varepsilon$
  - `o`ε`t`ε`t`ε`o` $\circ \emptyset$
- **Praktisch unlösbar (oder?)**
  - $\varepsilon \cup \emptyset$
  - $\varepsilon \circ \emptyset$
  - ε^*
  - $\emptyset^*$
- **Der Knüller**
  - $\emptyset \circ (\emptyset \cup (\emptyset \circ \emptyset))^*$

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**

$$A^* = \{x_1 x_2 \ldots x_k \mid k \geq 0 \text{ und für alle } i \in \{1, \ldots, k\} : x_i \in A\}$$

Die regulären Ausdrücke beschreiben genau die reguläre Sprachen (1. Teil: ⇒)

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

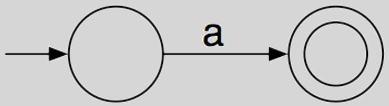
- **Lemma**
  - Jeder reguläre Ausdruck R beschreibt eine reguläre Sprache

Beweis

- Wir konvertieren R in einen NFA

1. Fall R = a, für a ∈ Σ

Automat:



– Formal:

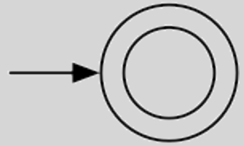
$$N = (\{q_1, q_2\}, \Sigma, \delta, q_1, \{q_2\})$$

$$\delta(q_1, a) = \{q_2\}$$

$$\delta(r, b) = \emptyset, \text{ für } r \neq q_1 \text{ oder } b \neq a$$

2. Fall R = ε

Automat:



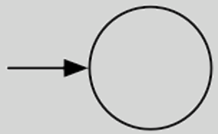
– Formal:

$$N = (\{q_1\}, \Sigma, \delta, q_1, \{q_1\})$$

$$\delta(r, b) = \emptyset, \text{ für alle } r,\ b$$

3. Fall R = ∅

Automat:



– Formal:

$$N = (\{q\}, \Sigma, \delta, q, \emptyset)$$

$$\delta(r, b) = \emptyset, \text{ für alle } r,\ b$$

4. Fall: R = ($R_1 \cup R_2$),

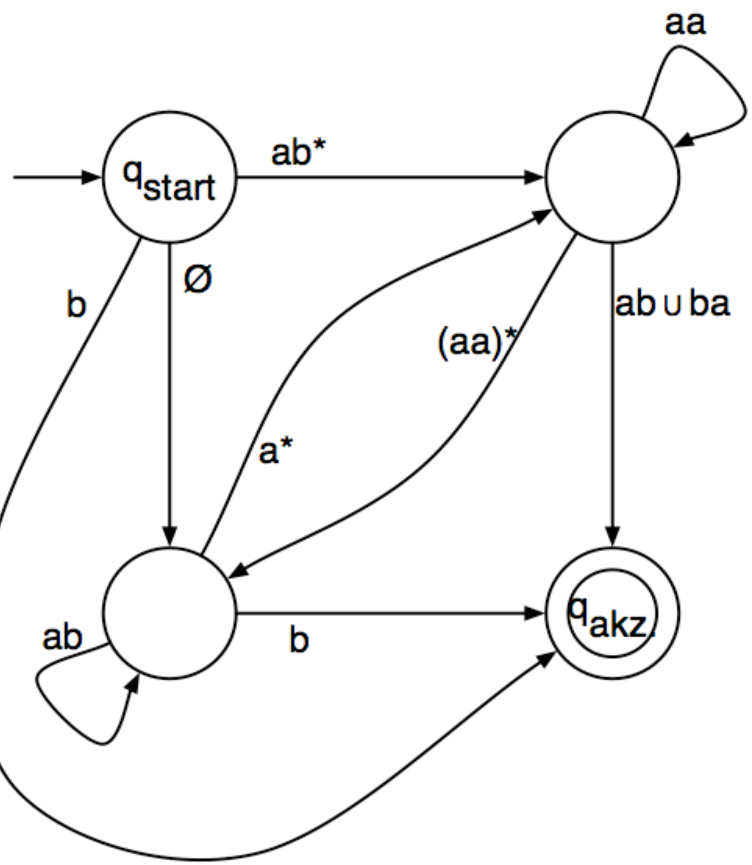
5. Fall: ($R_1 \circ R_2$),

6. Fall: (R_1)*

siehe Folien 03-3 bis 03-08

Die regulären Ausdrücke beschreiben genau die reguläre Sprachen (2. Teil: ⇐)

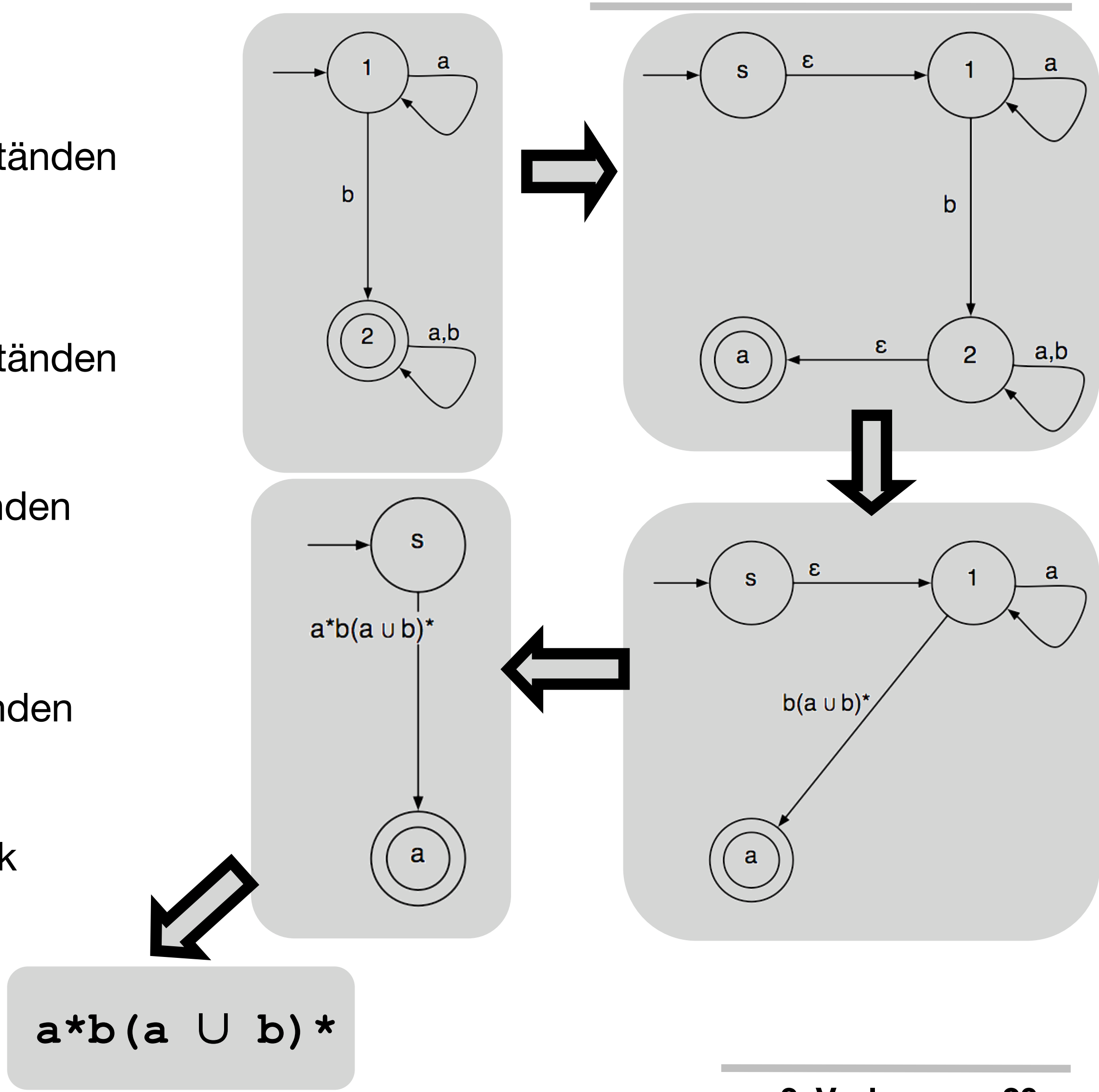
- **Strategie:**
  - Einführung der **verallgemeinerten nichtdeterministischen endlichen Automaten** (**G**eneralized **N**on-deterministic **F**inite **A**utomata - GNFA)
  - NFA → GNFA
  - GNFA → Regulärer Ausdruck
- **Eigenschaften GNFA:**
  - GNFA = NFA + reguläre Ausdrücke
  - Reguläre Ausdrücke auf den Übergängen
  - Ein akzeptierender Zustand
  - Alle Übergänge existieren
  - Ausnahmen:
    - Kein Übergang hin zum Startzustand
    - Kein Übergang ausgehend vom akzeptierenden Zustand



Die regulären Ausdrücke beschreiben genau die reguläre Sprachen (2. Teil: ⇐)

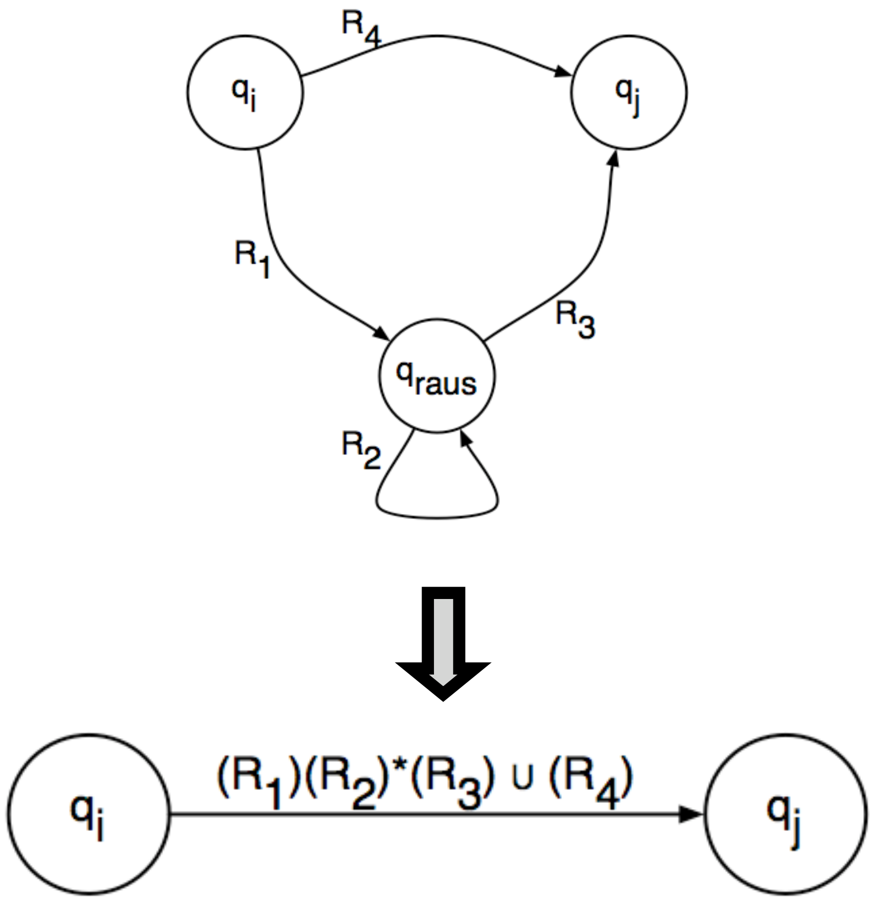
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

- **Strategie:**
  - NFA mit k Zuständen
    → GNFA mit k+2 Zuständen
  - GNFA mit k+2 Zuständen
    → GNFA mit k+1 Zuständen
  - GNFA mit k+1 Zuständen
    → GNFA mit k Zuständen
  - ...
  - GNFA mit 3 Zuständen
    → GNFA mit 2 Zuständen
  - GNFA mit 2 Zuständen
    → Regulärer Ausdruck



GNFA mit k Zuständen → GNFA mit k-1 Zuständen

- **Wie kann man einen Zustand im GNFA einsparen?**
- **Zustand q_{raus} soll raus**
- **Betrachte alle anderen Paare q_i, q_j**
- **Jeder Weg von q_i nach q_j kann entweder**
  - nach q_{raus} führen (R_1)
  - dort beliebig häufig q_{raus} die Schleife über q_{raus} nehmen $(R_2)^*$
  - dann nach q_j gehen (R_3)
- **oder**
  - überhaupt nicht über q_{raus} gehen (R_4)



Ende der 3. Vorlesung

CoNe Freiburg

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

Informatik III
Christian Schindelhauer
schindel@informatik.uni-freiburg.de