

# Endliche Automaten

Reguläre Mengen, Reguläre  
Ausdrücke, reguläre Sprachen  
und endliche Automaten

Karin Haenelt

# Zusammenhänge

- **reguläre Mengen** sind Mengen, die durch die Operationen der Vereinigung, der Konkatenation und die Bildung der Kleeneschen Hülle entstehen
- **Reguläre Ausdrücke** bilden eine Familie von formalen Sprachen. Eine formale Sprache ist durch eine Menge von Zeichenketten beschrieben.  
Reguläre Ausdrücke sind reguläre Mengen von Zeichenketten
- **Reguläre Sprachen**
  - werden durch **reguläre Ausdrücke** beschrieben
  - werden von **Grammatiken vom Typ 3** (Chomsky-Hierarchie der formalen Grammatiken) generiert
  - werden durch **endliche Automaten** erkannt

# Reguläre Mengen

- 1)  $\emptyset$  , die leere Menge ist eine reguläre Menge
- 2)  $\{e\}$  , die Menge mit dem Element  $e$  ist eine reguläre Menge
- 3) Jede endliche Menge ist eine reguläre Menge
- 4) Wenn  $R$  und  $S$  reguläre Mengen sind,  
so sind
  - $R \cup S$  Vereinigung/Disjunktion
  - $RS$  Konkatenation
  - $R^*$  Kleene'sche Hülleebenfalls reguläre Mengen

# Reguläre Ausdrücke

Sei  $\Sigma$  ein Alphabet. Die regulären Ausdrücke über  $\Sigma$  und die von ihnen beschriebenen Mengen werden wie folgt rekursiv definiert:

- 1)  $\emptyset$  ist ein regulärer Ausdruck und bezeichnet die leere Menge
- 2)  $\epsilon$  ist ein regulärer Ausdruck und bezeichnet die Menge  $\{\epsilon\}$
- 3) Für  $\forall a \in \Sigma$  ist  $a$  ein regulärer Ausdruck und bezeichnet die Menge  $\{a\}$
- 4) Wenn  $r$  und  $s$  reguläre Ausdrücke sind,

die die Sprachen  $R$  und  $S$  bezeichnen, so sind

$(r \mid s)$	Vereinigung/Disjunktion $R \cup S$	[r oder s]
$(rs)$	Konkatenation $RS$	[r gefolgt von s]
$(r^*)$	Kleene'sche Hülle $R^*$	[r null- oder mehrmals]

ebenfalls reguläre Ausdrücke

# Alphabet und Sprache über einem Alphabet

- ein Alphabet  $A$  ist eine endliche nicht-leere Menge von (unzerlegbaren Grund-) Zeichen
- $W(A)$  sei die Menge aller Ketten endlicher Länge aus Elementen von  $A$
- jede (auch die leere) Teilmenge  $L$  von  $W(A)$  heißt eine Sprache über  $A$
- Beispiele:
  - $A = \{a,b,c,d\}$ ,  $W(A) = \{e, a, b, c, d, aa, ab, ac, ad, ba, \dots, aaa, \dots\}$
  - $A =$  Menge der Wortformen einer Sprache
  - $A =$  Menge der Morpheme, Phoneme
  - $A =$  Menge der grammatikalischen Kategorien
  - $A =$  Menge der Zeichen, die in bestimmten Informationen vorkommen (z.B. \* † , 1 2 3 4 5 6 7 8 9 0 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Ä Ö Ü a b c d e f g h i j k l m n o p q r s t u v w x y z ä ö ü ß)

# Reguläre Sprachen

- Spezifikation
  - Bildung von Termen (hier: reguläre Ausdrücke) aus
    - Grundeinheiten: Alphabet
    - Operationen: Konkatenation, Vereinigung, Hülle
- Beispiel 1
  - Spezifikation: (lach | mach) (e|st|t)
  - Elemente der Sprache: *lache, lachst, lacht, mache, machst, macht*
- Beispiel 2
  - $[A-Z][a-z]^*$ ,  $[A-Z][a-z]^*$ ,  $^*[0-9]\{4\}$  in  $[A-Z][a-z]^*$ , † „ $[0-9]\{4\}$ “ in  $[A-Z][a-z]^*$
  - *Buonarroti, Michelangelo, \*1475 in Caprese , † 1564 in Roma*

# Definition der Operationen auf regulären Sprachen

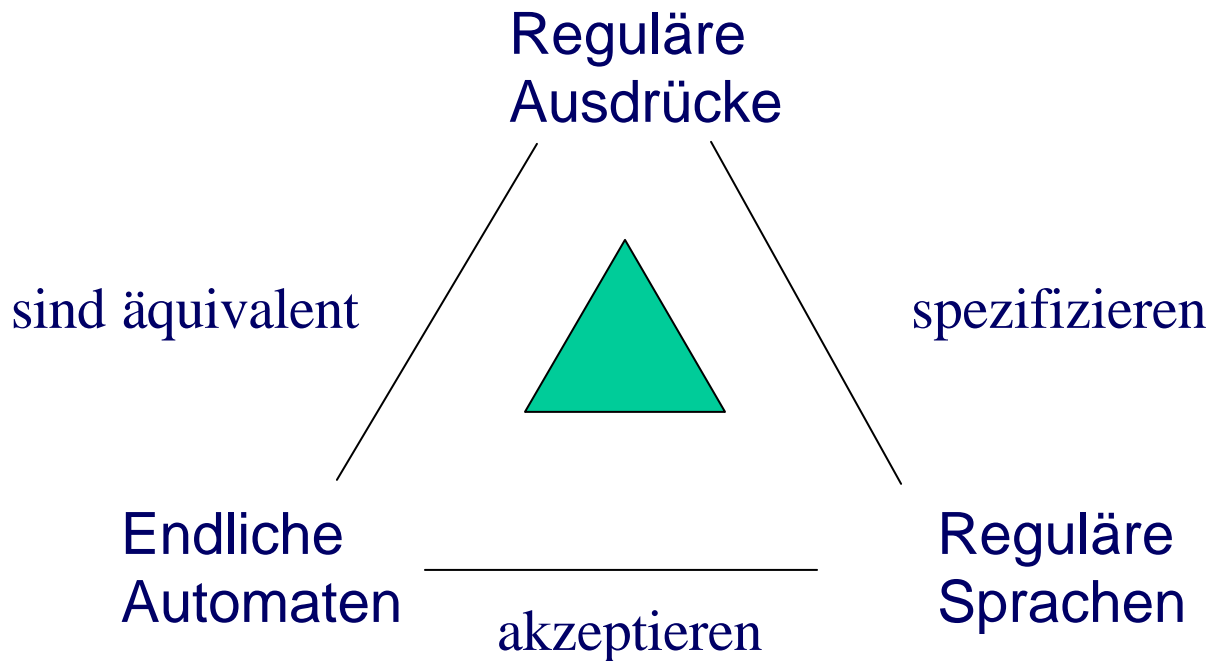
Vereinigung (Summe)	$L_1 \cup L_2 = \{p \mid p \in L_1 \vee p \in L_2\}$
Konkatenation (Produkt, Verkettung)	$L_1 \bullet L_2 = \{pq \mid p \in L_1 \wedge q \in L_2\}$  <i>Beispiel</i> $X = \{\text{ein, zwei}\}, Y = \{\text{mal, fach}\},$ $XY = \{\text{einmal, einfach, zweimal, zweifach}\}$
Potenz	$L^0 = \{\mathbf{e}\}, L^{n+1} = L \bullet L^n \text{ für } n \geq 0$
Hülle (Iteration, Stern)	$L^* = \bigcup_{n \geq 0} L^n, L^+ = \bigcup_{n \geq 1} L^n, L^* = L^+ \cup \{\mathbf{e}\}$

# Äquivalenz von endlichen Automaten und regulären Ausdrücken

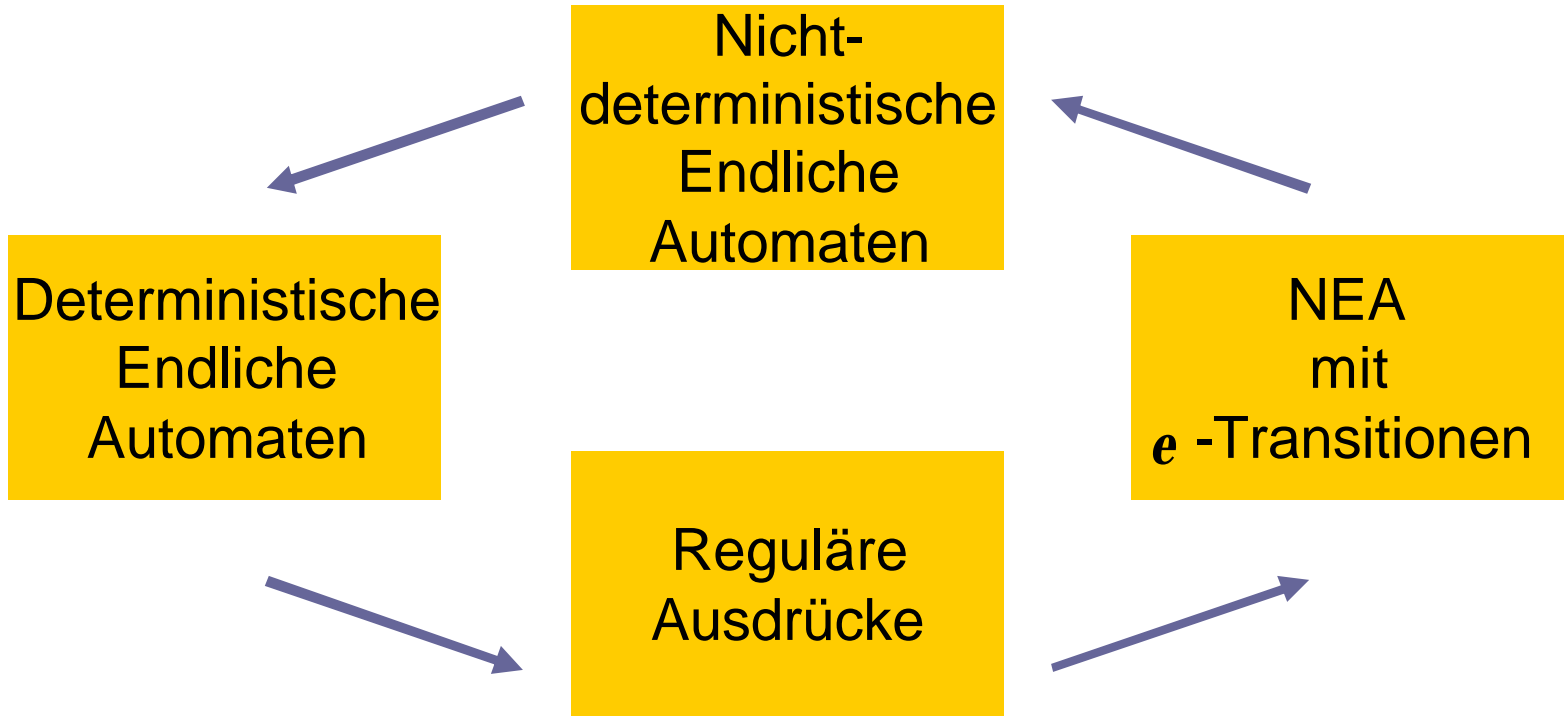
- Die von endlichen Automaten akzeptierten Sprachen sind genau die Sprachen, die durch reguläre Ausdrücke spezifiziert werden können.
- Beweis durch Konstruktion eines endlichen Automaten zu einem regulären Ausdruck
  - meistens durch Konstruktion eines NEA mit **e**-Transitionen nach Thompson (1968)
  - auch andere Verfahren bekannt

Hopcroft/Ullmann 1988:29

# Reguläre Ausdrücke, reguläre Sprachen und endliche Automaten



# Äquivalenzen



→ Konstruktion

Hopcroft/Ullmann 1988:30

# Äquivalenz von endlichen Automaten und regulären Ausdrücken

## Satz

Sei  $r$  ein regulärer Ausdruck. Dann gibt es einen NEA mit  $\epsilon$ -Transitionen, der  $L(r)$  akzeptiert

## Beweis

Wir zeigen durch *Induktion über die Struktur der Operatoren* im regulären Ausdruck, dass es einen NEA mit  $\epsilon$ -Transitionen und einem Endzustand, aus dem keine Transitionen herausführen, gibt, so dass  $L(M) = L(r)$

*Induktionsanfang:* null Operatoren

*Induktionsschritt:* einer oder mehrere Operatoren;

drei Fälle

- Vereinigung
- Konkatenation
- Hüllenbildung

Hopcroft/Ullmann 1988:30

Hopcroft/Motwani/Ullman, 2002: 112-114  
11

# Äquivalenz EA - RegAusdrücke

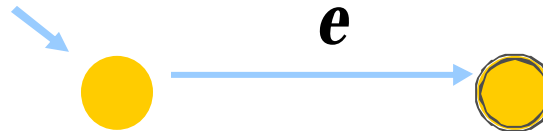
## Induktionsanfang

*Der Ausdruck muss  $\epsilon$ ,  $\emptyset$  oder  $a$  für ein  $a$  aus  $\Sigma$  sein.*

Regulärer  
Ausdruck

Automat

$$r = \epsilon$$



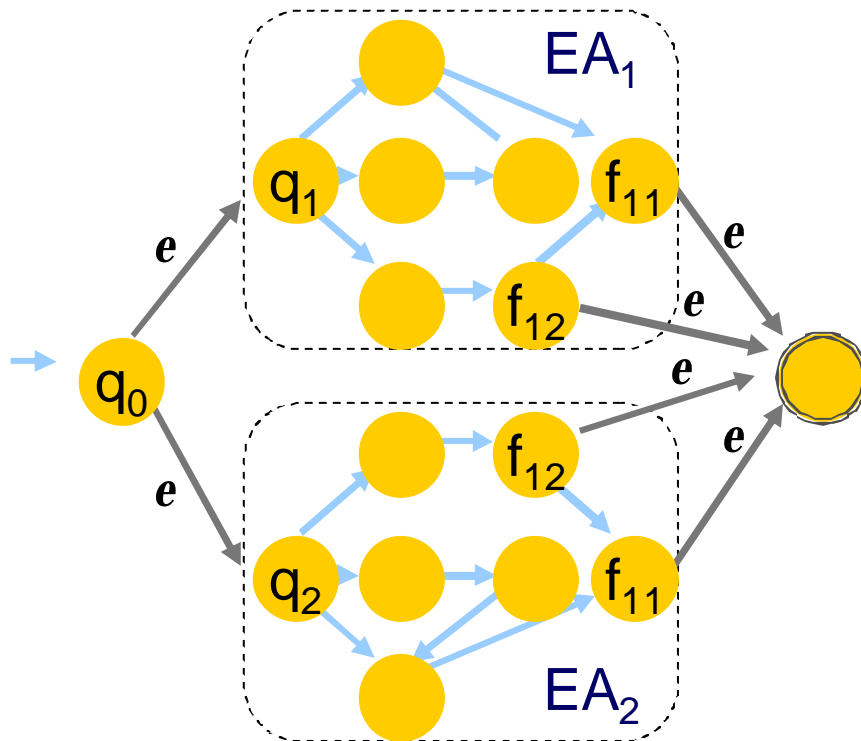
$$r = \{\}$$



$$r = a$$



# Vereinigung zweier Endlicher Automaten



$R_1 \cup R_2$

erzeuge

- Anfangszustand  $q_0$
- $e$ -Transitionen von  $q_0$  zu den Anfangszuständen von  $EA_1$  und  $EA_2$
- Endzustand  $f_0$
- $e$ -Transitionen von den Endzuständen von  $EA_1$  und  $EA_2$  zu  $f_0$

# Äquivalenz EA - RegAusdrücke

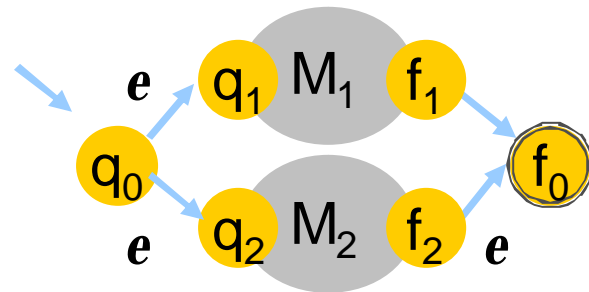
## Induktionsschritt: Vereinigung

$$R_1 \cup R_2 \quad r = r_1 \mid r_2$$

$$M_1 = (Q_1, \Sigma_1, \mathbf{d}_1, q_1, \{f_1\})$$

$$\cup \quad M_2 = (Q_2, \Sigma_2, \mathbf{d}_2, q_2, \{f_2\})$$

$$M = (Q_1 \cup Q_2 \cup \{q_0, f_0\}, \Sigma_1 \cup \Sigma_2, \mathbf{d}, q_0, \{f_0\})$$



$q_0$  neuer Anfangszustand

$f_0$  neuer Endzustand

$\mathbf{d}$  definiert durch

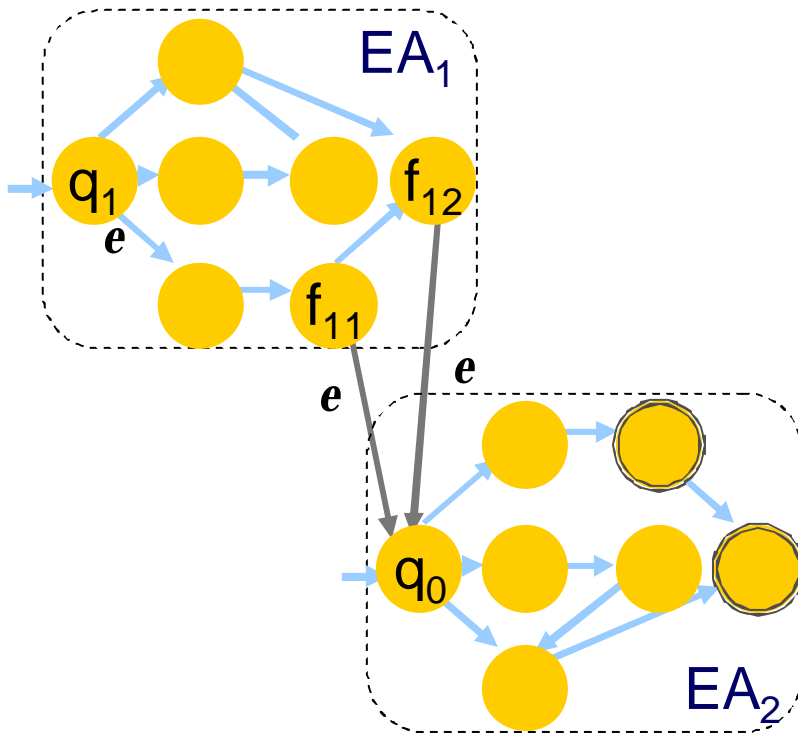
$$\mathbf{d}(q_0, \mathbf{e}) = \{q_1, q_2\}$$

$$\mathbf{d}(q, a) = \mathbf{d}_1(q, a) \text{ für } q \text{ aus } Q_1 - \{f_1\} \text{ und } a \text{ aus } \Sigma_1 \cup \mathbf{e}$$

$$\mathbf{d}(q, a) = \mathbf{d}_2(q, a) \text{ für } q \text{ aus } Q_2 - \{f_2\} \text{ und } a \text{ aus } \Sigma_2 \cup \mathbf{e}$$

$$\mathbf{d}(f_1, \mathbf{e}) = \mathbf{d}(f_2, \mathbf{e}) = \{f_0\}$$

# Konkatenation zweier Endlicher Automaten



erzeuge

$R_1R_2$

- $e$ -Transitionen von den Endzuständen von EA<sub>1</sub> zum Anfangszustand von EA<sub>2</sub>

# Äquivalenz EA - RegAusdrücke

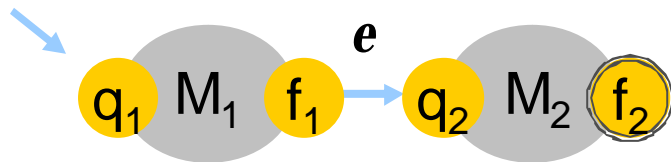
## Induktionsschritt: Konkatination

$$R_1 R_2 \quad r = r_1 r_2$$

$$M_1 = (Q_1, \Sigma_1, \mathbf{d}_1, q_1, \{f_1\})$$

$$M_2 = (Q_2, \Sigma_2, \mathbf{d}_2, q_2, \{f_2\})$$

$$M = (Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \mathbf{d}, \{q_1\}, \{f_2\})$$



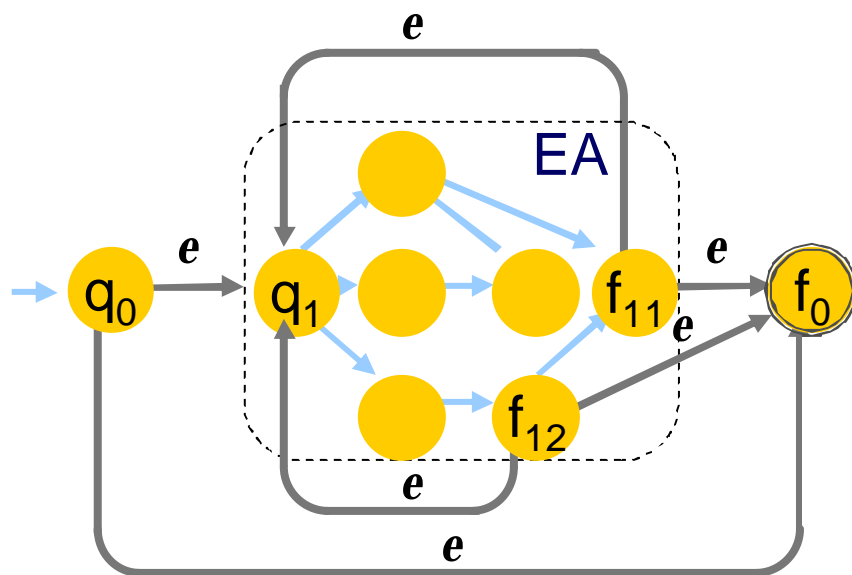
**d** definiert durch

$$\mathbf{d}(q, a) = \mathbf{d}_1(q, a) \text{ für } q \text{ aus } Q_1 - \{f_1\} \text{ und } a \text{ aus } \Sigma_1 \cup e$$

$$\mathbf{d}(q, a) = \mathbf{d}_2(q, a) \text{ für } q \text{ aus } Q_2 \text{ und } a \text{ aus } \Sigma_2 \cup e$$

$$\mathbf{d}(f_1, e) = \{q_2\}$$

# Hüllenbildung eines Endlichen Automaten



erzeuge

$R^*$

- Anfangszustand  $q_0$
- Endzustand  $f_0$
- $e$ -Transition von  $q_0$  nach  $q_1$
- $e$ -Transition von  $q_0$  nach  $f_0$   
(modelliert Möglichkeit von null Vorkommen)
- $e$ -Transition von  $F_1 = \{f_{11}, f_{12}, \dots, f_{1n}\}$  nach  $q_1$   
(modelliert Wiederholung)
- $e$ -Transition von  $F_1 = \{f_{11}, f_{12}, \dots, f_{1n}\}$  nach  $f_0$

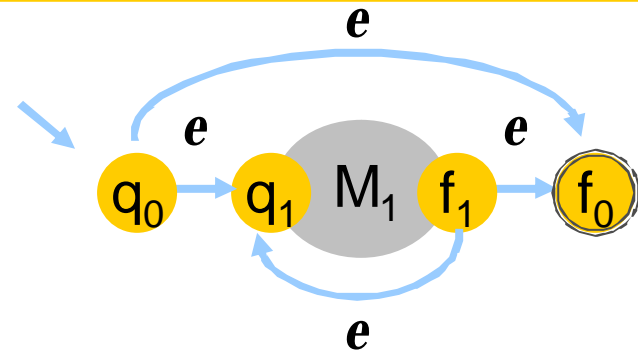
# Äquivalenz EA - RegAusdrücke

## Induktionsschritt: Hüllenbildung

$$R^* \quad r = r_1^*$$

$$M_1 = (Q_1, \Sigma_1, \mathbf{d}_1, q_1, \{f_1\})$$

$$M = (Q_1 \cup \{q_0, f_0\}, \Sigma_1, \mathbf{d}, q_0, \{f_0\})$$



**d** definiert durch

$$\mathbf{d}(q_0, \mathbf{e}) = \mathbf{d}(f_1, \mathbf{e}) = \{q_1, f_0\}$$

$$\mathbf{d}(q, a) = \mathbf{d}_1(q, a) \text{ für } q \text{ aus } Q_1 - \{f_1\} \text{ und } a \text{ aus } \Sigma_1 \cup \mathbf{e}$$

# Konstruktion eines EA für einen regulären Ausdruck

- Der Beweis der Äquivalenz ist ein Algorithmus zur Konvertierung eines regulären Ausdrucks in einen endlichen Automaten
- Für reguläre Ausdrücke ohne redundante Klammerung ist zu bestimmen, ob der Ausdruck der Form  $p|q$ ,  $pq$  oder  $p^*$  ist
  - Dies ist äquivalent zur Erkennung einer Zeichenkette einer kontextfreien Sprache
  - Algorithmus: Erzeugung mit Kellerautomat

Hopcroft/Ullmann 1988:34,47 19

# Literatur

- Hopcroft, John E. und Jeffrey D. Ullman. *Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie*. Bonn u. a. : Addison-Wesley, 1988 (engl. Original Introduction to automata theory, languages and computation).
- Kunze, Jürgen (2001). Computerlinguistik. Voraussetzungen, Grundlagen, Werkzeuge. Vorlesungsskript. Humboldt Universität zu Berlin. [http://www2.rz.hu-berlin.de/compling/Lehrstuhl/Skripte/Computerlinguistik\\_1/index.html](http://www2.rz.hu-berlin.de/compling/Lehrstuhl/Skripte/Computerlinguistik_1/index.html)
- Thompson, Ken (1968). Regular expression search algorithms. In: CACM 11(6): 419-422.
- .

# Versionen

- 22.03.2005
- 05.05., 14.04., 06.04.2004,
- 21.05., 15.01.2003