

# Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik  
Humboldt-Universität zu Berlin

WS 2013/14

## Themen dieser VL:

- Welche Rechenmodelle sind adäquat? **Automatentheorie**
- Welche Probleme sind lösbar? **Berechenbarkeitstheorie**
- Welcher Aufwand ist nötig? **Komplexitätstheorie**

## Themen der VL Algorithmen und Datenstrukturen:

- Wie lassen sich praktisch relevante Problemstellungen möglichst effizient lösen? **Algorithmik**

## Themen der VL Logik in der Informatik:

- Mathem. Grundlagen der Informatik, Beweise führen, Modellierung **Aussagenlogik, Prädikatenlogik**

- Rechenmaschinen spielen in der Informatik eine zentrale Rolle.
- Es gibt viele unterschiedliche math. Modelle.
- Diese können sich in der Berechnungskraft unterscheiden.
- Die Turingmaschine (TM) ist ein universales Berechnungsmodell, da sie alle anderen bekannten Rechenmodelle simulieren kann.
- Wir betrachten zunächst Einschränkungen des TM-Modells, die vielfältige praktische Anwendungen haben, wie z.B.
  - endliche Automaten (DFA, NFA),
  - Kellerautomaten (PDA, DPDA) etc.

# Der Algorithmenbegriff

- Der Begriff **Algorithmus** geht auf den persischen Gelehrten **Muhammed Al Chwarizmi** (8./9. Jhd.) zurück.
- Ältester bekannter nicht-trivialer Algorithmus: **Euklidischer Algorithmus** zur Berechnung des größten gemeinsamen Teilers zweier natürlicher Zahlen (300 v. Chr.).
- Von einem Algorithmus wird erwartet, dass er bei jeder zulässigen **Problemeingabe** nach endlich vielen Rechenschritten eine korrekte **Ausgabe** liefert.
- Problemeingaben können Zahlen, Formeln, Graphen etc. sein.
- Diese werden über einem Eingabealphabet  $\Sigma$  kodiert.

## Definition

- Ein **Alphabet** ist eine geordnete endliche Menge

$$\Sigma = \{a_1, \dots, a_m\}, \quad m \geq 1$$

von **Zeichen**  $a_i$ .

- Eine Folge  $x = x_1 \dots x_n \in \Sigma^n$  heißt **Wort** (der **Länge**  $n$ ).
- Die Menge aller Wörter über  $\Sigma$  ist

$$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n.$$

- Das (einzige) Wort der Länge  $n = 0$  ist das **leere Wort**, welches wir mit  $\varepsilon$  bezeichnen.
- Jede Teilmenge  $L \subseteq \Sigma^*$  heißt **Sprache** über dem Alphabet  $\Sigma$ .

## Beispiel

- Sei  $\Sigma$  ein Alphabet. Dann sind  $\emptyset, \Sigma^*, \Sigma$  und  $\{\varepsilon\}$  Sprachen über  $\Sigma$ .
  - $\emptyset$  enthält keine Wörter und heißt **leere Sprache**.  $\Sigma^*$  enthält dagegen alle Wörter über  $\Sigma$  und  $\Sigma$  enthält alle Wörter über  $\Sigma$  der Länge 1.
  - $\{\varepsilon\}$  enthält nur das leere Wort, ist also einelementig. Solche Sprachen werden auch als **Singleton-Sprachen** bezeichnet.
- 
- Da Sprachen Mengen sind, können wir sie bzgl. Inklusion vergleichen.
  - Zum Beispiel gilt  $\emptyset \subseteq \{\varepsilon\} \subseteq \Sigma^*$ .
  - Wir können Sprachen auch vereinigen, schneiden und komplementieren.
  - Seien  $A$  und  $B$  Sprachen über  $\Sigma$ . Dann ist
    - $A \cap B = \{x \in \Sigma^* \mid x \in A, x \in B\}$  der **Schnitt** von  $A$  und  $B$ ,
    - $A \cup B = \{x \in \Sigma^* \mid x \in A \vee x \in B\}$  die **Vereinigung** von  $A$  und  $B$ , und
    - $\overline{A} = \{x \in \Sigma^* \mid x \notin A\}$  das **Komplement** von  $A$ .

Neben den Mengenoperationen Schnitt, Vereinigung und Komplement gibt es auch spezielle Sprachoperationen.

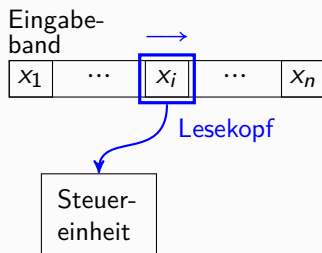
## Definition

- Das **Produkt** (**Verkettung**, **Konkatenation**) der Sprachen  $A$  und  $B$  ist  $AB = \{xy \mid x \in A, y \in B\}$ .
- Ist  $A = \{x\}$  eine Singletonsprache, so schreiben wir für  $\{x\}B$  auch einfach  $xB$ .
- Die  **$n$ -fache Potenz**  $A^n$  einer Sprache  $A$  ist induktiv definiert durch

$$A^n = \begin{cases} \{\varepsilon\}, & n = 0, \\ A^{n-1}A, & n > 0. \end{cases}$$

- Die **Sternhülle** von  $A$  ist  $A^* = \bigcup_{n \geq 0} A^n$ .
- Die **Plushülle** von  $A$  ist  $A^+ = \bigcup_{n \geq 1} A^n = AA^*$ .

## Ein endlicher Automat



- nimmt zu jedem Zeitpunkt genau einen von endlich vielen Zuständen an,
- macht bei Eingaben der Länge  $n$  genau  $n$  Rechenschritte und
- liest in jedem Schritt genau ein Eingabezeichen.



# Formale Definition eines endlichen Automaten

## Definition

Ein **endlicher Automat** (kurz: **DFA**; *deterministic finite automaton*) wird durch ein 5-Tupel  $M = (Z, \Sigma, \delta, q_0, E)$  beschrieben, wobei

- $Z \neq \emptyset$  eine **endliche** Menge von **Zuständen**,
- $\Sigma$  das **Eingabealphabet**,
- $\delta : Z \times \Sigma \rightarrow Z$  die **Überföhrungsfunktion**,
- $q_0 \in Z$  der **Startzustand** und
- $E \subseteq Z$  die Menge der **Endzustände** ist.

Die von  $M$  **akzeptierte** oder **erkannte Sprache** ist

$$L(M) = \left\{ x_1 \dots x_n \in \Sigma^* \mid \begin{array}{l} \text{es gibt } q_1, \dots, q_{n-1} \in Z, q_n \in E \text{ mit} \\ \delta(q_i, x_{i+1}) = q_{i+1} \text{ f\u00fcr } i = 0, \dots, n-1 \end{array} \right\}$$

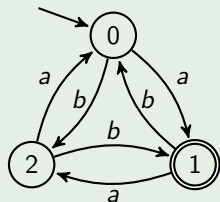
$q_0, q_1, \dots, q_n$  hei\u00dft **Rechnung** von  $M(x_1 \dots x_n)$ , falls  $\delta(q_i, x_{i+1}) = q_{i+1}$  f\u00fcr  $i = 0, \dots, n-1$  gilt. Sie hei\u00dft **akzeptierend**, falls  $q_n \in E$  ist.

## Beispiel

Sei  $M_3 = (Z, \Sigma, \delta, 0, E)$  ein DFA mit  $Z = \{0, 1, 2\}$ ,  $\Sigma = \{a, b\}$ ,  $E = \{1\}$  und der Überföhrungsfunktion

$\delta$	0	1	2
$a$	1	2	0
$b$	2	0	1

Graphische Darstellung:



Endzustände werden durch einen doppelten Kreis und der Startzustand wird durch einen Pfeil gekennzeichnet. ◀

Frage: Welche Wörter akzeptiert  $M_3$ ?

- $w_1 = aba$ ? Ja (Rechnung: 0, 1, 0, 1).
- $w_2 = abba$ ? Nein (Rechnung: 0, 1, 0, 2, 0).

### Behauptung

Die von  $M_3$  erkannte Sprache ist

$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}, \text{ wobei}$$

- $\#_a(x)$  die Anzahl der Vorkommen von  $a$  in  $x$  bezeichnet und
- $i \equiv_m j$  bedeutet, dass  $i - j$  durch  $m$  teilbar ist.

Beweis der Behauptung durch Induktion über die Länge von  $x$

Wir betrachten zunächst das Erreichbarkeitsproblem für DFAs.

## Frage

Sei  $M = (Z, \Sigma, \delta, q_0, E)$  ein DFA und sei  $x = x_1 \dots x_n \in \Sigma^*$ . Welchen Zustand erreicht  $M$  bei Eingabe  $x$  nach  $i$  Schritten?

## Antwort

- nach 0 Schritten:  $q_0$ ,
- nach 1 Schritt:  $\delta(q_0, x_1)$ ,
- nach 2 Schritten:  $\delta(\delta(q_0, x_1), x_2)$ ,
- nach  $i$  Schritten:  $\delta(\dots \delta(\delta(q_0, x_1), x_2), \dots x_i)$ .

# Das Erreichbarkeitsproblem für DFAs

## Definition

- Bezeichne  $\hat{\delta}(q, x)$  denjenigen Zustand, in dem sich  $M$  nach Lesen von  $x$  befindet, wenn  $M$  im Zustand  $q$  gestartet wird.
- Dann können wir die Funktion

$$\hat{\delta}: Z \times \Sigma^* \rightarrow Z$$

induktiv über die Länge von  $x$  wie folgt definieren.

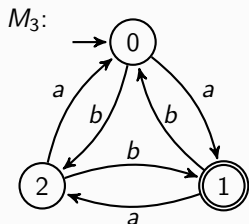
- Für  $q \in Z$ ,  $x \in \Sigma^*$  und  $a \in \Sigma$  sei

$$\begin{aligned}\hat{\delta}(q, \varepsilon) &= q, \\ \hat{\delta}(q, xa) &= \delta(\hat{\delta}(q, x), a).\end{aligned}$$

- Die von  $M$  erkannte Sprache lässt sich nun auch in der Form

$$L(M) = \{x \in \Sigma^* \mid \hat{\delta}(q_0, x) \in E\}$$

schreiben.



## Behauptung

$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}.$$

## Beweis

- 1 ist der einzige Endzustand von  $M$ .
- Daher ist  $L(M_3) = \{x \in \Sigma^* \mid \hat{\delta}(0, x) = 1\}$ .
- Obige Behauptung ist also äquivalent zu

$$\hat{\delta}(0, x) = 1 \Leftrightarrow \#_a(x) - \#_b(x) \equiv_3 1$$

- Folglich reicht es, folgende Kongruenzgleichung zu zeigen:

$$\hat{\delta}(0, x) \equiv_3 \#_a(x) - \#_b(x)$$

Beweis von  $\hat{\delta}(0, x) \equiv_3 \#_a(x) - \#_b(x)$ :

Wir führen Induktion über die Länge  $n$  von  $x$ .

**Induktionsanfang  $n = 0$ :** klar, da  $\hat{\delta}(0, \varepsilon) = \#_a(\varepsilon) = \#_b(\varepsilon) = 0$  ist.

**Induktionsschritt  $n \rightsquigarrow n + 1$ :**

- Sei  $x = x_1 \dots x_{n+1}$  gegeben und sei  $i = \hat{\delta}(0, x_1 \dots x_n)$ .
- Nach IV gilt  $i \equiv_3 \#_a(x_1 \dots x_n) - \#_b(x_1 \dots x_n)$ .
- Wegen  $\delta(i, a) \equiv_3 i + 1$  und  $\delta(i, b) \equiv_3 i - 1$  folgt daher

$$\begin{aligned} \delta(i, x_{n+1}) &\equiv_3 i + \#_a(x_{n+1}) - \#_b(x_{n+1}) \\ &\equiv_3 \#_a(x_1 \dots x_n) - \#_b(x_1 \dots x_n) + \#_a(x_{n+1}) - \#_b(x_{n+1}) \\ &= \#_a(x) - \#_b(x) \end{aligned}$$

und somit

$$\hat{\delta}(0, x) = \delta(\hat{\delta}(0, x_1 \dots x_n), x_{n+1}) = \delta(i, x_{n+1}) \equiv_3 \#_a(x) - \#_b(x).$$

## Definition

Eine von einem DFA akzeptierte Sprache wird als **regulär** bezeichnet. Die zugehörige Sprachklasse ist

$$\text{REG} = \{L(M) \mid M \text{ ist ein DFA}\}.$$

## Frage

Welche Sprachen gehören zu REG und welche nicht?



# Singletons sind regulär

## Vereinbarung

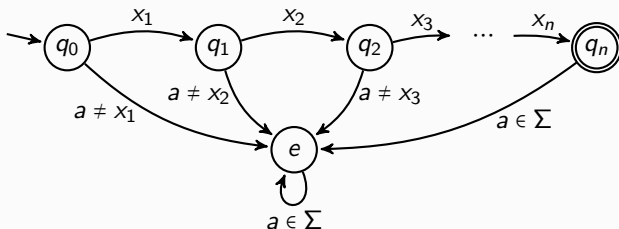
Für das Folgende sei  $\Sigma = \{a_1, \dots, a_m\}$  ein fest gewähltes Alphabet.

## Beobachtung 1

Alle Sprachen, die nur ein Wort  $x = x_1 \dots x_n \in \Sigma^*$  enthalten, sind regulär.

## Beweis

Folgender DFA  $M$  erkennt die Sprache  $L(M) = \{x\}$ :



# Abschlusseigenschaften von Sprachklassen

## Definition

- Ein ( $k$ -stelliger) **Sprachoperator** ist eine Abbildung  $op$ , die  $k$  Sprachen  $L_1, \dots, L_k$  auf eine Sprache  $op(L_1, \dots, L_k)$  abbildet.
- Eine Sprachklasse  $\mathcal{K}$  heißt unter  $op$  **abgeschlossen**, wenn gilt:

$$L_1, \dots, L_k \in \mathcal{K} \Rightarrow op(L_1, \dots, L_k) \in \mathcal{K}.$$

- Der **Abschluss** von  $\mathcal{K}$  unter  $op$  ist die (bzgl. Inklusion) kleinste Sprachklasse  $\mathcal{K}'$ , die  $\mathcal{K}$  enthält und unter  $op$  abgeschlossen ist.

## Beispiel

- Der 2-stellige Schnittoperator  $\cap$  bildet  $L_1$  und  $L_2$  auf  $L_1 \cap L_2$  ab.
- Der Abschluss der Singletonsprachen unter  $\cap$  besteht aus allen Singletonsprachen und der leeren Sprache.
- Der Abschluss der Singletonsprachen unter  $\cup$  besteht aus allen nichtleeren endlichen Sprachen.

## REG ist unter Komplement abgeschlossen

- Für eine Sprachklasse  $\mathcal{C}$  bezeichne  $co\text{-}\mathcal{C}$  die Klasse  $\{\bar{L} \mid L \in \mathcal{C}\}$  aller Komplemente von Sprachen in  $\mathcal{C}$ .
- $\mathcal{C}$  ist genau dann unter Komplementbildung abgeschlossen, wenn  $co\text{-}\mathcal{C} = \mathcal{C}$  ist.

### Beobachtung 2

Ist  $L \in \text{REG}$ , so ist auch die Sprache  $\bar{L} = \Sigma^* \setminus L$  regulär.

### Beweis

- Sei  $M = (Z, \Sigma, \delta, q_0, E)$  ein DFA mit  $L(M) = L$ .
- Dann akzeptiert der DFA

$$\bar{M} = (Z, \Sigma, \delta, q_0, Z \setminus E)$$

das Komplement  $\bar{L}$  von  $L$ .

# REG ist unter Schnitt abgeschlossen

## Beobachtung 3

Sind  $L_1, L_2 \in \text{REG}$ , so ist auch die Sprache  $L_1 \cap L_2$  regulär.

## Beweis

- Seien  $M_i = (Z_i, \Sigma, \delta_i, q_i, E_i)$ ,  $i = 1, 2$ , DFAs mit  $L(M_i) = L_i$ .
- Dann wird der Schnitt  $L_1 \cap L_2$  von dem DFA

$$M = (Z_1 \times Z_2, \Sigma, \delta, (q_1, q_2), E_1 \times E_2)$$

mit

$$\delta((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$$

erkannt.

- $M$  wird auch als **Kreuzproduktautomat** bezeichnet. □

**Beobachtung 4**

Die Vereinigung  $L_1 \cup L_2$  von regulären Sprachen  $L_1$  und  $L_2$  ist regulär.

**Beweis**

Es gilt  $L_1 \cup L_2 = \overline{(\overline{L_1} \cap \overline{L_2})}$ . □

**Frage**

Wie sieht der zugehörige DFA aus?

**Antwort**

$$M' = (Z_1 \times Z_2, \Sigma, \delta, (q_1, q_2), (E_1 \times Z_2) \cup (Z_1 \times E_2)).$$

## Korollar

Die Klasse REG der regulären Sprachen ist unter folgenden Operationen abgeschlossen:

- Komplement,
- Schnitt,
- Vereinigung.

## Folgerung

- Aus den Beobachtungen folgt, dass alle **endlichen** und alle **co-endlichen** Sprachen regulär sind.
- Da die reguläre Sprache

$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$$

weder endlich noch co-endlich ist, haben wir damit allerdings noch nicht alle regulären Sprachen erfasst.

## Wie umfangreich ist REG?

### Nächstes Ziel

Zeige, dass REG unter Produktbildung und Sternhülle abgeschlossen ist.

### Problem

Bei der Konstruktion eines DFA für das Produkt  $L_1L_2$  bereitet es Schwierigkeiten, den richtigen Zeitpunkt für den Übergang von (der Simulation von)  $M_1$  zu  $M_2$  zu finden.

### Lösungsidee

Ein **nichtdeterministischer** Automat (NFA) kann den richtigen Zeitpunkt für den Übergang „raten“.

### Verbleibendes Problem

Zeige, dass auch NFAs nur reguläre Sprachen erkennen.



# Nichtdeterministische endliche Automaten

## Definition

- Ein **nichtdet. endl. Automat** (kurz: **NFA**; *nondet. finite automaton*)

$$N = (Z, \Sigma, \Delta, Q_0, E)$$

ist genau so aufgebaut wie ein DFA, nur dass er

- eine Menge  $Q_0 \subseteq Z$  von Startzuständen hat und
- die Überföhrungsfunktion folgende Form hat:

$$\Delta : Z \times \Sigma \rightarrow \mathcal{P}(Z).$$

- Hierbei bezeichnet  $\mathcal{P}(Z)$  die **Potenzmenge** (also die Menge aller Teilmengen) von  $Z$ . Diese wird auch oft mit  $2^Z$  bezeichnet.
- Die von einem NFA  $N$  **akzeptierte** oder **erkannte Sprache** ist

$$L(N) = \left\{ x_1 \dots x_n \in \Sigma^* \mid \begin{array}{l} \exists q_0 \in Q_0, q_1, \dots, q_{n-1} \in Z, q_n \in E: \\ q_{i+1} \in \Delta(q_i, x_{i+1}) \text{ f\u00fcr } i = 0, \dots, n-1 \end{array} \right\}.$$

$q_0, q_1, \dots, q_n$  hei\u00dft **Rechnung** von  $N(x_1 \dots x_n)$ , falls  $q_0 \in Q_0$  und  $q_{i+1} \in \Delta(q_i, x_{i+1})$  f\u00fcr  $i = 0, \dots, n-1$  gilt.

- Ein NFA  $N$  kann bei einer Eingabe  $x$  also nicht nur eine, sondern mehrere verschiedene Rechnungen parallel ausführen.
- Ein Wort  $x$  gehört genau dann zu  $L(N)$ , wenn  $N(x)$  mindestens eine akzeptierende Rechnung hat.
- Im Gegensatz zu einem DFA, der jede Eingabe zu Ende liest, kann ein NFA  $N$  „stecken bleiben“.
- Dieser Fall tritt ein, wenn  $N$  in einen Zustand  $q$  gelangt, in dem er das nächste Eingabezeichen  $x_i$  wegen

$$\Delta(q, x_i) = \emptyset$$

nicht verarbeiten kann.

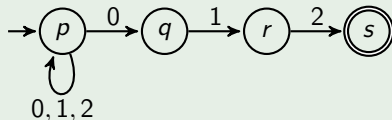
## Eigenschaften von NFAs

## Beispiel

- Betrachte den NFA  $N = (Z, \Sigma, \Delta, Q_0, E)$  mit  $Z = \{p, q, r, s\}$ ,  $\Sigma = \{0, 1, 2\}$ ,  $Q_0 = \{p\}$ ,  $E = \{s\}$  und der Überföhrungsfunktion

$\Delta$	$p$	$q$	$r$	$s$
0	$\{p, q\}$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\{p\}$	$\{r\}$	$\emptyset$	$\emptyset$
2	$\{p\}$	$\emptyset$	$\{s\}$	$\emptyset$

Graphische Darstellung:



- Dann ist  $L(M) = \{x012 \mid x \in \Sigma^*\}$  die Sprache aller W6rter, die mit dem Suffix 012 enden.



## Beobachtung 5

Seien  $N_i = (Z_i, \Sigma, \Delta_i, Q_i, E_i)$  NFAs mit  $L(N_i) = L_i$  für  $i = 1, 2$ . Dann wird auch das Produkt  $L_1 L_2$  von einem NFA erkannt.

## Beweis

- Wir können  $Z_1 \cap Z_2 = \emptyset$  annehmen.
- Dann gilt  $L(N) = L_1 L_2$  für den NFA  $N = (Z_1 \cup Z_2, \Sigma, \Delta, Q_1, E)$  mit

$$\Delta(p, a) = \begin{cases} \Delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p, a) \cup \bigcup_{q \in Q_2} \Delta_2(q, a), & p \in E_1, \\ \Delta_2(p, a), & \text{sonst} \end{cases}$$

und

$$E = \begin{cases} E_2, & Q_2 \cap E_2 = \emptyset, \\ E_1 \cup E_2, & \text{sonst.} \end{cases}$$

## Eigenschaften von NFAs

- Dann gilt  $L(N) = L_1L_2$  für den NFA  $N = (Z_1 \cup Z_2, \Sigma, \Delta, Q_1, E)$  mit

$$\Delta(p, a) = \begin{cases} \Delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p, a) \cup \bigcup_{q \in Q_2} \Delta_2(q, a), & p \in E_1, \\ \Delta_2(p, a), & \text{sonst,} \end{cases}$$

und  $E = E_2$ , falls  $Q_2 \cap E_2 = \emptyset$ , bzw.  $E = E_1 \cup E_2$  sonst.

Beweis von  $L_1L_2 \subseteq L(N)$ :

- Seien  $x = x_1 \cdots x_k \in L_1, y = y_1 \cdots y_l \in L_2$  und seien  $q_0, \dots, q_k$  und  $p_0, \dots, p_l$  akzeptierende Rechnungen von  $N_1(x)$  und  $N_2(y)$ .
- Dann gilt  $q_0 \in Q_1, q_k \in E_1$  und  $p_0 \in Q_2, p_l \in E_2$ .
- Im Fall  $l \geq 1$  ist zudem  $p_1 \in \Delta_2(p_0, y_1)$  und somit  $p_1 \in \Delta(q_k, y_1)$ .
- Im Fall  $l = 0$  ist zudem  $p_l \in Q_2 \cap E_2$  und somit  $q_k \in E$ .
- Also ist  $q_0, \dots, q_k, p_1, \dots, p_l$  eine akzeptierende Rechnung von  $N(xy)$ .

## Eigenschaften von NFAs

- Dann gilt  $L(N) = L_1L_2$  für den NFA  $N = (Z_1 \cup Z_2, \Sigma, \Delta, Q_1, E)$  mit

$$\Delta(p, a) = \begin{cases} \Delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p, a) \cup \bigcup_{q \in Q_2} \Delta_2(q, a), & p \in E_1, \\ \Delta_2(p, a), & \text{sonst,} \end{cases}$$

und  $E = E_2$ , falls  $Q_2 \cap E_2 = \emptyset$ , bzw.  $E = E_1 \cup E_2$  sonst.

Beweis von  $L(N) \subseteq L_1L_2$ :

- Sei  $x = x_1 \cdots x_n \in L(N)$  und sei  $q_0, \dots, q_n$  eine akz. Rechnung von  $N(x)$ .
- Dann gilt  $q_0 \in Q_1$ ,  $q_n \in E$ ,  $q_0, \dots, q_i \in Z_1$  und  $q_{i+1}, \dots, q_n \in Z_2$  für ein  $i$ .
- Im Fall  $i = n$  ist  $q_n \in E_1$  (d.h.  $x \in L_1$ ) und  $Q_2 \cap E_2 \neq \emptyset$  (d.h.  $\varepsilon \in L_2$ ).
- Im Fall  $i < n$  impliziert der Übergang  $q_{i+1} \in \Delta(q_i, x_{i+1})$ , dass  $q_i \in E_1$  und  $q_{i+1} \in \Delta_2(q, x_{i+1})$  für ein  $q \in Q_2$  ist.
- Also ist  $q_0, \dots, q_i$  eine akz. Rechnung von  $N_1(x_1 \cdots x_i)$  und  $q, q_{i+1}, \dots, q_n$  eine akz. Rechnung von  $N_2(x_{i+1} \cdots x_n)$ , d.h.  $x \in L_1L_2$ . □

## Eigenschaften von NFAs

## Beobachtung 6

Ist  $N = (Z, \Sigma, \Delta, Q_0, E)$  ein NFA, so wird auch die Sprache  $L(N)^*$  von einem NFA erkannt.

## Beweis

Die Sprache  $L(N)^*$  wird von dem NFA

$$N' = (Z \cup \{q_{neu}\}, \Sigma, \Delta', Q_0 \cup \{q_{neu}\}, E \cup \{q_{neu}\})$$

mit

$$\Delta'(p, a) = \begin{cases} \Delta(p, a), & p \in Z \setminus E, \\ \Delta(p, a) \cup \bigcup_{q \in Q_0} \Delta(q, a), & p \in E, \\ \emptyset, & p = q_{neu} \end{cases}$$

erkannt. □

## Ziel

Zeige, dass REG unter Produktbildung und Sternhülle abgeschlossen ist.

## Problem

Bei der Konstruktion eines DFA für das Produkt  $L_1L_2$  bereitet es Schwierigkeiten, den richtigen Zeitpunkt für den Übergang von (der Simulation von)  $M_1$  zu  $M_2$  zu finden.

## Lösungsidee (bereits umgesetzt)

Ein **nichtdeterministischer** Automat (NFA) kann den richtigen Zeitpunkt für den Übergang „raten“.

## Noch zu zeigen

NFAs erkennen genau die regulären Sprachen.



# NFAs erkennen genau die regulären Sprachen

## Satz (Rabin und Scott)

$\text{REG} = \{L(N) \mid N \text{ ist ein NFA}\}.$

### Beweis von $\text{REG} \subseteq \{L(N) \mid N \text{ ist ein NFA}\}$

Diese Inklusion ist klar, da jeder DFA  $M = (Z, \Sigma, \delta, q_0, E)$  leicht in einen äquivalenten NFA

$$N = (Z, \Sigma, \Delta, Q_0, E)$$

transformiert werden kann, indem wir  $\Delta(q, a) = \{\delta(q, a)\}$  und  $Q_0 = \{q_0\}$  setzen. □

Für die umgekehrte Inklusion ist das **Erreichbarkeitsproblem für NFAs** von zentraler Bedeutung.

# Das Erreichbarkeitsproblem für NFAs

## Frage

Sei  $N = (Z, \Sigma, \Delta, Q_0, E)$  ein NFA und sei  $x = x_1 \dots x_n$  eine Eingabe. Welche Zustände sind in  $i$  Schritten erreichbar?

## Antwort

- in 0 Schritten: alle Zustände in  $Q_0$ .
- in einem Schritt: alle Zustände in

$$Q_1 = \bigcup_{q \in Q_0} \Delta(q, x_1).$$

- in  $i$  Schritten: alle Zustände in

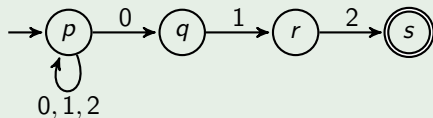
$$Q_i = \bigcup_{q \in Q_{i-1}} \Delta(q, x_i).$$

## Idee

- Wir können einen NFA  $N = (Z, \Sigma, \Delta, Q_0, E)$  durch einen DFA  $M = (Z', \Sigma, \delta, q'_0, E')$  simulieren, der in seinem Zustand die Information speichert, in welchen Zuständen sich  $N$  momentan befinden könnte.
- Die Zustände von  $M$  sind also Teilmengen  $Q$  von  $Z$  (d.h.  $Z' = \mathcal{P}(Z)$ ) mit  $Q_0$  als Startzustand (d.h.  $q'_0 = Q_0$ ) und der Endzustandsmenge  $E' = \{Q \subseteq Z \mid Q \cap E \neq \emptyset\}$ .
- Die Überföhrungsfunktion  $\delta : \mathcal{P}(Z) \times \Sigma \rightarrow \mathcal{P}(Z)$  von  $M$  berechnet dann für einen Zustand  $Q \subseteq Z$  und ein Zeichen  $a \in \Sigma$  die Menge 
$$\delta(Q, a) = \bigcup_{q \in Q} \Delta(q, a)$$
 aller Zustände, in die  $N$  gelangen kann, wenn  $N$  ausgehend von einem beliebigen Zustand  $q \in Q$  das Zeichen  $a$  liest.

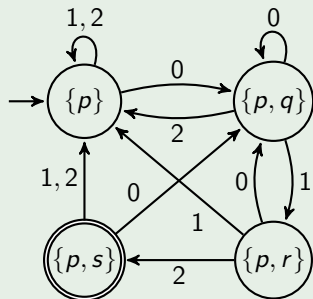
## Beispiel

- Betrachte den NFA  $N$



- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:

$\delta$	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$
$\{p, r\}$	$\{p, q\}$	$\{p\}$	$\{p, s\}$
$\{p, s\}$	$\{p, q\}$	$\{p\}$	$\{p\}$



**Bemerkung**

- Im obigen Beispiel werden für die Konstruktion des Potenzmengenautomaten nur 4 der insgesamt

$$\|\mathcal{P}(Z)\| = 2^{\|Z\|} = 2^4 = 16$$

Zustände benötigt, da die übrigen 12 Zustände nicht erreichbar sind.

- Es gibt jedoch Beispiele, bei denen alle  $2^{\|Z\|}$  Zustände benötigt werden (siehe Übungen).

# NFAs erkennen genau die regulären Sprachen

## Beweis von $\{L(N) \mid N \text{ ist ein NFA}\} \subseteq \text{REG}$

- Sei  $N = (Z, \Sigma, \Delta, Q_0, E)$  ein NFA und sei  $M = (\mathcal{P}(Z), \Sigma, \delta, Q_0, E')$  der zugehörige **Potenzmengenautomat** mit  $\delta(Q, a) = \bigcup_{q \in Q} \Delta(q, a)$  und  $E' = \{Q \subseteq Z \mid Q \cap E \neq \emptyset\}$ .
- Dann folgt die Korrektheit von  $M$  leicht mittels folgender Behauptung, die wir auf der nächsten Folie beweisen.

### Behauptung

$\hat{\delta}(Q_0, x)$  enthält genau die von  $N$  nach Lesen von  $x$  erreichbaren Zustände.

- Für alle Wörter  $x \in \Sigma^*$  gilt
  - $x \in L(N) \iff N \text{ kann nach Lesen von } x \text{ einen Endzustand erreichen}$
  - $\stackrel{\text{Beh.}}{\iff} \hat{\delta}(Q_0, x) \cap E \neq \emptyset$
  - $\iff \hat{\delta}(Q_0, x) \in E'$
  - $\iff x \in L(M).$

## Beweis der Behauptung

## Behauptung

$\hat{\delta}(Q_0, x)$  enthält genau die von  $N$  nach Lesen von  $x$  erreichbaren Zustände.

Beweis durch Induktion über die Länge  $n$  von  $x$

$n = 0$ : klar, da  $\hat{\delta}(Q_0, \varepsilon) = Q_0$  ist.

$n - 1 \rightsquigarrow n$ : Sei  $x = x_1 \dots x_n$  gegeben. Nach IV enthält

$$Q_{n-1} = \hat{\delta}(Q_0, x_1 \dots x_{n-1})$$

die Zustände, die  $N$  nach Lesen von  $x_1 \dots x_{n-1}$  erreichen kann. Wegen

$$\hat{\delta}(Q_0, x) = \delta(Q_{n-1}, x_n) = \bigcup_{q \in Q_{n-1}} \Delta(q, x_n)$$

enthält dann aber  $\hat{\delta}(Q_0, x)$  die Zustände, die  $N$  nach Lesen von  $x$  erreichen kann. □

## Korollar

Die Klasse REG der regulären Sprachen ist unter folgenden Operationen abgeschlossen:

- Komplement,
- Schnitt,
- Vereinigung,
- Produkt,
- Sternhülle.



## Nächstes Ziel

Zeige, dass REG als Abschluss der endl. Sprachen unter Vereinigung, Produkt und Sternhülle charakterisierbar ist.

## Bereits gezeigt:

Jede Sprache, die mittels der Operationen Vereinigung, Produkt und Sternhülle (sowie Schnitt und Komplement) angewandt auf endliche Sprachen darstellbar ist, ist regulär.

## Noch zu zeigen:

Jede reguläre Sprache lässt sich aus endlichen Sprachen mittels Vereinigung, Produkt und Sternhülle erzeugen.

Induktive Definition der Menge  $RA_{\Sigma}$  aller regulären Ausdrücke über  $\Sigma$

Die Symbole  $\emptyset$ ,  $\epsilon$  und  $a$  ( $a \in \Sigma$ ) sind reguläre Ausdrücke über  $\Sigma$ , die

- die leere Sprache  $L(\emptyset) = \emptyset$ ,
- die Sprache  $L(\epsilon) = \{\epsilon\}$  und
- für jedes  $a \in \Sigma$  die Sprache  $L(a) = \{a\}$  beschreiben.

Sind  $\alpha$  und  $\beta$  reguläre Ausdrücke über  $\Sigma$ , die die Sprachen  $L(\alpha)$  und  $L(\beta)$  beschreiben, so sind auch  $\alpha\beta$ ,  $(\alpha|\beta)$  und  $(\alpha)^*$  reguläre Ausdrücke über  $\Sigma$ , die folgende Sprachen beschreiben:

- $L(\alpha\beta) = L(\alpha)L(\beta)$ ,
- $L(\alpha|\beta) = L(\alpha) \cup L(\beta)$ ,
- $L((\alpha)^*) = L(\alpha)^*$ .

**Bemerkung**

$RA_{\Sigma}$  ist eine Sprache über dem Alphabet  $\Gamma = \Sigma \cup \{\emptyset, \epsilon, |, \cup, *, (, )\}$ .

# Reguläre Ausdrücke

## Beispiel

Die regulären Ausdrücke  $(\epsilon)^*$ ,  $(\emptyset)^*$ ,  $(0|1)^*00$  und  $(\epsilon 0|\emptyset(1)^*)$  beschreiben folgende Sprachen:

$\gamma$	$(\epsilon)^*$	$(\emptyset)^*$	$(0 1)^*00$	$(0 (\epsilon 0 \emptyset(1)^*))$
$L(\gamma)$	$\{\epsilon\}$	$\{\epsilon\}$	$\{x00 \mid x \in \{0, 1\}^*\}$	$\{0\}$



## Vereinbarungen

- Um Klammern zu sparen, definieren wir folgende **Präzedenzordnung**: Der Sternoperator  $*$  bindet stärker als der Produktoperator und dieser wiederum stärker als der Vereinigungsoperator.
- Für  $(0|(\epsilon 0|\emptyset(1)^*))$  können wir also kurz  $0|\epsilon 0|\emptyset 1^*$  schreiben.
- Da der reguläre Ausdruck  $\gamma\gamma^*$  die Sprache  $L(\gamma)^+$  beschreibt, verwenden wir  $\gamma^+$  als Abkürzung für den Ausdruck  $\gamma\gamma^*$ .

## Satz

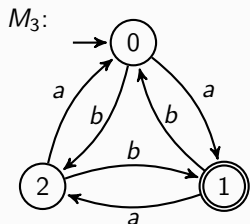
$\{L(\gamma) \mid \gamma \text{ ist ein regulärer Ausdruck über } \Sigma\} \subseteq \text{REG}.$

## Beweis.

Klar, da

- die Basisausdrücke  $\emptyset$ ,  $\epsilon$  und  $a$ ,  $a \in \Sigma^*$ , nur reguläre Sprachen beschreiben und
- die Sprachklasse REG unter Produkt, Vereinigung und Sternhülle abgeschlossen ist.





## Frage

Wie lässt sich die Sprache

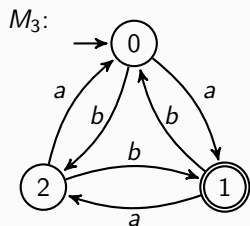
$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$$

durch einen regulären Ausdruck beschreiben?

## Antwort

- Sei  $L_{p,q}$  die Sprache aller Wörter  $x$ , die  $M_3$  vom Zustand  $p$  in den Zustand  $q$  überführen (d.h.  $\hat{\delta}(p, x) = q$ ).
- Weiter sei  $L_{p,q}^{\neq r}$  die Sprache aller Wörter  $x = x_1 \cdots x_n \in L_{p,q}$ , die hierzu nur Zustände ungleich  $r$  benutzen (d.h.  $\hat{\delta}(p, x_1 \cdots x_i) \neq r$  für  $i = 1, \dots, n-1$ ).
- Dann gilt  $L(M_3) = L_{0,1} = L_{0,0} L_{0,1}^{\neq 0}$  und  $L_{0,0} = (L_{0,0}^{\neq 0})^*$ , also  $L(M_3) = (L_{0,0}^{\neq 0})^* L_{0,1}^{\neq 0}$ .

## Antwort (Fortsetzung)



- Dann gilt  $L(M_3) = (L_{0,0}^{\neq 0})^* L_{0,1}^{\neq 0}$ .
- $L_{0,1}^{\neq 0}$  und  $L_{0,0}^{\neq 0}$  lassen sich durch folgende reguläre Ausdrücke beschreiben:

$$\gamma_{0,1}^{\neq 0} = (a|bb)(ab)^*$$

$$\gamma_{0,0}^{\neq 0} = a(ab)^*(aa|b) | b(ba)^*(a|bb) | \varepsilon.$$

- Also ist  $L(M_3)$  durch folgenden regulären Ausdruck beschreibbar:

$$\gamma_{0,1} = (a(ab)^*(aa|b) | b(ba)^*(a|bb))^* (a|bb)(ab)^*.$$

## Satz

$$\text{REG} \subseteq \{L(\gamma) \mid \gamma \text{ ist ein regulärer Ausdruck}\}.$$

## Beweis

- Wir konstruieren zu einem DFA  $M = (Z, \Sigma, \delta, q_0, E)$  einen regulären Ausdruck  $\gamma$  mit  $L(\gamma) = L(M)$ .
- Wir nehmen an, dass  $Z = \{1, \dots, m\}$  und  $q_0 = 1$  ist.
- Dann lässt sich  $L(M)$  als Vereinigung

$$L(M) = \bigcup_{q \in E} L_{1,q}$$

von Sprachen der Form  $L_{p,q} = \{x \in \Sigma^* \mid \hat{\delta}(p, x) = q\}$  darstellen.

- Es reicht also, reguläre Ausdrücke für die Sprachen  $L_{p,q}$  mit  $1 \leq p, q \leq m$  anzugeben.

## Satz

$$\text{REG} \subseteq \{L(\gamma) \mid \gamma \text{ ist ein regulärer Ausdruck}\}.$$

## Beweis (Fortsetzung)

- Es reicht also, reguläre Ausdrücke für die Sprachen  $L_{p,q}$  mit  $1 \leq p, q \leq m$  anzugeben.
- Hierzu betrachten wir für  $r = 0, \dots, m$  die Sprachen

$$L_{p,q}^r = \{x \in L_{p,q} \mid \text{für } i = 1, \dots, n-1 \text{ ist } \hat{\delta}(p, x_1 \dots x_i) \leq r\}.$$

- Wegen  $L_{p,q} = L_{p,q}^m$  reicht es, reguläre Ausdrücke für die Sprachen  $L_{p,q}^r$  mit  $1 \leq p, q \leq m$  und  $0 \leq r \leq m$  anzugeben.
- Wir zeigen induktiv über  $r$ , dass die Sprachen  $L_{p,q}^r$  durch reguläre Ausdrücke beschreibbar sind.



## Satz

$$\text{REG} \subseteq \{L(\gamma) \mid \gamma \text{ ist ein regulärer Ausdruck}\}.$$

## Beweis (Schluss)

- Wir zeigen induktiv über  $r$ , dass die Sprachen  $L_{p,q}^r$  durch reguläre Ausdrücke beschreibbar sind.

$r = 0$ : In diesem Fall sind die Sprachen

$$L_{p,q}^0 = \begin{cases} \{a \in \Sigma \mid \delta(p, a) = q\}, & p \neq q, \\ \{a \in \Sigma \mid \delta(p, a) = q\} \cup \{\varepsilon\}, & \text{sonst} \end{cases}$$

endlich und somit durch reguläre Ausdrücke beschreibbar.

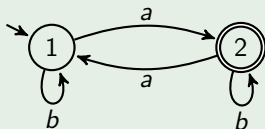
$r \rightsquigarrow r + 1$ : Wegen

$$L_{p,q}^{r+1} = L_{p,q}^r \cup L_{p,r+1}^r (L_{r+1,r+1}^r)^* L_{r+1,q}^r$$

sind mit  $L_{p,q}^r$ ,  $1 \leq p, q \leq m$ , auch die Sprachen  $L_{p,q}^{r+1}$ ,  $1 \leq p, q \leq m$ , durch reguläre Ausdrücke beschreibbar.  $\square$

## Beispiel

- Betrachte den DFA  $M$



- Da  $M$  insgesamt  $m = 2$  Zustände und nur den Endzustand 2 besitzt, ist

$$L(M) = \bigcup_{q \in E} L_{1,q} = L_{1,2} = L_{1,2}^2.$$

## Beispiel (Fortsetzung)

- Um reguläre Ausdrücke  $\gamma_{p,q}^r$  für die Sprachen  $L_{p,q}^r$  zu bestimmen, benutzen wir für  $r \geq 0$  die Rekursionsformel

$$\gamma_{p,q}^{r+1} = \gamma_{p,q}^r | \gamma_{p,r+1}^r (\gamma_{r+1,r+1}^r)^* \gamma_{r+1,q}^r.$$

- Damit erhalten wir

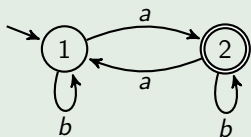
$$\gamma_{1,2}^2 = \gamma_{1,2}^1 | \gamma_{1,2}^1 (\gamma_{2,2}^1)^* \gamma_{2,2}^1,$$

$$\gamma_{1,2}^1 = \gamma_{1,2}^0 | \gamma_{1,1}^0 (\gamma_{1,1}^0)^* \gamma_{1,2}^0,$$

$$\gamma_{2,2}^1 = \gamma_{2,2}^0 | \gamma_{2,1}^0 (\gamma_{1,1}^0)^* \gamma_{1,2}^0.$$

- Es genügt also, die regulären Ausdrücke  $\gamma_{1,1}^0$ ,  $\gamma_{1,2}^0$ ,  $\gamma_{2,1}^0$ ,  $\gamma_{2,2}^0$ ,  $\gamma_{1,2}^1$ ,  $\gamma_{2,2}^1$  und  $\gamma_{1,2}^2$  zu berechnen.

## Beispiel (Fortsetzung)

DFA  $M$ 

## Rekursionsformeln

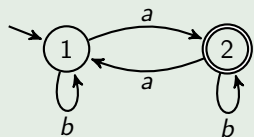
$$L_{p,p}^0 = \{a \mid \delta(p, a) = p\} \cup \{\varepsilon\},$$

$$L_{p,q}^0 = \{a \mid \delta(p, a) = q\} \text{ für } p \neq q,$$

$$L_{p,q}^{r+1} = L_{p,q}^r \cup L_{p,r+1}^r (L_{r+1,r+1}^r)^* L_{r+1,q}^r.$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0				
1				
2				

## Beispiel (Fortsetzung)

DFA  $M$ 

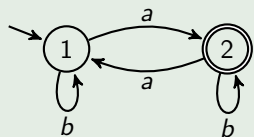
Rekursionsformeln

$$L_{1,1}^0 = \{a \in \Sigma \mid \delta(1, a) = 1\} \cup \{\epsilon\} = \{\epsilon, b\}$$

$$\leadsto \gamma_{1,1}^0 = \epsilon|b$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon b$			
1				
2				

## Beispiel (Fortsetzung)

DFA  $M$ 

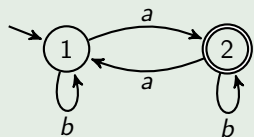
Rekursionsformeln

$$L_{1,2}^0 = \{a \in \Sigma \mid \delta(1, a) = 2\} = \{a\}$$

$$\leadsto \gamma_{1,2}^0 = a$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon b$	$a$		
1				
2				

## Beispiel (Fortsetzung)

DFA  $M$ 

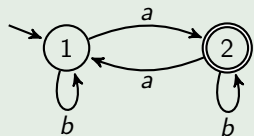
## Rekursionsformeln

$$L_{2,1}^0 = \{a \in \Sigma \mid \delta(2, a) = 1\} = \{a\}$$

$$\leadsto \gamma_{2,1}^0 = a$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon b$	a	a	
1				
2				

## Beispiel (Fortsetzung)

DFA  $M$ 

## Rekursionsformeln

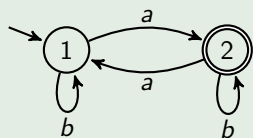
$$L_{2,2}^0 = \{a \in \Sigma \mid \delta(2, a) = 2\} \cup \{\epsilon\} = \{\epsilon, b\}$$

$$\leadsto \gamma_{2,2}^0 = \epsilon|b$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon b$	$a$	$a$	$\epsilon b$
1	-			
2				



## Beispiel (Fortsetzung)

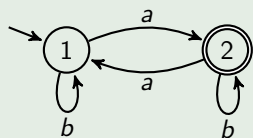
DFA  $M$ 

Rekursionsformeln

$$\begin{aligned}
 \gamma_{1,2}^1 &= \gamma_{1,2}^0 | \gamma_{1,1}^0 (\gamma_{1,1}^0)^* \gamma_{1,2}^0 \\
 &= a | (\epsilon | b) (\epsilon | b)^* a \\
 &\equiv b^* a
 \end{aligned}$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon   b$	$a$	$a$	$\epsilon   b$
1	-	$b^* a$		
2				

## Beispiel (Fortsetzung)

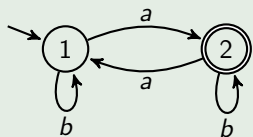
DFA  $M$ 

Rekursionsformeln

$$\begin{aligned}
 r_{2,2}^1 &= r_{2,2}^0 | r_{2,1}^0 (r_{1,1}^0)^* r_{1,2}^0 \\
 &= (\epsilon | b) | a (\epsilon | b)^* a \\
 &\equiv \epsilon | b | ab^* a
 \end{aligned}$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon   b$	$a$	$a$	$\epsilon   b$
1	-	$b^* a$	-	$\epsilon   b   ab^* a$
2	-			

## Beispiel (Fortsetzung)

DFA  $M$ 

Rekursionsformeln

$$\begin{aligned}
 \gamma_{1,2}^2 &= \gamma_{1,2}^1 | \gamma_{1,2}^1 (\gamma_{2,2}^1)^* \gamma_{2,2}^1 \\
 &= b^* a | b^* a (\epsilon | b | ab^* a)^* (\epsilon | b | ab^* a) \\
 &\equiv b^* a (b | ab^* a)^*
 \end{aligned}$$

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon   b$	$a$	$a$	$\epsilon   b$
1	-	$b^* a$	-	$\epsilon   b   ab^* a$
2	-	$b^* a (b   ab^* a)^*$	-	-

# Charakterisierungen der Klasse REG

## Korollar

Sei  $L$  eine Sprache. Dann sind folgende Aussagen äquivalent:

- $L$  ist regulär,
- es gibt einen DFA  $M$  mit  $L = L(M)$ ,
- es gibt einen NFA  $N$  mit  $L = L(N)$ ,
- es gibt einen regulären Ausdruck  $\gamma$  mit  $L = L(\gamma)$ ,
- $L$  lässt sich mit den Operationen Vereinigung, Produkt und Sternhülle aus endlichen Sprachen gewinnen,
- $L$  lässt sich mit den Operationen  $\cap$ ,  $\cup$ , Komplement, Produkt und Sternhülle aus endlichen Sprachen gewinnen.

## Ausblick

- Als nächstes wenden wir uns der Frage zu, wie sich die Anzahl der Zustände eines DFA minimieren lässt.
- Da hierbei Äquivalenzrelationen eine wichtige Rolle spielen, befassen wir uns zunächst mit Relationalstrukturen.