

# Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik  
Humboldt-Universität zu Berlin

WS 2020/21

## Themen dieser VL:

- Welche Rechenmodelle eignen sich zur Lösung welcher algorithmischen Problemstellungen? **Automatentheorie**
- Welche algorithmischen Probleme sind überhaupt lösbar? **Berechenbarkeitstheorie**
- Welcher Aufwand ist zur Lösung eines geg. algorithmischen Problems 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**

- Überblick über die wichtigsten Rechenmodelle (Automaten) wie z.B.
  - endliche Automaten
  - Kellerautomaten
  - Turingmaschinen
  - Registermaschinen
  - Schaltkreise
- Charakterisierung der Klassen aller mit diesen Rechenmodellen lösbarer Probleme durch
  - unterschiedliche Typen von formalen Grammatiken
  - Abschlusseigenschaften unter geeigneten Sprachoperationen
  - Reduzierbarkeit auf typische Probleme (Vollständigkeit)
- Erkennen von Grenzen der Berechenbarkeit
- Klassifikation wichtiger algorithmischer Probleme nach ihrer Komplexität

- Rechenmaschinen spielen in der Informatik eine zentrale Rolle
- Es gibt viele unterschiedliche math. Modelle für Rechenmaschinen
- Diese können sich in ihrer 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 Begriff **Algorithmus** geht auf den persischen Gelehrten **Muhammed Al Chwarizmi** (8./9. Jhd.) zurück
- Ältester bekannter nicht-trivialer Algorithmus: **Euklidischer Algorithmus** zur Berechnung des ggT (300 v. Chr.)
- Von einem Algorithmus wird erwartet, dass er bei jeder zulässigen **Problemeingabe** nach endlich vielen Rechenschritten eine korrekte **Ausgabe** liefert
- Eine wichtige Rolle spielen Entscheidungsprobleme, bei denen jede Eingabe nur mit ja oder nein beantwortet wird
- Die (maximale) Anzahl der Rechenschritte bei allen möglichen Eingaben ist nicht beschränkt, d.h. mit wachsender Eingabelänge kann auch die Rechenzeit beliebig anwachsen
- Die Beschreibung eines Algorithmus muss jedoch endlich sein
- Problemeingaben können Zahlen, Formeln, Graphen etc. sein
- Diese werden über einem Eingabealphabet  $\Sigma$  kodiert

## Definition

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

$$\Sigma = \{a_1, \dots, a_m\}$$

von  $m \geq 1$  **Zeichen**  $a_1 < \dots < a_m$

- Eine Folge  $x = x_1 \dots x_n$  von  $n \geq 0$  Zeichen  $x_i \in \Sigma$  heißt **Wort** der **Länge**  $n$  über  $\Sigma$
- Die Länge von  $x$  wird mit  $|x|$  und die Menge aller Wörter der Länge  $n$  über  $\Sigma$  wird mit  $\Sigma^n$  bezeichnet
- Die Menge aller Wörter über  $\Sigma$  ist

$$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$$

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

## Beispiel

- Sprachen über  $\Sigma$  sind beispielsweise  $\emptyset$ ,  $\Sigma^*$ ,  $\Sigma$  und  $\{\varepsilon\}$
- $\emptyset$  enthält keine Wörter und heißt **leere Sprache**
- $\Sigma^*$  enthält dagegen alle Wörter über  $\Sigma$
- $\Sigma$  enthält alle Wörter über  $\Sigma$  der Länge 1
- $\{\varepsilon\}$  enthält nur das leere Wort, ist also einelementig
- Sprachen, die genau ein Wort enthalten, werden auch als **Singletonsprachen** bezeichnet
- in der Informatik spielen Programmiersprachen eine wichtige Rolle

- 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
  - $A \cap B = \{x \in \Sigma^* \mid x \in A \wedge x \in B\}$  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  $A$



## Definition

Die **Konkatenation** von zwei Wörtern  $x = x_1 \dots x_n$  und  $y = y_1 \dots y_m$  ist das Wort  $x \circ y = x_1 \dots x_n y_1 \dots y_m$ , das wir auch einfach mit  $xy$  bezeichnen

## Beispiel

- Für  $x = aba$  und  $y = abab$  erhalten wir  $xy = abaabab$  und  $yx = abababa$
- Die Konkatenation ist also nicht kommutativ
- Allerdings ist  $\circ$  assoziativ, d.h. es gilt  $x(yz) = (xy)z$   
Daher können wir hierfür auch einfach  $xyz$  schreiben
- Es gibt auch ein neutrales Element, da  $x\varepsilon = \varepsilon x = x$  ist
- Eine algebraische Struktur  $(M, \square, e)$  mit einer assoziativen Operation  $\square : M \times M \rightarrow M$  und einem neutralen Element  $e$  heißt **Monoid**
- $(\Sigma^*, \circ, \varepsilon)$  ist also ein Monoid

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

## Definition

- Das **Produkt** (**Verkettung**, **Konkatenation**) von zwei Sprachen  $A$  und  $B$  ist

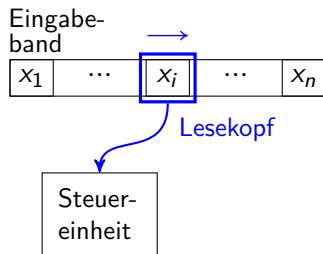
$$AB = \{xy \mid x \in A, y \in B\}$$

- 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** einer Sprache  $A$  ist  $A^* = \bigcup_{n \geq 0} A^n$
- Die **Plushülle** einer Sprache  $A$  ist  $A^+ = \bigcup_{n \geq 1} A^n = AA^*$

- Ein einfaches Rechenmodell zum Erkennen von Sprachen ist der endliche Automat:



- 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

## 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

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

- Eine Zustandsfolge  $q_0, q_1, \dots, q_n$  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, und andernfalls **verwerfend**

## Frage

Welche Sprachen lassen sich durch endliche Automaten erkennen und welche nicht?

## Definition

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

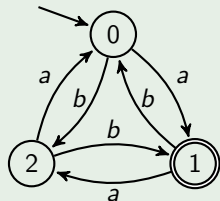
$$\text{REG} = \{L(M) \mid M \text{ ist ein DFA}\}$$

## 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  $M_3$ ?

- Ist  $w_1 = aba \in L(M_3)$ ? Ja (akzeptierende Rechnung: 0, 1, 0, 1)
- Ist  $w_2 = abba \in L(M_3)$ ? Nein (verwerfende 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
- $i \equiv_m j$  (in Worten:  $i$  ist kongruent zu  $j$  modulo  $m$ ) 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  $M$  nach Lesen der Eingabe  $x$ ?

## Antwort

- nach 0 Schritten: den Startzustand  $q_0$
- nach 1 Schritt: den Zustand  $\delta(q_0, x_1)$
- nach 2 Schritten: den Zustand  $\delta(\delta(q_0, x_1), x_2)$
- $\vdots$
- nach  $n$  Schritten: den Zustand  $\delta(\dots \delta(\delta(q_0, x_1), x_2), \dots x_n)$



# 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

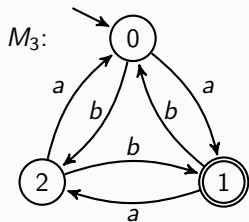
$$\begin{aligned} \hat{\delta}(q, \varepsilon) &= q, \\ \hat{\delta}(q, xa) &= \delta(\hat{\delta}(q, x), a) \end{aligned}$$

- Die von  $M$  erkannte Sprache lässt sich nun elegant durch

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

beschreiben

## DFAs beherrschen Modulare Arithmetik



## Behauptung

Für alle  $x \in \{a, b\}^*$  gilt:

$$x \in L(M_3) \Leftrightarrow \#_a(x) - \#_b(x) \equiv_3 1$$

## Beweis

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

für alle  $x \in \{a, b\}^*$  gilt:  $\hat{\delta}(0, x) = 1 \Leftrightarrow \#_a(x) - \#_b(x) \equiv_3 1$
- Folglich reicht es, für alle  $x \in \{a, b\}^*$  folgende Kongruenz zu zeigen:

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

**Induktionsbehauptung:** Für alle  $x \in \{a, b\}^n$  gilt  $\hat{\delta}(0, x) \equiv_3 \#_a(x) - \#_b(x)$

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

**Induktionsschritt ( $n \rightsquigarrow n + 1$ ):** Sei  $x = x_1 \dots x_{n+1} \in \{a, b\}^{n+1}$  gegeben

- Nach Induktionsvoraussetzung (IV) gilt für  $x' = x_1 \dots x_n$ :

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

- Zudem gilt für alle  $i \in Z = \{0, 1, 2\}$ :

$$\begin{aligned} \delta(i, x_{n+1}) &\equiv_3 \begin{cases} i + 1, & x_{n+1} = a \\ i - 1, & x_{n+1} = b \end{cases} \\ &= i + \#_a(x_{n+1}) - \#_b(x_{n+1}) \end{aligned} \quad (*)$$

- Somit folgt

$$\begin{aligned} \hat{\delta}(0, x) &= \delta(\hat{\delta}(0, x'), x_{n+1}) \\ &\equiv_3 \hat{\delta}(0, x') + \#_a(x_{n+1}) - \#_b(x_{n+1}) \quad (*) \\ &\equiv_3 \#_a(x') - \#_b(x') + \#_a(x_{n+1}) - \#_b(x_{n+1}) \quad (IV) \\ &\equiv_3 \#_a(x) - \#_b(x) \end{aligned}$$

# Singletonsprachen sind regulär

## Vereinbarung

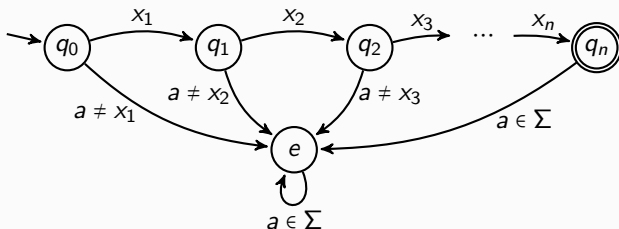
Für das Folgende sei  $\Sigma = \{a_1, \dots, a_m\}$  ein fest gewähltes Alphabet

## 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  $L(M) = \{x\}$ :



# REG ist unter Komplement abgeschlossen

## 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 wird das Komplement  $\bar{L}$  von  $L$  von dem DFA  $\bar{M} = (Z, \Sigma, \delta, q_0, Z \setminus E)$  akzeptiert

□

## Definition

Für eine Sprachklasse  $\mathcal{C}$  bezeichne  $\text{co-}\mathcal{C}$  die Klasse  $\{\bar{L} \mid L \in \mathcal{C}\}$  aller Komplemente von Sprachen in  $\mathcal{C}$

## Korollar

$\text{co-REG} = \text{REG}$

## 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

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

# 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
- Der Abschluss der Singletonsprachen unter  $\cap$ ,  $\cup$  und Komplement besteht aus allen endlichen und co-endlichen Sprachen



## 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  $M$  für das Produkt  $L(M_1)L(M_2)$  bereitet es Schwierigkeiten, den richtigen Zeitpunkt für das Ende der Simulation des DFA  $M_1$  und den Start der Simulation des DFA  $M_2$  zu finden

## Lösungsidee

Ein **nichtdeterministischer** endlicher Automat (NFA) kann den richtigen Zeitpunkt „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 oft auch mit  $2^Z$  bezeichnet

- Die von einem NFA  $N$  **akzeptierte (oder erkannte) Sprache** ist

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

- Eine Zustandsfolge  $q_0, \dots, q_n$  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

# Eigenschaften von NFAs

- 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

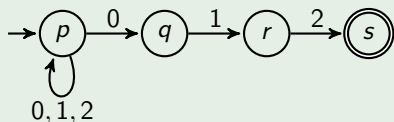
## Eigenschaften von NFAs

## 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:



- Ist  $w_1 = 012 \in L(N)$ ? **Ja** (akzeptierende Rechnung:  $p, q, r, s$ )  
Es gibt aber auch verwerfende Rechnungen bei Eingabe  $w_1$ :  $p, p, p, p$
- Ist  $w_2 = 021 \in L(N)$ ? **Nein**, da es keine akzeptierende Rechnung gibt
- Es gilt  $L(N) = \{x012 \mid x \in \Sigma^*\}$

## 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

$$\Delta(p, a) = \begin{cases} \Delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p, a) \cup \bigcup_{q \in Q_2} \Delta_2(q, a), & p \in E_1, \\ \Delta_2(p, a), & p \in Z_2 \end{cases}$$

und

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

# Ein NFA für das Produkt von regulären Sprachen

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

$$\Delta(p, a) = \begin{cases} \Delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p, a) \cup \bigcup_{q \in Q_2} \Delta_2(q, a), & p \in E_1, \\ \Delta_2(p, a), & p \in Z_2 \end{cases}$$

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 ist  $q_0, \dots, q_k, p_1, \dots, p_l$  eine akz. Rechnung von  $N(xy)$ , da

- $q_0 \in Q_1$  und  $p_l \in E_2$  ist, und
- im Fall  $l \geq 1$  wegen  $q_k \in E_1, p_0 \in Q_2$  und  $p_1 \in \Delta_2(p_0, y_1)$  zudem  $p_1 \in \Delta(q_k, y_1)$  und
- im Fall  $l = 0$  wegen  $q_k \in E_1$  und  $p_l \in Q_2 \cap E_2$  zudem  $q_k \in E$  ist



# Ein NFA für das Produkt von regulären Sprachen

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

$$\Delta(p, a) = \begin{cases} \Delta_1(p, a), & p \in Z_1 \setminus E_1, \\ \Delta_1(p, a) \cup \bigcup_{q \in Q_2} \Delta_2(q, a), & p \in E_1, \\ \Delta_2(p, a), & p \in Z_2 \end{cases}$$

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

## Beweis von $L(N) \subseteq L_1 L_2$ :

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  $i \leq n$

Wir zeigen, dass  $q_0, \dots, q_i$  eine akz. Rechnung von  $N_1(x_1 \cdots x_i)$  und

$q, q_{i+1}, \dots, q_n$  für ein  $q \in Q_2$  eine akz. Rechnung von  $N_2(x_{i+1} \cdots x_n)$  ist:

- 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. Zudem ist  $q_n \in E \cap Z_2 = E_2$
- Im Fall  $i = n$  ist  $q_n \in E \cap Z_1$ , was  $q_n \in E_1$  und  $Q_2 \cap E_2 \neq \emptyset$  impliziert

# Ein NFA für die Sternhülle einer regulären Sprache

## 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  $M = (Z, \Sigma, \delta, q_0, E)$  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

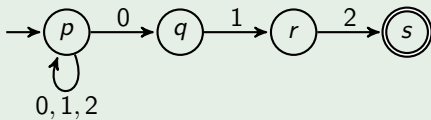
$$\delta(Q, a) = \bigcup_{q \in Q} \Delta(q, a)$$

aller Zustände, in die  $N$  gelangen kann, wenn  $N$  ausgehend von einem beliebigen Zustand  $q \in Q$  das Zeichen  $a$  liest

- $M$  wird auch als der zu  $N$  gehörige **Potenzmengenautomat** bezeichnet

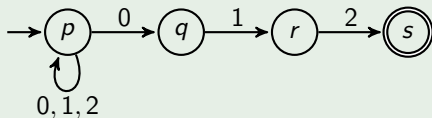
## Beispiel

- Betrachte den NFA  $N$



## Beispiel

- Betrachte den NFA  $N$

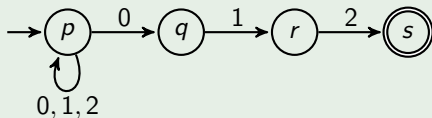


- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:



## Beispiel

- Betrachte den NFA  $N$

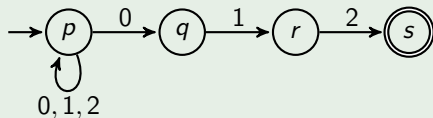


- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:

$\delta$	0	1	2
----------	---	---	---

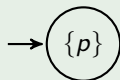
## Beispiel

- Betrachte den NFA  $N$



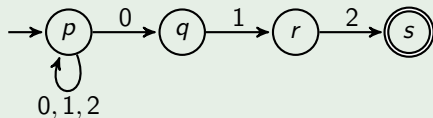
- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:

$\delta$	0	1	2
$\{p\}$			



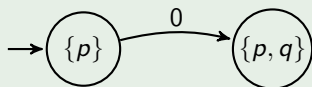
## Beispiel

- Betrachte den NFA  $N$



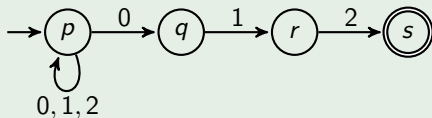
- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:

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



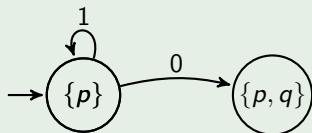
## Beispiel

- Betrachte den NFA  $N$



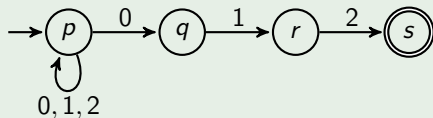
- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:

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



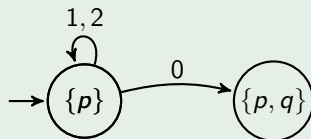
## Beispiel

- Betrachte den NFA  $N$



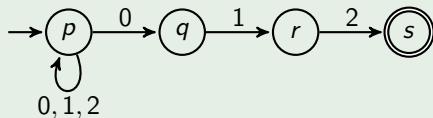
- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:

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



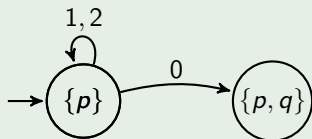
## Beispiel

- Betrachte den NFA  $N$



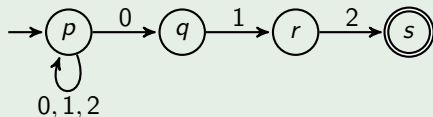
- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:

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



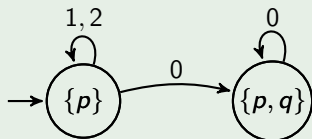
## Beispiel

- Betrachte den NFA  $N$



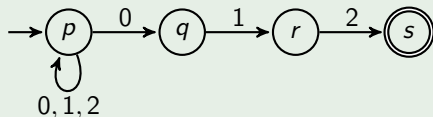
- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:

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



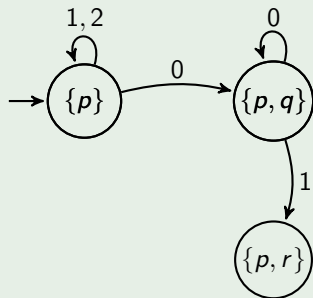
## Beispiel

- Betrachte den NFA  $N$



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

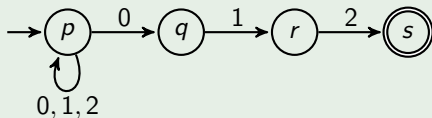
$\delta$	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	





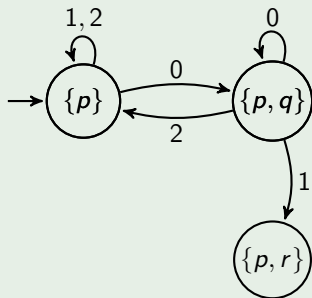
## Beispiel

- Betrachte den NFA  $N$



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

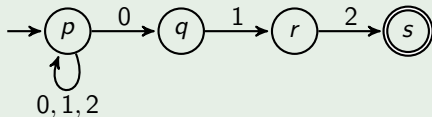
$\delta$	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$



## Simulation von NFAs durch DFAs

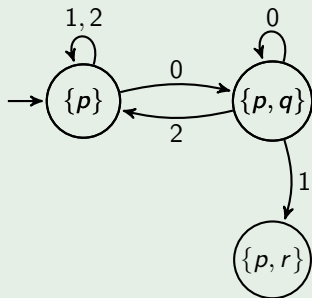
## Beispiel

- Betrachte den NFA  $N$



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

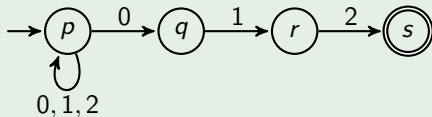
$\delta$	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$
$\{p, r\}$			



## Simulation von NFAs durch DFAs

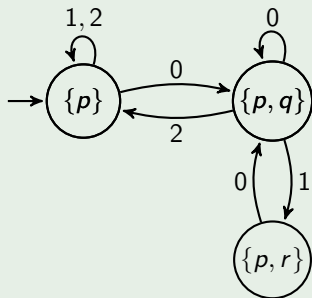
## Beispiel

- Betrachte den NFA  $N$



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

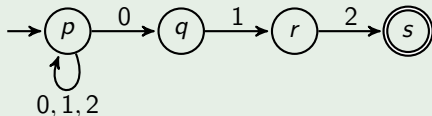
$\delta$	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$
$\{p, r\}$	$\{p, q\}$		



## Simulation von NFAs durch DFAs

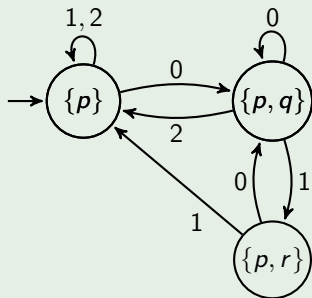
## Beispiel

- Betrachte den NFA  $N$



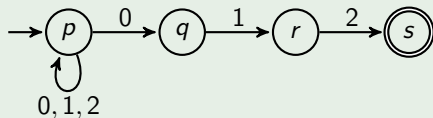
- Ausgehend von  $Q_0 = \{p\}$  liefert  $\delta$  dann die folgenden Werte:

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



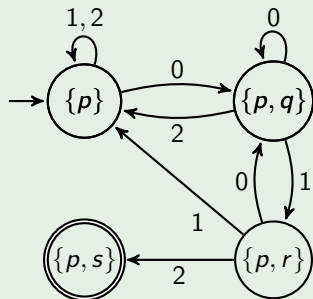
## Beispiel

- Betrachte den NFA  $N$



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

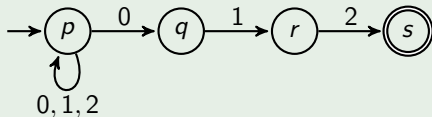
$\delta$	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$
$\{p, r\}$	$\{p, q\}$	$\{p\}$	$\{p, s\}$



## Simulation von NFAs durch DFAs

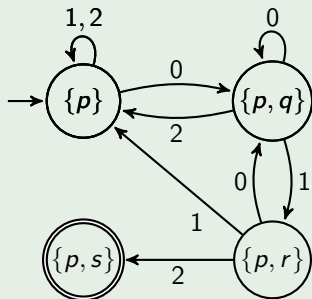
## Beispiel

- Betrachte den NFA  $N$



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

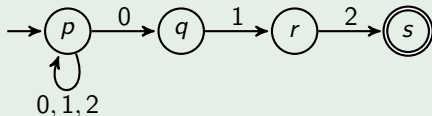
$\delta$	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$
$\{p, r\}$	$\{p, q\}$	$\{p\}$	$\{p, s\}$
$\{p, s\}$			



## Simulation von NFAs durch DFAs

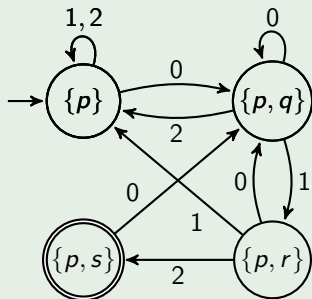
## Beispiel

- Betrachte den NFA  $N$



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

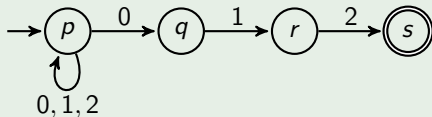
$\delta$	0	1	2
$\{p\}$	$\{p, q\}$	$\{p\}$	$\{p\}$
$\{p, q\}$	$\{p, q\}$	$\{p, r\}$	$\{p\}$
$\{p, r\}$	$\{p, q\}$	$\{p\}$	$\{p, s\}$
$\{p, s\}$	$\{p, q\}$		



## Simulation von NFAs durch DFAs

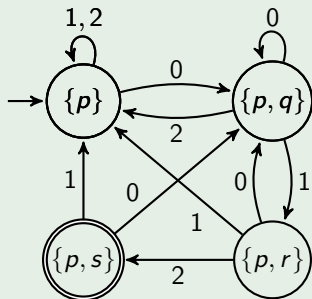
## Beispiel

- Betrachte den NFA  $N$



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

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

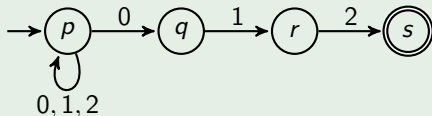




## Simulation von NFAs durch DFAs

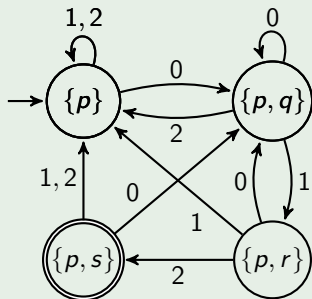
## Beispiel

- 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 (hierbei bezeichnet  $\|A\|$  die Mächtigkeit einer Menge  $A$ )

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

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  $M$  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
  - $x \in L(N) \iff N \text{ kann nach Lesen von } x \text{ einen Endzustand erreichen}$
  - $\iff \hat{\delta}(Q_0, x) \cap E \neq \emptyset$
  - $\iff \hat{\delta}(Q_0, x) \in E'$
  - $\iff x \in L(M)$

## 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

$n \rightsquigarrow n + 1$ : Sei  $x = x_1 \dots x_{n+1}$  gegeben. Nach IV enthält

$$Q_n = \hat{\delta}(Q_0, x_1 \dots x_n)$$

die Zustände, die  $N$  nach Lesen von  $x_1 \dots x_n$  erreichen kann.  
Wegen

$$\hat{\delta}(Q_0, x) = \delta(Q_n, x_{n+1}) = \bigcup_{q \in Q_n} \Delta(q, x_{n+1})$$

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

## Satz (Rabin und Scott)

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

## 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 endlichen 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

# Konstruktive Charakterisierung von REG

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  $\Gamma = \Sigma \cup \{\emptyset, \epsilon, |, *, (, )\}$

# Reguläre Ausdrücke

## Beispiel

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

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

## Beweis der Inklusion von rechts nach links.

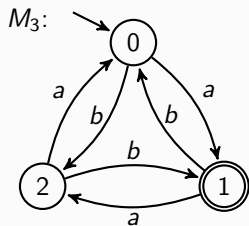
Klar, da

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



# Charakterisierung von REG durch reguläre Ausdrücke

Für die umgekehrte Inklusion betrachten wir zunächst den DFA  $M_3$ .



## Frage

Wie lässt sich die Sprache

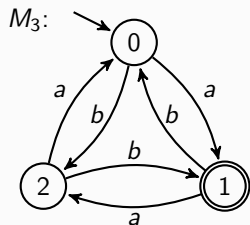
$$L(M_3) = \{x \in \{a, b\}^* \mid \#_a(x) - \#_b(x) \equiv_3 1\}$$

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.  $L_{p,q} = \{x \in \{a, b\}^* \mid \hat{\delta}(p, x) = q\}$ )
- 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  $L(M_3) = L_{0,1} = L_{0,0} L_{0,1}^{\neq 0}$ , wobei  $L_{0,0} = (L_{0,0}^{\neq 0})^*$  ist

## Antwort (Fortsetzung)



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

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

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

- 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

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

## Beweis der Inklusion von links nach rechts.

- 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

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

die wir auch einfach mit  $L_{p,q}^r$  bezeichnen

- 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)

$r = 0$ : 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, also durch reg. Ausdrücke  $\gamma_{p,q}^0$  beschreibbar

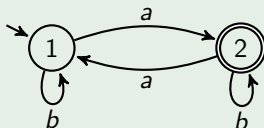
$r \rightsquigarrow r+1$ : Nach IV existieren reguläre Ausdrücke  $\gamma_{p,q}^r$  für die Sprachen  $L_{p,q}^r$ . Wegen

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

sind dann  $\gamma_{p,q}^{r+1} = \gamma_{p,q}^r \mid \gamma_{p,r+1}^r (\gamma_{r+1,r+1}^r)^* \gamma_{r+1,q}^r$  reguläre Ausdrücke für die Sprachen  $L_{p,q}^{r+1}$ . □

## Beispiel

- Betrachte den DFA  $M$



- Da  $M$  nur einen Endzustand  $q = 2$  und insgesamt  $m = 2$  Zustände besitzt, folgt

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

## 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

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

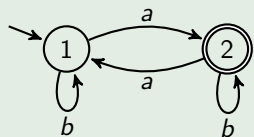
$$\gamma_{1,2}^1 = \gamma_{1,2}^0 | \gamma_{1,1}^0 (\gamma_{1,1}^0)^* \gamma_{1,2}^0$$

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

- Für die Berechnung von  $\gamma_{1,2}^2$  werden also nur die regulären Ausdrücke  $\gamma_{1,1}^0$ ,  $\gamma_{1,2}^0$ ,  $\gamma_{2,1}^0$ ,  $\gamma_{2,2}^0$ ,  $\gamma_{2,2}^1$  und  $\gamma_{1,2}^1$  benötigt



## Beispiel (Fortsetzung)

DFA  $M$ 

## Rekursionsformeln

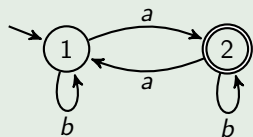
$$L_{p,p}^0 = \{c \in \Sigma \mid \delta(p, c) = p\} \cup \{\varepsilon\}$$

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

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

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\gamma_{1,1}^0$	$\gamma_{1,2}^0$	$\gamma_{2,1}^0$	$\gamma_{2,2}^0$
1	-	$\gamma_{1,2}^1$	-	$\gamma_{2,2}^1$
2	-	$\gamma_{1,2}^2$	-	-

## Beispiel (Fortsetzung)

DFA  $M$ 

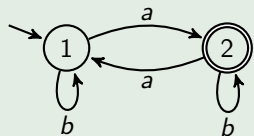
Rekursionsformel

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

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

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon b$	$\gamma_{1,2}^0$	$\gamma_{2,1}^0$	$\gamma_{2,2}^0$
1	-	$\gamma_{1,2}^1$	-	$\gamma_{2,2}^1$
2	-	$\gamma_{1,2}^2$	-	-

## Beispiel (Fortsetzung)

DFA  $M$ 

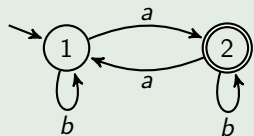
Rekursionsformel

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

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

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon b$	$a$	$\gamma_{2,1}^0$	$\gamma_{2,2}^0$
1	-	$\gamma_{1,2}^1$	-	$\gamma_{2,2}^1$
2	-	$\gamma_{1,2}^2$	-	-

## Beispiel (Fortsetzung)

DFA  $M$ 

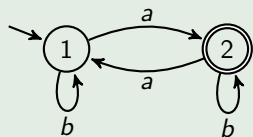
Rekursionsformel

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

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

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\epsilon b$	$a$	$a$	$\gamma_{2,2}^0$
1	-	$\gamma_{1,2}^1$	-	$\gamma_{2,2}^1$
2	-	$\gamma_{1,2}^2$	-	-

## Beispiel (Fortsetzung)

DFA  $M$ 

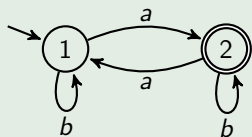
Rekursionsformel

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

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

r	p, q			
	1, 1	1, 2	2, 1	2, 2
0	$\varepsilon b$	$a$	$a$	$\varepsilon b$
1	-	$\gamma_{1,2}^1$	-	$\gamma_{2,2}^1$
2	-	$\gamma_{1,2}^2$	-	-

## Beispiel (Fortsetzung)

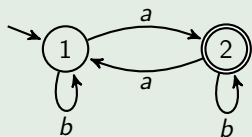
DFA  $M$ 

Rekursionsformel

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

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

## Beispiel (Fortsetzung)

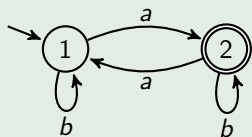
DFA  $M$ 

Rekursionsformel

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

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	-	$\gamma_{1,2}^2$	-	-

## Beispiel (Fortsetzung)

DFA  $M$ 

Rekursionsformel

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

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	-	$b^* a (b   ab^* a)^*$	-	-



## Korollar

Für jede Sprache  $L$  sind folgende Aussagen äquivalent:

- $L$  ist regulär (d.h. 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 Vereinigung, Schnitt, 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

# Relationalstrukturen

## Definition

- Sei  $A$  eine nichtleere Menge,  $R$  ist eine  **$k$ -stellige Relation auf  $A$** , wenn  $R \subseteq A^k = \underbrace{A \times \dots \times A}_{k\text{-mal}} = \{(a_1, \dots, a_k) \mid a_i \in A \text{ für } i = 1, \dots, k\}$  ist
- Für  $i = 1, \dots, n$  sei  $R_i$  eine  $k_i$ -stellige Relation auf  $A$ . Dann heißt  $(A; R_1, \dots, R_n)$  **Relationalstruktur**
- Die Menge  $A$  heißt der **Individuenbereich**, die **Trägermenge** oder die **Grundmenge** der Relationalstruktur

## Bemerkung

- Wir werden hier hauptsächlich den Fall  $n = 1$ ,  $k_1 = 2$ , also  $(A, R)$  mit  $R \subseteq A \times A$  betrachten
- Man nennt dann  $R$  eine **(binäre) Relation** auf  $A$
- Oft wird für  $(a, b) \in R$  auch die **Infix-Schreibweise**  $aRb$  benutzt

## Beispiel

- $(F, M)$  mit  $F = \{f \mid f \text{ ist Fluss in Europa}\}$  und  
 $M = \{(f, g) \in F \times F \mid f \text{ mündet in } g\}$
- $(U, B)$  mit  $U = \{x \mid x \text{ ist Berliner}\}$  und  
 $B = \{(x, y) \in U \times U \mid x \text{ ist Bruder von } y\}$
- $(\mathcal{P}(M), \subseteq)$ , wobei  $M$  eine beliebige Menge und  $\subseteq$  die Inklusionsrelation auf den Teilmengen von  $M$  ist
- $(A, Id_A)$  mit  $Id_A = \{(x, x) \mid x \in A\}$  (die **Identität auf  $A$** )
- $(\mathbb{R}, \leq)$
- $(\mathbb{Z}, |)$ , wobei  $|$  die "teilt"-Relation bezeichnet (d.h.  $a|b$ , falls ein  $c \in \mathbb{Z}$  mit  $b = ac$  existiert)

# Mengentheoretische Operationen auf Relationen

- Da Relationen Mengen sind, können wir den **Schnitt**, die **Vereinigung**, die **Differenz** und das **Komplement** von Relationen bilden:

$$R \cap S = \{(x, y) \in A \times A \mid xRy \wedge xSy\}$$

$$R \cup S = \{(x, y) \in A \times A \mid xRy \vee xSy\}$$

$$R - S = \{(x, y) \in A \times A \mid xRy \wedge \neg xSy\}$$

$$\bar{R} = (A \times A) - R$$

- Sei  $\mathcal{M} \subseteq \mathcal{P}(A \times A)$  eine beliebige Menge von Relationen auf  $A$ . Dann sind der **Schnitt über  $\mathcal{M}$**  und die **Vereinigung über  $\mathcal{M}$**  folgende Relationen:

$$\bigcap \mathcal{M} = \bigcap_{R \in \mathcal{M}} R = \{(x, y) \mid \forall R \in \mathcal{M} : xRy\}$$

$$\bigcup \mathcal{M} = \bigcup_{R \in \mathcal{M}} R = \{(x, y) \mid \exists R \in \mathcal{M} : xRy\}$$

# Weitere Operationen auf Relationen

## Definition

- Die **transponierte (konverse) Relation** zu  $R$  ist

$$R^T = \{(y, x) \mid xRy\}$$

- $R^T$  wird oft auch mit  $R^{-1}$  bezeichnet
- Zum Beispiel ist  $(\mathbb{R}, \leq^T) = (\mathbb{R}, \geq)$
- Das **Produkt** (oder die **Komposition**) zweier Relationen  $R$  und  $S$  ist

$$R \circ S = \{(x, z) \in A \times A \mid \exists y \in A : xRy \wedge ySz\}$$

## Beispiel

Ist  $B$  die Relation "ist Bruder von",  $V$  "ist Vater von",  $M$  "ist Mutter von" und  $E = V \cup M$  "ist Elternteil von", so ist  $B \circ E$  die Onkel-Relation ◀

## Notation

- Für  $R \circ S$  wird auch  $R ; S$ ,  $R \cdot S$  oder einfach  $RS$  geschrieben.
- Für  $\underbrace{R \circ \dots \circ R}_{n\text{-mal}}$  schreiben wir auch  $R^n$ . Dabei ist  $R^0 = Id$

## Vorsicht!

Das Relationenprodukt  $R^n$  darf nicht mit dem kartesischen Produkt

$$\underbrace{R \times \dots \times R}_{n\text{-mal}}$$

verwechselt werden

## Vereinbarung

Wir vereinbaren, dass  $R^n$  das  $n$ -fache Relationenprodukt bezeichnen soll, falls  $R$  eine Relation ist

## Definition

Sei  $R$  eine Relation auf  $A$ . Dann heißt  $R$

**reflexiv**, falls  $\forall x \in A : xRx$  (also  $Id_A \subseteq R$ )

**irreflexiv**, falls  $\forall x \in A : \neg xRx$  (also  $Id_A \subseteq \bar{R}$ )

**symmetrisch**, falls  $\forall x, y \in A : xRy \Rightarrow yRx$  (also  $R \subseteq R^T$ )

**asymmetrisch**, falls  $\forall x, y \in A : xRy \Rightarrow \neg yRx$  (also  $R \subseteq \overline{R^T}$ )

**antisymmetrisch**, falls  $\forall x, y \in A : xRy \wedge yRx \Rightarrow x = y$  (also  $R \cap R^T \subseteq Id$ )

**konnex**, falls  $\forall x, y \in A : xRy \vee yRx$  (also  $A \times A \subseteq R \cup R^T$ )

**semikonnex**, falls  $\forall x, y \in A : x \neq y \Rightarrow xRy \vee yRx$  (also  $\overline{Id} \subseteq R \cup R^T$ )

**transitiv**, falls  $\forall x, y, z \in A : xRy \wedge yRz \Rightarrow xRz$  (also  $R^2 \subseteq R$ )

gilt

## Äquivalenz- und Ordnungsrelationen

	refl.	sym.	trans.	antisym.	asym.	konnex	semikon.
Äquivalenzrelation	✓	✓	✓				
(Halb-)Ordnung	✓		✓	✓			
Striktordnung			✓			✓	
lineare Ordnung			✓	✓			✓
lin. Striktord.			✓			✓	✓
Quasiordnung	✓		✓				

### Bemerkung

In der Tabelle sind nur die definierenden Eigenschaften durch ein "✓" gekennzeichnet. Das schließt nicht aus, dass noch weitere Eigenschaften vorliegen



## Beispiel

- Die Relation "ist Schwester von" ist zwar in einer reinen Damengesellschaft symmetrisch, i.a. jedoch weder symmetrisch noch asymmetrisch noch antisymmetrisch.
- Die Relation "ist Geschwister von" ist zwar symmetrisch, aber weder reflexiv noch transitiv und somit keine Äquivalenzrelation.
- $(\mathbb{R}, <)$  ist irreflexiv, asymmetrisch, transitiv und semikonnex und somit eine lineare Striktordnung.
- $(\mathbb{R}, \leq)$  und  $(\mathcal{P}(M), \subseteq)$  sind reflexiv, antisymmetrisch und transitiv und somit Ordnungen.
- $(\mathbb{R}, \leq)$  ist auch konnex und somit eine lineare Ordnung.
- $(\mathcal{P}(M), \subseteq)$  ist zwar im Fall  $\|M\| \leq 1$  konnex, aber im Fall  $\|M\| \geq 2$  weder semikonnex noch konnex.

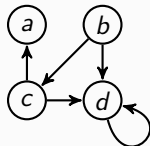


# Darstellung von endlichen Relationen

## Graphische Darstellung

$$A = \{a, b, c, d\}$$

$$R = \{(b, c), (b, d), (c, a), (c, d), (d, d)\}$$



- Eine Relation  $R$  auf einer (endlichen) Menge  $A$  kann durch einen **gerichteten Graphen** (kurz **Digraphen**)  $G = (A, R)$  mit **Knotenmenge**  $A$  und **Kantenmenge**  $R$  veranschaulicht werden
- Hierzu stellen wir jedes Element  $x \in A$  als einen Knoten dar und verbinden jedes Knotenpaar  $(x, y) \in R$  durch eine gerichtete Kante (Pfeil)
- Zwei durch eine Kante verbundene Knoten heißen **adjazent** oder **benachbart**

# Darstellung von endlichen Relationen

## Definition

Sei  $R$  eine binäre Relation auf  $A$

- Die Menge der Nachfolger bzw. Vorgänger von  $x$  ist

$$R[x] = \{y \in A \mid xRy\} \text{ bzw. } R^{-1}[x] = \{y \in A \mid yRx\}$$

- Der Ausgangsgrad eines Knotens  $x$  ist  $\deg^+(x) = \|R[x]\|$
- Der Eingangsgrad von  $x$  ist  $\deg^-(x) = \|R^{-1}[x]\|$
- Ist  $R$  symmetrisch, so können wir die Pfeilspitzen auch weglassen
- In diesem Fall heißt  $\deg(x) = \deg^-(x) = \deg^+(x)$  der Grad von  $x$  und  $R[x] = R^{-1}[x]$  die Nachbarschaft von  $x$  in  $G$
- $G$  ist schleifenfrei, falls  $R$  irreflexiv ist
- Ist  $R$  irreflexiv und symmetrisch, so nennen wir  $G = (A, R)$  einen (ungerichteten) Graphen
- Eine irreflexive und symmetrische Relation  $R$  wird meist als Menge der ungeordneten Paare  $E = \{\{a, b\} \mid aRb\}$  notiert

# Darstellung von endlichen Relationen

## Matrixdarstellung (Adjazenzmatrix)

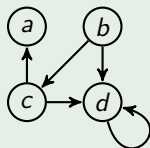
Eine Relation  $R$  auf  $A = \{a_1, \dots, a_n\}$  lässt sich auch durch die boolesche  $(n \times n)$ -Matrix  $M_R = (m_{ij})$  darstellen mit

$$m_{ij} = \begin{cases} 1, & a_i R a_j \\ 0, & \text{sonst} \end{cases}$$

## Beispiel

Die Relation  $R = \{(b, c), (b, d), (c, a), (c, d), (d, d)\}$  auf  $A = \{a, b, c, d\}$  hat beispielsweise die Matrixdarstellung

$$M_R = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



# Darstellung von endlichen Relationen

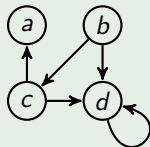
## Listendarstellung (Adjazenzlisten)

$R$  lässt sich auch durch eine Tabelle darstellen, die jedem Element  $x \in A$  seine Nachfolger in Form einer Liste zuordnet

### Beispiel

Die Relation  $R = \{(b, c), (b, d), (c, a), (c, d), (d, d)\}$  auf  $A = \{a, b, c, d\}$  lässt sich beispielsweise durch folgende Adjazenzlisten darstellen:

$x$ :	$R[x]$
$a$ :	-
$b$ :	$c, d$
$c$ :	$a, d$
$d$ :	$d$



## Berechnung von $R \circ S$

- Sind  $M_R = (r_{ij})$  und  $M_S = (s_{ij})$  boolesche  $(n \times n)$ -Matrizen für  $R$  und  $S$ , so erhalten wir für  $T = R \circ S$  die Matrix  $M_T = (t_{ij})$  mit

$$t_{ij} = \bigvee_{k=1, \dots, n} (r_{ik} \wedge s_{kj})$$

- Die Nachfolgermenge  $T[x]$  von  $x$  bzgl. der Relation  $T = R \circ S$  berechnet sich zu

$$T[x] = \bigcup_{y \in R[x]} S[y]$$

## Beispiel

Betrachte die Relationen  $R = \{(a, a), (a, c), (c, b), (c, d)\}$  und  $S = \{(a, b), (d, a), (d, c)\}$  auf der Menge  $A = \{a, b, c, d\}$ .

Relation	$R$	$S$	$R \circ S$	$S \circ R$
Digraph				
Adjazenzmatrix	$\begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{matrix}$	$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$
Adjazenzlisten	$\begin{matrix} a: a, c \\ b: - \\ c: b, d \\ d: - \end{matrix}$	$\begin{matrix} a: b \\ b: - \\ c: - \\ d: a, c \end{matrix}$	$\begin{matrix} a: b \\ b: - \\ c: a, c \\ d: - \end{matrix}$	$\begin{matrix} a: - \\ b: - \\ c: - \\ d: a, b, c, d \end{matrix}$

## Frage

Welche Paare muss man zu einer Relation  $R$  mindestens hinzufügen, damit  $R$  transitiv wird?

## Antwort

- Es ist klar, dass der Schnitt von transitiven Relationen wieder transitiv ist
- Die **transitive Hülle** von  $R$  ist

$$R^+ = \bigcap \{S \subseteq A \times A \mid S \text{ ist transitiv und } R \subseteq S\}$$

- $R^+$  ist also eine transitive Relation, die  $R$  enthält
- Da  $R^+$  zudem in jeder Relation mit diesen Eigenschaften enthalten ist, gibt es keine transitive Relation mit weniger Paaren, die  $R$  enthält
- Da auch die Reflexivität und die Symmetrie bei der Schnittbildung erhalten bleiben, lassen sich nach demselben Muster weitere Hüllenoperatoren definieren



## Definition

Sei  $R$  eine Relation auf  $A$

- Die **reflexive Hülle** von  $R$  ist

$$h_{\text{refl}}(R) = \bigcap \{S \subseteq A \times A \mid S \text{ ist reflexiv und } R \subseteq S\}$$

- Die **symmetrische Hülle** von  $R$  ist

$$h_{\text{sym}}(R) = \bigcap \{S \subseteq A \times A \mid S \text{ ist symmetrisch und } R \subseteq S\}$$

- Die **reflexiv-transitive Hülle** von  $R$  ist

$$R^* = \bigcap \{S \subseteq A \times A \mid S \text{ ist reflexiv, transitiv und } R \subseteq S\}$$

- Die **Äquivalenzhülle** von  $R$  ist

$$h_{\text{äq}}(R) = \bigcap \{E \subseteq A \times A \mid E \text{ ist eine Äquivalenzrelation mit } R \subseteq E\}$$

## Satz

$$h_{\text{refl}}(R) = R \cup \text{Id}_A, \quad h_{\text{sym}}(R) = R \cup R^T, \quad R^+ = \bigcup_{n \geq 1} R^n, \quad R^* = \bigcup_{n \geq 0} R^n$$

## Beweis

Siehe Übungen. □

Bemerkung. Sei  $G = (V, E)$  ein Digraph.

- Ein Paar  $(a, b)$  ist genau dann in der reflexiv-transitiven Hülle  $E^*$  von  $E$  enthalten, wenn es ein  $n \geq 0$  gibt mit  $aE^n b$
- Dies bedeutet, dass es Elemente  $x_0, \dots, x_n \in V$  gibt mit
$$x_0 = a, x_n = b \text{ und } (x_{i-1}, x_i) \in E \text{ f\"ur } i = 1, \dots, n$$
- $x_0, \dots, x_n$  heit **Weg** der Lnge  $n$  von  $a$  nach  $b$  in  $G$
- $G$  heit **zusammenhngend**, wenn es in  $G$  fr je zwei Knoten  $a$  und  $b$  einen Weg von  $a$  nach  $b$  oder einen Weg von  $b$  nach  $a$  gibt
- $G$  heit **stark zusammenhngend**, wenn es in  $G$  von jedem Knoten  $a$  einen Weg zu jedem Knoten  $b$  gibt

## Definition

$(A, R)$  heißt **Äquivalenzrelation**, wenn  $R$  eine reflexive, symmetrische und transitive Relation auf  $A$  ist

## Beispiel

- Auf der Menge aller Geraden im  $\mathbb{R}^2$  die Parallelität
- Auf der Menge aller Menschen "im gleichen Jahr geboren wie"
- Auf  $\mathbb{Z}$  die Relation "gleicher Rest bei Division durch  $m$ ".



## Definition

- Ist  $E$  eine Äquivalenzrelation, so nennt man die Nachbarschaft  $E[x]$  die **von  $x$  repräsentierte Äquivalenzklasse** und bezeichnet sie auch mit  $[x]_E$  (oder einfach mit  $[x]$ , falls  $E$  aus dem Kontext ersichtlich ist):

$$[x]_E = [x] = E[x] = \{y \mid xEy\}$$

- Eine Menge  $S \subseteq A$  heißt **Repräsentantensystem**, falls sie genau ein Element aus jeder Äquivalenzklasse enthält
- Die Menge aller Äquivalenzklassen von  $E$  wird **Quotienten- oder Faktormenge** von  $A$  bzgl.  $E$  genannt und mit  $A/E$  bezeichnet:

$$A/E = \{[x]_E \mid x \in A\}$$

- Die Anzahl  $\|A/E\|$  der Äquivalenzklassen von  $E$  wird auch als der **Index von  $E$**  (kurz:  $\text{index}(E)$ ) bezeichnet

## Beispiel

Für die weiter oben betrachteten Äquivalenzrelationen erhalten wir folgende Klasseneinteilungen:

- Für die Parallelität auf der Menge aller Geraden im  $\mathbb{R}^2$ : alle Geraden mit derselben Richtung (oder Steigung) bilden jeweils eine Äquivalenzklasse
- Ein Repräsentantensystem wird beispielsweise durch die Menge aller Ursprungsgeraden gebildet
- Für die Relation "im gleichen Jahr geboren wie" auf der Menge aller Menschen: jeder Jahrgang bildet eine Äquivalenzklasse
- Für die Relation "gleicher Rest bei Division durch  $m$ " auf  $\mathbb{Z}$ : jede der  $m$  Restklassen  $[0], [1], \dots, [m-1]$  mit

$$[r] = \{a \in \mathbb{Z} \mid a \bmod m = r\}$$

bildet eine Äquivalenzklasse

- Repräsentantensystem:  $\{0, 1, \dots, m-1\}$ .

## Bemerkungen

- Die kleinste Äquivalenzrelation auf  $A$  ist die Identität  $Id_A$ , die größte ist die Allrelation  $A \times A$
- Die Äquivalenzklassen der Identität enthalten jeweils nur ein Element, d.h.  $[x]_{Id_A} = \{x\}$  für alle  $x \in A$
- Die Allrelation erzeugt dagegen nur eine Äquivalenzklasse, nämlich  $[x]_{A \times A} = A$  für alle  $x \in A$
- Die Identität  $Id_A$  hat nur ein Repräsentantensystem, nämlich  $A$
- Dagegen kann jede Singletonmenge  $\{x\}$  mit  $x \in A$  als Repräsentantensystem für die Allrelation  $A \times A$  fungieren

Wie wir sehen werden, bilden die Äquivalenzklassen eine Zerlegung von  $A$

## Definition

Eine Familie  $\{B_i \mid i \in I\}$  von nichtleeren Teilmengen  $B_i \subseteq A$  heißt **Partition** (oder **Zerlegung**) der Menge  $A$ , falls gilt:

- die Mengen  $B_i$  **überdecken**  $A$ , d.h.  $A = \bigcup_{i \in I} B_i$  und
- die Mengen  $B_i$  sind **paarweise disjunkt**, d.h. für je zwei verschiedene Mengen  $B_i \neq B_j$  gilt  $B_i \cap B_j = \emptyset$



## Bemerkungen

- Für zwei Äquivalenzrelationen  $E \subseteq E'$  sind auch die Äquivalenzklassen  $[x]_E$  von  $E$  in den Klassen  $[x]_{E'}$  von  $E'$  enthalten
- Folglich ist jede Äquivalenzklasse von  $E'$  die Vereinigung von (evtl. mehreren) Äquivalenzklassen von  $E$
- Im Fall  $E \subseteq E'$  sagt man auch,  $E$  bewirkt eine **feinere** Zerlegung von  $A$  als  $E'$
- Demnach ist die Identität die **feinste** und die Allrelation die **größte** Äquivalenzrelation

## Satz

Sei  $E$  eine Relation auf  $A$ . Dann sind folgende Aussagen äquivalent:

- ①  $E$  ist eine Äquivalenzrelation auf  $A$
- ② es gibt eine Partition  $\{B_i \mid i \in I\}$  von  $A$  mit  $xEy \Leftrightarrow \exists i \in I : x, y \in B_i$

## Beweis.

① impliziert ②: Sei  $E$  eine Äquivalenzrelation auf  $A$ . Dann bildet  $\{E[x] \mid x \in A\}$  eine Partition von  $A$  mit der gewünschten Eigenschaft:

- Da  $E$  reflexiv ist, gilt  $xEx$  und somit  $x \in E[x]$ , d.h.  $A = \bigcup_{x \in A} E[x]$
- Ist  $E[x] \cap E[y] \neq \emptyset$  und  $u \in E[x] \cap E[y]$ , so folgt  $E[x] = E[y]$ :

$$z \in E[x] \Leftrightarrow xEz \stackrel{xEu}{\Leftrightarrow} uEz \stackrel{yEu}{\Leftrightarrow} yEz \Leftrightarrow z \in E[y]$$

- Zudem gilt

$$\exists z \in A : x, y \in E[z] \Leftrightarrow \exists z : z \in E[x] \cap E[y] \Leftrightarrow E[x] = E[y] \stackrel{y \in E[y]}{\Leftrightarrow} xEy$$

## Satz

Sei  $E$  eine Relation auf  $A$ . Dann sind folgende Aussagen äquivalent:

- 1  $E$  ist eine Äquivalenzrelation auf  $A$
- 2 es gibt eine Partition  $\{B_i \mid i \in I\}$  von  $A$  mit  $xEy \Leftrightarrow \exists i \in I : x, y \in B_i$

## Beweis.

2 **impliziert** 1: Existiert umgekehrt eine Partition  $\{B_i \mid i \in I\}$  von  $A$  mit  $xEy \Leftrightarrow \exists i \in I : x, y \in B_i$ , so ist  $E$

- reflexiv, da zu jedem  $x \in A$  eine Menge  $B_i$  mit  $x \in B_i$  existiert,
- symmetrisch, da aus  $x, y \in B_i$  auch  $y, x \in B_i$  folgt, und
- transitiv, da aus  $x, y \in B_i$  und  $y, z \in B_j$  wegen  $y \in B_i \cap B_j$  die Gleichheit  $B_i = B_j$  und somit  $x, z \in B_i$  folgt. □

# Ordnungsrelationen

## Definition

$(A, R)$  heißt **Ordnung** (auch **Halbordnung** oder **partielle Ordnung**), wenn  $R$  eine reflexive, antisymmetrische und transitive Relation auf  $A$  ist

## Beispiel

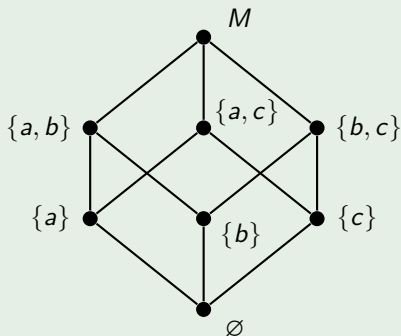
- $(\mathcal{P}(M), \subseteq)$ ,  $(\mathbb{Z}, \leq)$ ,  $(\mathbb{R}, \leq)$ ,  $(\mathbb{N}, |)$ , sind Ordnungen.  $(\mathbb{Z}, |)$  ist keine Ordnung, aber eine Quasiordnung.
- Ist  $R$  eine Relation auf  $A$  und  $B \subseteq A$ , so ist  $R_B = R \cap (B \times B)$  die **Einschränkung** von  $R$  auf  $B$
- Einschränkungen von (linearen) Ordnungen sind ebenfalls (lineare) Ordnungen
- Beispielsweise ist  $(\mathbb{Q}, \leq)$  die Einschränkung von  $(\mathbb{R}, \leq)$  auf  $\mathbb{Q}$  und  $(\mathbb{N}, |)$  die Einschränkung von  $(\mathbb{Z}, |)$  auf  $\mathbb{N}$ . ◀

- Sei  $\leq$  eine Ordnung auf  $A$  und sei  $<$  die Relation  $\leq \setminus Id_A$ , d.h.  
$$x < y \Leftrightarrow x \leq y \wedge x \neq y$$
- Ein Element  $x \in A$  heißt **unterer Nachbar** von  $y$  (kurz:  $x \triangleleft y$ ), falls  $x < y$  gilt und kein  $z \in A$  existiert mit  $x < z < y$
- $\triangleleft$  ist also die Relation  $< \setminus <^2$
- Um die Ordnung  $(A, \leq)$  in einem **Hasse-Diagramm** darzustellen, wird nur der Digraph der Relation  $(A, \triangleleft)$  gezeichnet
- Weiterhin wird im Fall  $x \triangleleft y$  der Knoten  $y$  oberhalb des Knotens  $x$  gezeichnet, so dass auf die Pfeilspitzen verzichtet werden kann

# Das Hasse-Diagramm für $(\mathcal{P}(M); \subseteq)$

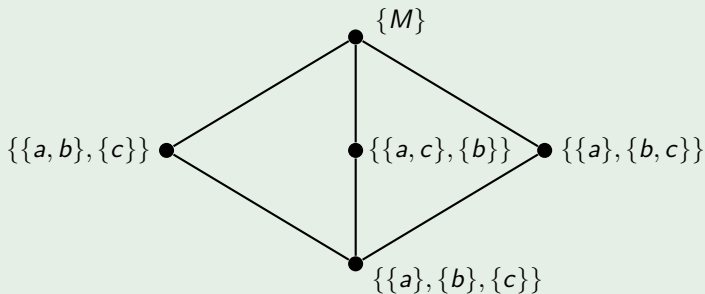
## Beispiel

Die Inklusion  $\subseteq$  auf  $\mathcal{P}(M)$  mit  $M = \{a, b, c\}$  lässt sich durch folgendes Hasse-Diagramm darstellen:



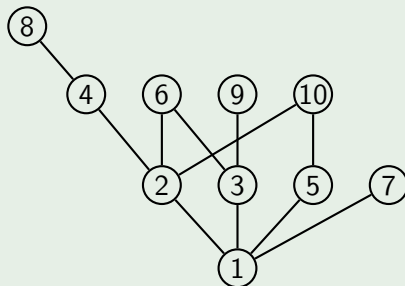
## Beispiel

Die "feiner als" Relation auf der Menge aller Partitionen von  $M = \{a, b, c\}$  ist durch folgendes Hasse-Diagramm darstellbar:



## Beispiel

Die Einschränkung der "teilt"-Relation auf die Menge  $\{1, 2, \dots, 10\}$  ist durch folgendes Hasse-Diagramm darstellbar:





# Maximale, minimale, größte und kleinste Elemente

## Definition

Sei  $\leq$  eine Ordnung auf  $A$  und sei  $b$  ein Element in einer Teilmenge  $B \subseteq A$

- $b$  heißt **kleinstes Element** oder **Minimum** von  $B$  ( $b = \min B$ ), falls gilt:

$$\forall b' \in B : b \leq b'$$

- $b$  heißt **größtes Element** oder **Maximum** von  $B$  ( $b = \max B$ ), falls gilt:

$$\forall b' \in B : b' \leq b$$

- $b$  heißt **minimal** in  $B$ , falls es in  $B$  kein kleineres Element gibt:

$$\forall b' \in B : b' \leq b \Rightarrow b' = b$$

- $b$  heißt **maximal** in  $B$ , falls es in  $B$  kein größeres Element gibt:

$$\forall b' \in B : b \leq b' \Rightarrow b = b'$$

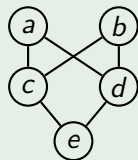
## Bemerkung

Wegen der Antisymmetrie kann es in  $B$  höchstens ein kleinstes und höchstens ein größtes Element geben

# Maximale, minimale, größte und kleinste Elemente

## Beispiel

Betrachte folgende Ordnung



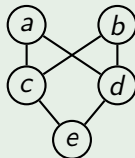
$B$	minimal in $B$	maximal in $B$	$\min B$	$\max B$
$\{a, b\}$	$a, b$	$a, b$	-	-
$\{c, d\}$	$c, d$	$c, d$	-	-
$\{a, b, c\}$	$c$	$a, b$	$c$	-
$\{a, b, c, e\}$	$e$	$a, b$	$e$	-
$\{a, c, d, e\}$	$e$	$a$	$e$	$a$

## Definition

Sei  $\leq$  eine Ordnung auf  $A$  und sei  $B \subseteq A$ . Dann heißt

- ein Element  $u \in A$  mit  $u \leq b$  für alle  $b \in B$  **untere Schranke von  $B$**
- ein Element  $o \in A$  mit  $b \leq o$  für alle  $b \in B$  **obere Schranke von  $B$**
- $B$  **nach oben beschränkt**, wenn  $B$  eine obere Schranke hat
- $B$  **nach unten beschränkt**, wenn  $B$  eine untere Schranke hat
- $B$  **beschränkt**, wenn  $B$  nach oben und nach unten beschränkt ist

## Beispiel (Fortsetzung)



$B$	minimal	maximal	min	max	untere Schranken	obere Schranken
$\{a, b\}$	$a, b$	$a, b$	-	-	$c, d, e$	-
$\{c, d\}$	$c, d$	$c, d$	-	-	$e$	$a, b$
$\{a, b, c\}$	$c$	$a, b$	$c$	-	$c, e$	-
$\{a, b, c, e\}$	$e$	$a, b$	$e$	-	$e$	-
$\{a, c, d, e\}$	$e$	$a$	$e$	$a$	$e$	$a$

## Definition

Sei  $\leq$  eine Ordnung auf  $A$  und sei  $B \subseteq A$

- Besitzt  $B$  eine größte untere Schranke  $i$ , d.h. besitzt die Menge  $U$  aller unteren Schranken von  $B$  ein größtes Element  $i$ , so heißt  $i$  das **Infimum von  $B$**  ( $i = \inf B$ ):

$$(\forall b \in B : b \geq i) \wedge [\forall u \in A : (\forall b \in B : b \geq u) \Rightarrow u \leq i]$$

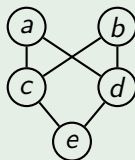
- Besitzt  $B$  eine kleinste obere Schranke  $s$ , d.h. besitzt die Menge  $O$  aller oberen Schranken von  $B$  ein kleinstes Element  $s$ , so heißt  $s$  das **Supremum von  $B$**  ( $s = \sup B$ ):

$$(\forall b \in B : b \leq s) \wedge [\forall o \in A : (\forall b \in B : b \leq o) \Rightarrow s \leq o]$$

## Bemerkung

$B$  kann nicht mehr als ein Supremum und ein Infimum haben

## Beispiel (Schluss)



$B$	minimal	maximal	min	max	untere obere Schranken		inf	sup
$\{a, b\}$	$a, b$	$a, b$	-	-	$c, d, e$	-	-	-
$\{c, d\}$	$c, d$	$c, d$	-	-	$e$	$a, b$	$e$	-
$\{a, b, c\}$	$c$	$a, b$	$c$	-	$c, e$	-	$c$	-
$\{a, b, c, e\}$	$e$	$a, b$	$e$	-	$e$	-	$e$	-
$\{a, c, d, e\}$	$e$	$a$	$e$	$a$	$e$	$a$	$e$	$a$

## Bemerkung

- In einer endlichen linearen Ordnung  $(A; \leq)$  besitzt jede nichtleere Teilmenge  $B \subseteq A$  ein Maximum und ein Minimum sowie ein Supremum und ein Infimum, wobei  $\sup B = \max B$  und  $\inf B = \min B$
- Zudem ist  $\sup \emptyset = \min A$  und  $\inf \emptyset = \max A$
- Dagegen müssen in einer unendlichen linearen Ordnung nicht einmal beschränkte Teilmengen ein Supremum oder Infimum besitzen
- So hat in der linear geordneten Menge  $(\mathbb{Q}, \leq)$  die Teilmenge

$$B = \{x \in \mathbb{Q} \mid x^2 \leq 2\} = \{x \in \mathbb{Q} \mid x^2 < 2\}$$

weder ein Supremum noch ein Infimum

- Dagegen hat in  $(\mathbb{R}, \leq)$  jede beschränkte Teilmenge  $B$  ein Supremum und ein Infimum (aber möglicherweise kein Maximum oder Minimum)

Definition. Sei  $R$  eine binäre Relation auf einer Menge  $M$ .

- $R$  heißt **rechtseindeutig**, falls für alle  $x, y, z \in M$  gilt:

$$xRy \wedge xRz \Rightarrow y = z$$

- $R$  heißt **linkseindeutig**, falls für alle  $x, y, z \in M$  gilt:

$$xRz \wedge yRz \Rightarrow x = y$$

- Der **Nachbereich**  $N(R)$  und der **Vorbereich**  $V(R)$  von  $R$  sind

$$N(R) = \bigcup_{x \in M} R[x] \quad \text{und} \quad V(R) = \bigcup_{x \in M} R^T[x]$$

- $R$  ist also genau dann rechtseindeutig, wenn jedes Element  $x \in M$  höchstens einen Nachfolger hat, also  $R[x]$  höchstens einelementig ist,
- und genau dann linkseindeutig, wenn jedes Element  $x \in M$  höchstens einen Vorgänger hat, also  $R^{-1}[x]$  höchstens einelementig ist



Abbildungen ordnen jedem Element ihres Definitionsbereichs genau ein Element zu

## Definition

Eine rechtseindeutige Relation  $R$  mit  $V(R) = A$  und  $N(R) \subseteq B$  heißt **Abbildung** oder **Funktion von  $A$  nach  $B$**  (kurz  $R : A \rightarrow B$ )

## Bemerkung

- Wie üblich werden wir Abbildungen meist mit kleinen Buchstaben  $f, g, h, \dots$  bezeichnen und für  $(x, y) \in f$  nicht  $xfy$  sondern  $f(x) = y$  oder  $f : x \mapsto y$  schreiben
- Ist  $f : A \rightarrow B$  eine Abbildung, so wird der Vorbereich  $V(f) = A$  der **Definitionsbereich** und die Menge  $B$  der **Wertebereich** oder **Wertevorrat** von  $f$  genannt
- Der Nachbereich  $N(f)$  wird als **Bild** von  $f$  bezeichnet

## Definition

Sei  $f : A \rightarrow B$  eine Abbildung

- Im Fall  $N(f) = B$  heißt  $f$  **surjektiv**
- Ist  $f$  linkseindeutig, so heißt  $f$  **injektiv**
- In diesem Fall impliziert  $f(x) = f(y)$  die Gleichheit  $x = y$
- Eine injektive und surjektive Abbildung heißt **bijektiv**
- Ist  $f$  injektiv, so ist auch  $f^{-1} : N(f) \rightarrow A$  eine Abbildung, die als die zu  $f$  **inverse Abbildung** bezeichnet wird

## Bemerkung

Man beachte, dass der Definitionsbereich  $V(f^{-1}) = N(f)$  von  $f^{-1}$  nur dann gleich  $B$  ist, wenn  $f$  auch surjektiv, also eine Bijektion ist

## Definition

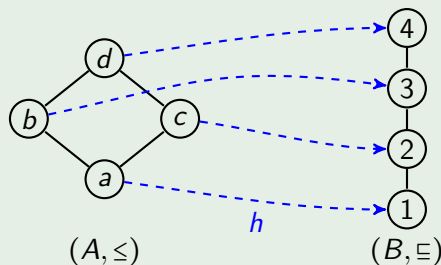
Seien  $(A_1, R_1)$  und  $(A_2, R_2)$  Relationalstrukturen

- Eine Abbildung  $h: A_1 \rightarrow A_2$  heißt **Homomorphismus**, falls für alle  $a, b \in A_1$  gilt:

$$aR_1b \Rightarrow h(a)R_2h(b)$$

- Sind  $(A_1, R_1)$  und  $(A_2, R_2)$  Ordnungen, so spricht man auch von **Ordnungshomomorphismen** oder einfach von **monotonen** Abbildungen
- Injektive Ordnungshomomorphismen werden auch **streng monotone** Abbildungen genannt

## Beispiel



- Die Abbildung  $h : A \rightarrow B$  ist ein bijektiver Ordnungshomomorphismus (also eine monotone Bijektion) zwischen  $(A, \leq)$  und  $(B, \subseteq)$
- Die Umkehrabbildung  $h^{-1}$  ist jedoch nicht monoton, da zwar  $2 \subseteq 3$ , aber  $h^{-1}(2) = b \not\subseteq c = h^{-1}(3)$  gilt
- Dagegen ist für jede monotone Bijektion  $f$  zwischen **linearen** Ordnungen auch ihre Umkehrabbildung  $f^{-1}$  monoton.

## Definition

- Seien  $(A_1, R_1)$  und  $(A_2, R_2)$  Relationalstrukturen
- Ein bijektiver Homomorphismus  $h: A_1 \rightarrow A_2$ , bei dem auch  $h^{-1}$  ein Homomorphismus ist, d.h. es gilt für alle  $a, b \in A_1$ ,

$$aR_1b \Leftrightarrow h(a)R_2h(b)$$

heißt **Isomorphismus**

- In diesem Fall heißen die Strukturen  $(A_1, R_1)$  und  $(A_2, R_2)$  **isomorph** (kurz:  $(A_1, R_1) \cong (A_2, R_2)$ )

Sind  $(A_1, R_1)$  und  $(A_2, R_2)$  isomorph, so bedeutet dies, dass sich die beiden Strukturen nur in der Benennung ihrer Elemente unterscheiden

# Isomorphismen

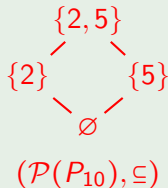
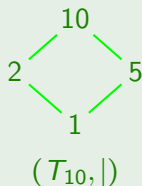
## Beispiel

- Die Abbildung  $h: x \mapsto e^x$  ist ein Isomorphismus zwischen den linearen Ordnungen  $(\mathbb{R}, \leq)$  und  $(\mathbb{R}^+, \leq)$
- Für  $n \in \mathbb{N}$  sei

$$T_n = \{k \in \mathbb{N} \mid k \text{ teilt } n\}$$

und

$$P_n = \{p \in T_n \mid p \text{ ist prim}\}$$



- Dann ist die Abbildung

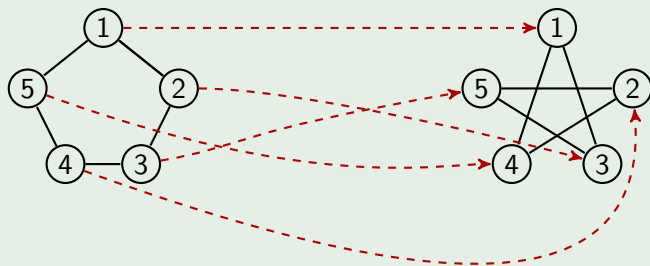
$$h: k \mapsto P_k$$

ein (surjektiver) Ordnungshomomorphismus von  $(T_n, |)$  auf  $(\mathcal{P}(P_n), \subseteq)$

- $h$  ist sogar ein Isomorphismus, falls  $n$  **quadratifrei** ist (d.h. es gibt keine Primzahl  $p$ , so dass  $p^2$  die Zahl  $n$  teilt)



## Beispiel


 $G = (V, E)$ 

$v$	1	2	3	4	5
$h_1(v)$	1	3	5	2	4
$h_2(v)$	1	4	2	5	3

 $G' = (V, E')$ 

- Die beiden Graphen  $G$  und  $G'$  sind isomorph
- Zwei Isomorphismen sind beispielsweise  $h_1$  und  $h_2$ .

## Beispiel

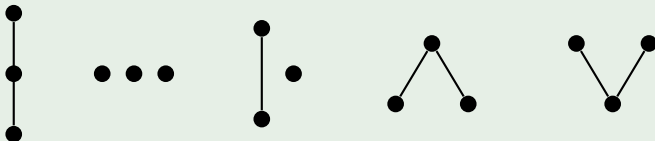
- Während auf der Knotenmenge  $V = \{1, 2, 3\}$  insgesamt  $2^{\binom{3}{2}} = 2^3 = 8$  verschiedene Graphen existieren, gibt es auf dieser Menge nur 4 verschiedene nichtisomorphe Graphen:





## Beispiel

- Es existieren genau 5 nichtisomorphe Ordnungen mit 3 Elementen:



- Anders ausgedrückt: Die Klasse aller dreielementigen Ordnungen zerfällt unter der Isomorphierelation  $\cong$  in fünf Äquivalenzklassen, die durch obige fünf Hasse-Diagramme repräsentiert werden.



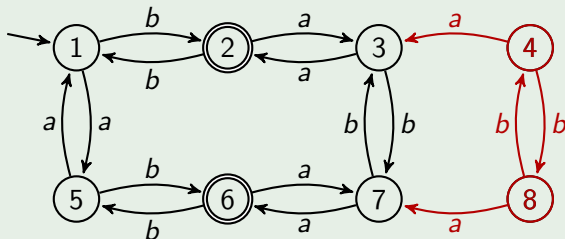
# Minimierung von DFAs

## Frage

Wie können wir feststellen, ob ein DFA  $M = (Z, \Sigma, \delta, q_0, E)$  eine minimale Anzahl von Zuständen besitzt (und  $Z$  evtl. verkleinern)?

## Beispiel

- Betrachte den DFA  $M$



- Zunächst können alle Zustände entfernt werden, die vom Startzustand aus **unerreichbar** sind.



## Frage

Wie können wir feststellen, ob ein DFA  $M = (Z, \Sigma, \delta, q_0, E)$  eine minimale Anzahl von Zuständen besitzt (und  $Z$  evtl. verkleinern)?

## Antwort

- Zunächst können alle unerreichbaren Zustände entfernt werden
- Zudem lassen sich zwei Zustände  $p$  und  $q$  verschmelzen, wenn  $M$  von  $p$  und  $q$  aus jeweils dieselben Wörter akzeptiert
- Für  $z \in Z$  sei

$$M_z = (Z, \Sigma, \delta, z, E).$$

- Dann können wir  $p$  und  $q$  verschmelzen (in Zeichen:  $p \sim_M q$ ), wenn  $L(M_p) = L(M_q)$  ist
- Offensichtlich ist  $\sim_M$  eine Äquivalenzrelation auf  $Z$

# Minimierung von DFAs

## Idee

Verschmelze jeden Zustand  $q$  mit allen äquivalenten Zuständen  $p \sim_M q$  zu einem neuen Zustand

## Notation

- Für die durch  $q$  repräsentierte Äquivalenzklasse

$$[q]_{\sim_M} = \{p \in Z \mid p \sim_M q\} = \{p \in Z \mid L(M_p) = L(M_q)\}$$

schreiben wir auch einfach  $[q]$  oder  $\tilde{q}$

- Für eine Teilmenge  $Q \subseteq Z$  bezeichne

$$\tilde{Q} = \{\tilde{q} \mid q \in Q\}$$

die Menge aller Äquivalenzklassen, die einen Zustand in  $Q$  enthalten

- Dann führt obige Idee auf den DFA

$$M' = (\tilde{Z}, \Sigma, \delta', \tilde{q}_0, \tilde{E}) \quad \text{mit} \quad \delta'(\tilde{q}, a) = \widetilde{\delta(q, a)},$$

wobei zu zeigen ist, dass  $\delta'(\tilde{q}, a)$  nicht von der Wahl des Repräsentanten  $q$  für die Äquivalenzklasse  $\tilde{q}$  abhängt

## Wie können wir $M'$ aus $M$ berechnen?

- Es genügt, die Äquivalenzrelation  $\sim_M$  auf der Zustandsmenge  $Z$  zu berechnen
- Hierzu genügt es, die Menge  $D = \{\{p, q\} \subseteq Z \mid p \not\sim_M q\}$  zu berechnen
- Sei  $A \Delta B = (A \setminus B) \cup (B \setminus A)$  die **symmetrische Differenz** zweier Mengen  $A$  und  $B$ . Dann gilt

$$p \not\sim_M q \Leftrightarrow L(M_p) \neq L(M_q) \Leftrightarrow L(M_p) \Delta L(M_q) \neq \emptyset$$

- Wörter  $x \in L(M_p) \Delta L(M_q)$  heißen **Unterscheider** zwischen  $p$  und  $q$
- Für  $i \geq 0$  sei  $D^{=i}$  die Menge aller Paare  $\{p, q\} \in D$ , die einen Unterscheider  $x$  der Länge  $|x| = i$  haben, und  $D_i$  sei die Menge aller Paare  $\{p, q\} \in D$ , die einen Unterscheider  $x$  der Länge  $|x| \leq i$  haben
- Dann gilt
  - $D_i = D^{=0} \cup D^{=1} \cup \dots \cup D^{=i}$  und
  - $D = \bigcup_{j \geq 0} D^{=j} = \bigcup_{i \geq 0} D_i$

- Offenbar unterscheidet  $\varepsilon$  Endzustände und Nichtendzustände, d.h.

$$D_0 = D^0 = \left\{ \{p, q\} \subseteq Z \mid p \in E, q \notin E \right\}$$

- Zudem gilt

$$\{p, q\} \in D^{i+1} \Leftrightarrow \exists a \in \Sigma : \{\delta(p, a), \delta(q, a)\} \in D^i,$$

da 2 Zustände  $p$  und  $q$  genau dann einen Unterscheider  $x = x_1 \dots x_{i+1}$  der Länge  $i + 1$  haben, wenn die beiden Zustände  $\delta(p, x_1)$  und  $\delta(q, x_1)$  einen Unterscheider  $x = x_2 \dots x_{i+1}$  der Länge  $i$  haben

- Folglich ist

$$D_{i+1} = \underbrace{D_i}_{D=0 \cup \dots \cup D=i} \cup \underbrace{\left\{ \{p, q\} \subseteq Z \mid \exists a \in \Sigma : \{\delta(p, a), \delta(q, a)\} \in D_i \right\}}_{D=1 \cup \dots \cup D=i+1}$$

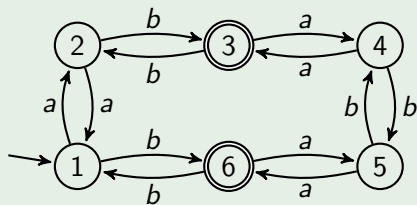
- Da es nur endlich viele Zustandspaare gibt, gibt es ein  $i \geq 0$  mit  $D = D_i$
- Offensichtlich gilt:  $D = D_i \Leftrightarrow D_{i+1} = D_i$

**Algorithmus** min-DFA( $M$ )

1    **Input:** DFA  $M = (Z, \Sigma, \delta, q_0, E)$   
2    entferne alle unerreichbaren Zustände aus  $Z$   
3     $D' := \{\{p, q\} \subseteq Z \mid p \in E, q \notin E\}$   
4    **repeat**  
5        $D := D'$   
6        $D' := D \cup \{\{p, q\} \mid \exists a \in \Sigma : \{\delta(p, a), \delta(q, a)\} \in D\}$   
7    **until**  $D' = D$   
8    **Output:**  $M' = (\tilde{Z}, \Sigma, \delta', \tilde{q}_0, \tilde{E})$ , wobei  $\delta'(\tilde{q}, a) = \overline{\delta(q, a)}$  ist  
9    und für jeden Zustand  $q \in Z$  gilt:  $\tilde{q} = \{p \in Z \mid \{p, q\} \notin D\}$

## Beispiel

Betrachte den DFA  $M$



2					
3	$\epsilon$	$\epsilon$			
4			$\epsilon$		
5			$\epsilon$		
6	$\epsilon$	$\epsilon$		$\epsilon$	$\epsilon$
	1	2	3	4	5

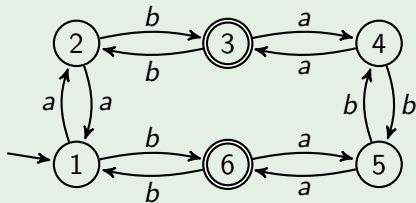
Dann enthält  $D_0$  die Paare

$\{1, 3\}, \{1, 6\}, \{2, 3\}, \{2, 6\}, \{3, 4\}, \{3, 5\}, \{4, 6\}, \{5, 6\}$ .



## Algorithmus zur Minimierung eines DFA

## Beispiel

Betrachte den DFA  $M$ 

2					
3	$\epsilon$	$\epsilon$			
4	$a$	$a$	$\epsilon$		
5	$a$	$a$	$\epsilon$		
6	$\epsilon$	$\epsilon$		$\epsilon$	$\epsilon$
	1	2	3	4	5

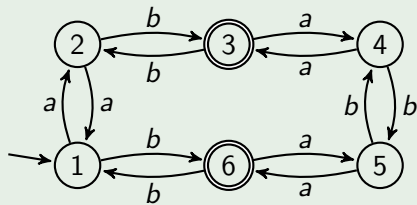
Wegen

$\{p, q\}$	$\{1, 4\}$	$\{1, 5\}$	$\{2, 4\}$	$\{2, 5\}$
$\{\delta(q, a), \delta(p, a)\}$	$\{2, 3\}$	$\{2, 6\}$	$\{1, 3\}$	$\{1, 6\}$

enthält  $D_1$  zusätzlich die Paare  $\{1, 4\}$ ,  $\{1, 5\}$ ,  $\{2, 4\}$ ,  $\{2, 5\}$ .

## Algorithmus zur Minimierung eines DFA

## Beispiel

Betrachte den DFA  $M$ 

2					
3	$\epsilon$	$\epsilon$			
4	$a$	$a$	$\epsilon$		
5	$a$	$a$	$\epsilon$		
6	$\epsilon$	$\epsilon$		$\epsilon$	$\epsilon$
	1	2	3	4	5

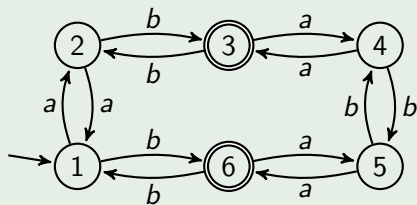
Da nun jedoch keines der verbliebenen Paare  $\{1,2\}$ ,  $\{3,6\}$ ,  $\{4,5\}$  wegen

$\{p, q\}$	$\{1,2\}$	$\{3,6\}$	$\{4,5\}$
$\{\delta(p, a), \delta(q, a)\}$	$\{1,2\}$	$\{4,5\}$	$\{3,6\}$
$\{\delta(p, b), \delta(q, b)\}$	$\{3,6\}$	$\{1,2\}$	$\{4,5\}$

zu  $D_2$  gehört, ist  $D_2 = D_1$  und somit  $D = D_1$ .

## Beispiel

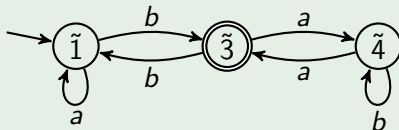
Betrachte den DFA  $M$



2					
3	$\epsilon$	$\epsilon$			
4	$a$	$a$	$\epsilon$		
5	$a$	$a$	$\epsilon$		
6	$\epsilon$	$\epsilon$		$\epsilon$	$\epsilon$
	1	2	3	4	5

Da die Paare  $\{1, 2\}$ ,  $\{3, 6\}$  und  $\{4, 5\}$  nicht in  $D$  enthalten sind, können die Zustände 1 und 2, 3 und 6, sowie 4 und 5 verschmolzen werden.

Demnach hat  $M'$  die Zustände  $\tilde{1} = \{1, 2\}$ ,  $\tilde{3} = \{3, 6\}$  und  $\tilde{4} = \{4, 5\}$ :



## Satz

Sei  $M = (Z, \Sigma, \delta, q_0, E)$  ein DFA ohne unerreichbare Zustände. Dann ist

$$M' = (\tilde{Z}, \Sigma, \delta', \tilde{q}_0, \tilde{E}) \text{ mit } \delta'(\tilde{q}, a) = \overline{\delta(q, a)}$$

ein DFA für  $L(M)$  mit einer minimalen Anzahl von Zuständen

## Beweis

Wir beweisen den Satz durch folgende 3 Behauptungen:

**Beh. 1:**  $\delta'$  ist wohldefiniert, da  $\delta'(\tilde{q}, a) = \overline{\delta(q, a)}$  nicht von der Wahl des Repräsentanten  $q$  für die Äquivalenzklasse  $\tilde{q}$  abhängt.

**Beh. 2:**  $L(M') = L(M)$

**Beh. 3:**  $M'$  hat eine minimale Anzahl von Zuständen

## Behauptung 1

$\delta'$  ist wohldefiniert, da  $\delta'(\tilde{q}, a) = \overline{\delta(q, a)}$  nicht von der Wahl des Repräsentanten  $q$  für die Äquivalenzklasse  $\tilde{q}$  abhängt

## Beweis von Behauptung 1

- Hierzu ist die Implikation  $p \sim_M q \Rightarrow \delta(p, a) \sim_M \delta(q, a)$  zu zeigen
- Diese ist äquivalent zur Kontraposition  $\delta(p, a) \not\sim_M \delta(q, a) \Rightarrow p \not\sim_M q$ , die wir bereits gezeigt haben:

$$\begin{aligned} \delta(p, a) \not\sim_M \delta(q, a) &\Rightarrow \text{es gibt einen Unterscheider } x \\ &\quad \text{zwischen } \delta(p, a) \text{ und } \delta(q, a) \\ &\Rightarrow \text{es gibt einen Unterscheider } ax \\ &\quad \text{zwischen } p \text{ und } q \\ &\Rightarrow p \not\sim_M q \end{aligned}$$

□

## Behauptung 2

$$L(M') = L(M)$$

## Beweis von Behauptung 2

- Sei  $x = x_1 \dots x_n \in \Sigma^*$  eine Eingabe und seien  $q_0, q_1, \dots, q_n$  die von  $M(x)$  besuchten Zustände, d.h. es gilt  $\delta(q_{i-1}, x_i) = q_i$  für  $i = 1, \dots, n$
- Nach Definition von  $\delta'$  folgt daher  $\delta'(\tilde{q}_{i-1}, x_i) = \tilde{q}_i$  für  $i = 1, \dots, n$ , d.h.  $M'$  besucht bei Eingabe  $x$  die Zustände  $\tilde{q}_0, \tilde{q}_1, \dots, \tilde{q}_n$
- Da aber  $\tilde{q}_n$  entweder nur End- oder nur Nicht-Endzustände enthält, gehört  $q_n$  genau dann zu  $E$ , wenn  $\tilde{q}_n$  zu  $\tilde{E}$  gehört, d.h. es gilt

$$x \in L(M) \Leftrightarrow x \in L(M').$$



## Behauptung 3

$M'$  hat eine minimale Anzahl von Zuständen

## Beweis von Behauptung 3

- Wegen  $\|\tilde{Z}\| \leq \|Z\|$  ist  $M'$  sicher dann minimal, wenn  $M$  minimal ist
- Es reicht also zu zeigen, dass die Anzahl  $\|\tilde{Z}\| = \text{index}(\sim_M)$  der Zustände von  $M'$  nur von der erkannten Sprache  $L = L(M)$  abhängt
- Wegen  $p \sim_M q \Leftrightarrow L(M_p) = L(M_q)$  gilt  $\text{index}(\sim_M) = \|\{L(M_z) \mid z \in Z\}\|$
- Für jedes Wort  $x \in \Sigma^*$  sei

$L_x = \{y \in \Sigma^* \mid xy \in L\}$  die Restsprache von  $L$  für das Präfix  $x$ .

- Dann gilt  $\{L_x \mid x \in \Sigma^*\} = \{L(M_z) \mid z \in Z\}$ :
  - $\subseteq$ : Klar, da  $L_x = L(M_z)$  für  $z = \hat{\delta}(q_0, x)$  ist
  - $\supseteq$ : Auch klar, da jedes  $z \in Z$  über ein  $x \in \Sigma^*$  erreichbar ist
- Also hängt  $\text{index}(\sim_M) = \|\{L_x \mid x \in \Sigma^*\}\|$  nur von  $L$  ab □

## Beispiel

- Die Sprache

$$L = \{x_1 \dots x_n \in \{0, 1\}^* \mid n \geq 2 \text{ und } x_{n-1} = 0\}$$

hat die vier Restsprachen

$$L_x = \begin{cases} L, & x \in \{\varepsilon, 1\} \text{ oder } x \text{ endet mit } 11, \\ L \cup \{0, 1\}, & x = 0 \text{ oder } x \text{ endet mit } 10, \\ L \cup \{\varepsilon, 0, 1\}, & x \text{ endet mit } 00, \\ L \cup \{\varepsilon\}, & x \text{ endet mit } 01 \end{cases}$$

- Entsprechend gibt es für  $L$  einen DFA mit 4 Zuständen, aber keinen mit 3 Zuständen.





# Die Nerode-Relation einer Sprache $L$

- Eine interessante Folgerung aus obigem Beweis ist, dass eine reguläre Sprache  $L \subseteq \Sigma^*$  nur endlich viele Restsprachen hat
- Daraus folgt, dass die durch

$$x \sim_L y \iff L_x = L_y$$

auf  $\Sigma^*$  definierte Äquivalenzrelation  $\sim_L$  für jede reguläre Sprache  $L \subseteq \Sigma^*$  einen endlichen Index hat

- Die Relation  $\sim_L$  wird als **Nerode-Relation** von  $L$  bezeichnet

- Sei  $L = L(M)$  für einen DFA  $M = (Z, \Sigma, \delta, q_0, E)$  und für  $x \in \Sigma^*$  sei  $L_x = \{z \in \Sigma^* \mid xz \in L\}$  die Restsprache von  $L$  für  $x$ .

- Zudem sei  $M' = (\tilde{Z}, \Sigma, \delta', \tilde{q}_0, \tilde{E})$  der zu  $M$  gehörige Minimal-DFA

- Dieser erreicht nach Lesen eines Wortes  $x$  den Zustand  $\widehat{\delta}(q_0, x)$

- Wegen

$$\begin{aligned} \widehat{\delta}(q_0, x) = \widehat{\delta}(q_0, y) &\Leftrightarrow \widehat{\delta}(q_0, x) \sim_M \widehat{\delta}(q_0, y) \\ &\Leftrightarrow L(M_{\widehat{\delta}(q_0, x)}) = L(M_{\widehat{\delta}(q_0, y)}) \Leftrightarrow L_x = L_y \end{aligned}$$

können wir den Zustand  $\widehat{\delta}(q_0, x)$  von  $M'$  auch mit  $L_x$  bezeichnen

- Dies führt auf den zu  $M'$  isomorphen DFA  $M_L = (Z_L, \Sigma, \delta_L, L_\epsilon, E_L)$  mit

$$Z_L = \{L_x \mid x \in \Sigma^*\}, \quad E_L = \{L_x \mid x \in L\} \quad \text{und} \quad \delta_L(L_x, a) = L_{xa},$$

der sich direkt aus der Sprache  $L$  gewinnen lässt

- $M_L$  wird auch als Restsprachen-DFA für  $L$  bezeichnet

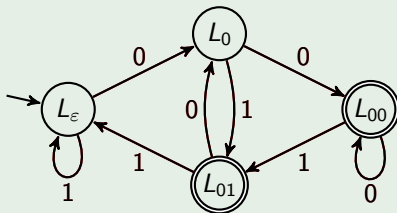
## Beispiel

- Die Sprache  $L = \{x_1 \dots x_n \in \{0, 1\}^* \mid n \geq 2 \text{ und } x_{n-1} = 0\}$  hat die Restsprachen

$$L_x = \begin{cases} L, & x \in \{\varepsilon, 1\} \text{ oder } x \text{ endet mit } 11, \\ L \cup \{0, 1\}, & x = 0 \text{ oder } x \text{ endet mit } 10, \\ L \cup \{\varepsilon, 0, 1\}, & x \text{ endet mit } 00, \\ L \cup \{\varepsilon\}, & x \text{ endet mit } 01 \end{cases}$$

- Konstruktion von  $M_L$ :

$L_x$	$L_\varepsilon$	$L_0$	$L_{00}$	$L_{01}$
$L_{x0}$	$L_0$	$L_{00}$	$L_{00}$	$L_0$
$L_{x1}$	$L_\varepsilon$	$L_{01}$	$L_{01}$	$L_\varepsilon$



- Für die Konstruktion von  $M_L$  spielt es keine Rolle, wie die Restsprachen  $L_x$  konkret aussehen, d.h. ihre Bestimmung ist nicht erforderlich. ◀

## Der Satz von Myhill und Nerode

- Notwendig und hinreichend für die Existenz von  $M_L$  ist, dass die Menge  $Z_L = \{L_x \mid x \in \Sigma^*\}$  aller Restsprachen von  $L$  endlich ist
- Dies ist genau dann der Fall, wenn die durch  $L$  auf  $\Sigma^*$  definierte **Nerode-Relation**  $\sim_L$  mit

$$x \sim_L y \Leftrightarrow L_x = L_y$$

einen endlichen Index hat

- Ist  $M$  bereits minimal, so haben die Zustände des zugehörigen Minimal-DFA  $M'$  die Form  $\tilde{q} = \{q\}$ , d.h.  $M'$  und  $M$  sind isomorph
- Da  $M'$  wiederum isomorph zu  $M_L$  ist, ist jeder minimale DFA  $M$  mit  $L(M) = L$  isomorph zu  $M_L$
- Folglich gibt es für jede reguläre Sprache  $L$  bis auf Isomorphie nur einen Minimal-DFA

# Der Satz von Myhill und Nerode

## Satz (Myhill und Nerode)

- 1 Für jede reguläre Sprache  $L$  gibt es bis auf Isomorphie nur einen Minimal-DFA. Dieser hat  $index(\sim_L)$  Zustände
- 2  $REG = \{L \mid index(\sim_L) < \infty\}$

## Bemerkung

- Sei  $R$  ein Repräsentantensystem für die Nerode-Relation  $\sim_L$  von  $L$ , d.h.  $\{L_x \mid x \in \Sigma^*\} = \{L_r \mid r \in R\}$  und  $L_r \neq L_{r'}$  für alle  $r, r' \in R$  mit  $r \neq r'$
- Dann können wir die Zustände des Minimal-DFA anstelle von  $L_x$  auch mit den Repräsentanten  $r \in R$  bezeichnen
- Dies führt auf den Minimal-DFA  $M_R = (R, \Sigma, \delta, \varepsilon, E)$ , wobei wir  $\varepsilon \in R$  annehmen und  $\delta(r, a) \in R$  der Repräsentant der Äquivalenzklasse  $\tilde{ra} = \{x \in \Sigma^* \mid x \sim_L ra\}$  sowie  $E = R \cap L$  ist
- Wir bezeichnen  $M_R$  als den zu  $R$  gehörigen **Repräsentanten-DFA** für  $L$

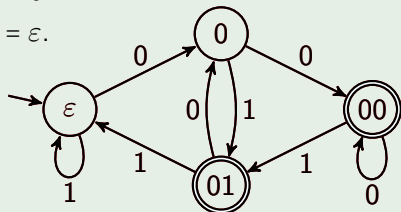
Direkte Konstruktion von  $M_R$  aus  $L$ 

## Beispiel

Für die Sprache  $L = \{x_1 \dots x_n \in \{0, 1\}^* \mid n \geq 2 \text{ und } x_{n-1} = 0\}$  lässt sich ein Repräsentanten-DFA  $M_R$  wie folgt konstruieren:

- 1 Sei  $r_1 = \varepsilon$ . Wegen  $r_1 0 = 0 \not\sim_L r_1$  ist  $r_2 = 0$  und  $\delta(\varepsilon, 0) = 0$ .
- 2 Wegen  $r_1 1 = 1 \sim_L \varepsilon$  ist  $\delta(\varepsilon, 1) = \varepsilon$ .
- 3 Wegen  $r_2 0 = 00 \not\sim_L r_i$  für  $i = 1, 2$  ist  $r_3 = 00$  und  $\delta(0, 0) = 00$ .
- 4 Wegen  $r_2 1 = 01 \not\sim_L r_i$  für  $i = 1, 2, 3$  ist  $r_4 = 01$  und  $\delta(0, 1) = 01$ .
- 5 Wegen  $r_3 0 = 000 \sim_L 00$  ist  $\delta(00, 0) = 00$ .
- 6 Wegen  $r_3 1 = 001 \sim_L 01$  ist  $\delta(00, 1) = 01$ .
- 7 Wegen  $r_4 0 = 010 \sim_L 0$  ist  $\delta(01, 0) = 0$ .
- 8 Wegen  $r_4 1 = 011 \sim_L \varepsilon$  ist  $\delta(01, 1) = \varepsilon$ .

$r$	$\varepsilon$	0	00	01
$\delta(r, 0)$	0	00	00	0
$\delta(r, 1)$	$\varepsilon$	01	01	$\varepsilon$



## Korollar

Sei  $L \subseteq \Sigma^*$  eine Sprache. Dann sind folgende Aussagen äquivalent:

- $L$  ist regulär (d.h. 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  $L = L(\gamma)$ ,
- die Nerode-Relation  $\sim_L$  von  $L$  auf  $\Sigma^*$  hat endlichen Index

Wir können also beweisen, dass eine Sprache  $L$  **nicht** regulär ist, indem wir

- unendlich viele verschiedene Restsprachen finden, bzw.
- unendlich viele Wörter finden, die paarweise inäquivalent bzgl.  $\sim_L$  sind

## Satz

Die Sprache  $L = \{a^n b^n \mid n \geq 0\}$  ist nicht regulär

## Beweis

- Wegen

$$b^i \in L_{a^i} \Delta L_{a^j} \quad (\text{für } 0 \leq i < j)$$

sind die Restsprachen  $L_{a^i}$ ,  $i \geq 0$ , paarweise verschieden.

- Wegen

$$a^i \sim_L a^j \Leftrightarrow L_{a^i} = L_{a^j}$$

folgt auch, dass  $a^i \not\sim_L a^j$  für  $i < j$  gilt und somit  $\text{index}(\sim_L) = \infty$  ist.  $\square$



## Frage

Gibt es noch andere Methoden, um zu zeigen, dass eine Sprache nicht regulär ist?

## Antwort

Oft führt die Kontraposition folgender Aussage zum Ziel

## Satz (Pumping-Lemma für reguläre Sprachen)

Zu jeder regulären Sprache  $L$  gibt es eine Zahl  $l \geq 0$ , so dass sich alle Wörter  $x \in L$  mit  $|x| \geq l$  in  $x = uvw$  zerlegen lassen mit

- 1  $v \neq \varepsilon$ ,
- 2  $|uv| \leq l$  und
- 3  $uv^i w \in L$  für alle  $i \geq 0$

Das kleinste solche  $l$  wird auch die **Pumpingzahl** von  $L$  genannt

# Das Pumping-Lemma

## Beispiel

- Die Sprache

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

lässt sich „pumpen“ (mit Pumpingzahl  $l = 3$ )

- Sei  $x \in L$  beliebig mit  $|x| \geq 3$ 
  - 1. Fall:  $x$  hat das Präfix ab.  
 Zerlege  $x = uvw$  mit  $u = \varepsilon$  und  $v = ab$ .
  - 2. Fall:  $x$  hat das Präfix aab.  
 Zerlege  $x = uvw$  mit  $u = a$  und  $v = ab$ .
  - 3. Fall:  $x$  hat das Präfix aaa.  
 Zerlege  $x = uvw$  mit  $u = \varepsilon$  und  $v = aaa$ .
  - Restliche Fälle (Präfixe ba, bba und bbb): analog

## Beispiel

- Sei  $L$  eine **endliche** Sprache
- Offenbar lässt sich kein Wort  $x \in L$  „pumpen“
- Dennoch hat  $L$  eine endliche Pumpingzahl
- Sei nämlich

$$l_{max} = \begin{cases} \max_{x \in L} |x|, & L \neq \emptyset, \\ -1, & \text{sonst.} \end{cases}$$

- Dann lässt sich jedes Wort  $x \in L$  der Länge  $|x| > l_{max}$  „pumpen“, da solche Wörter gar nicht existieren
- Also ist die Pumpingzahl von  $L$  höchstens  $l_{max} + 1$
- Zudem ist die Pumpingzahl von  $L$  größer als  $l_{max}$ , da es im Fall  $l_{max} \geq 0$  ein Wort  $x \in L$  der Länge  $|x| = l_{max}$  gibt, das nicht „pumpbar“ ist
- Also hat  $L$  die Pumpingzahl  $l_{max} + 1$

# Das Pumping-Lemma

## Satz (Pumping-Lemma für reguläre Sprachen)

Zu jeder regulären Sprache  $L$  gibt es eine Zahl  $l \geq 0$ , so dass sich alle Wörter  $x \in L$  mit  $|x| \geq l$  in  $x = uvw$  zerlegen lassen mit

- ①  $v \neq \varepsilon$ ,
- ②  $|uv| \leq l$  und
- ③  $uv^i w \in L$  für alle  $i \geq 0$

Das kleinste solche  $l$  wird auch die **Pumpingzahl** von  $L$  genannt

## Beweis

- Sei  $M = (Z, \Sigma, \delta, q_0, E)$  ein DFA für  $L$  mit  $l$  Zuständen und sei  $x = x_1 \dots x_n \in L$  mit  $n = |x| \geq m$
- Dann muss  $M(x)$  nach spätestens  $l$  Schritten einen Zustand zum zweiten Mal annehmen, d.h. es ex  $0 \leq j < k \leq l$  und  $z \in Z$  mit

$$\hat{\delta}(q_0, x_1 \dots x_j) = z \text{ und}$$

$$\hat{\delta}(q_0, x_1 \dots x_j x_{j+1} \dots x_k) = z$$

## Beweis

- Sei  $M = (Z, \Sigma, \delta, q_0, E)$  ein DFA für  $L$  mit  $l$  Zuständen und sei  $x = x_1 \dots x_n \in L$  mit  $n = |x| \geq l$
- Dann muss  $M(x)$  nach spätestens  $l$  Schritten einen Zustand zum zweiten Mal annehmen, d.h. es ex  $0 \leq j < k \leq l$  und  $z \in Z$  mit

$$\hat{\delta}(q_0, x_1 \dots x_j) = z \text{ und}$$

$$\hat{\delta}(q_0, x_1 \dots x_j x_{j+1} \dots x_k) = z$$

- Setze  $u = x_1 \dots x_j$ ,  $v = x_{j+1} \dots x_k$  und  $w = x_{k+1} \dots x_n$ .
- Dann gilt  $|v| = k - j \geq 1$  (d.h.  $v \neq \varepsilon$ ) und  $|uv| = k \leq l$ .
- Zudem gehört für alle  $i \geq 0$  das Wort  $uv^i w$  zu  $L$ , da wegen  $\hat{\delta}(z, v^i) = z$

$$\hat{\delta}(q_0, uv^i w) = \hat{\delta}(\underbrace{\hat{\delta}(\hat{\delta}(q_0, u), v^i)}_z, w) = \hat{\delta}(\underbrace{\hat{\delta}(\hat{\delta}(q_0, u), v)}_z, w) = \hat{\delta}(q_0, x)$$

in  $E$  ist.



## Bemerkung

- Sei  $\min_{DFA}(L)$  ( $\min_{NFA}(L)$ ) die minimale Anzahl von Zuständen eines DFA (bzw. NFA) einer regulären Sprache  $L$  und sei  $l_{reg}(L)$  die Pumping-Zahl für  $L$
- Da wir im Beweis des Pumping-Lemmas auch einen NFA für  $L$  mit  $l = \min_{NFA}(L)$  Zuständen wählen können, folgt

$$l_{reg}(L) \leq \min_{NFA}(L) \leq \min_{DFA}(L) = index(\sim_L)$$

- Tatsächlich gibt es für jedes  $i \geq 1$  eine Sprache  $L$  mit

$$l_{reg}(L) = index(\sim_L) = i$$

- Andererseits gibt es für jedes  $i \geq 1$  auch eine Sprache  $L$  mit

$$l_{reg}(L) = 1 \text{ und } index(\sim_L) = i$$

- Dagegen ist  $L = \emptyset$  die einzige Sprache mit der Pumping-Zahl 0
- Für diese gilt  $index(\sim_{\emptyset}) = 1$

Um also  $L \notin \text{REG}$  zu zeigen, genügt es,

- für jede Zahl  $l \geq 0$  ein Wort  $x \in L$  der Länge  $|x| \geq l$  zu finden, so dass
- für jede Zerlegung  $x = uvw$  mindestens eine der folgenden drei Bedingungen verletzt ist:
  - ❶  $v \neq \varepsilon$ ,
  - ❷  $|uv| \leq l$  oder
  - ❸  $uv^i w \in L$  für alle  $i \geq 0$

Beispiel:  $L = \{a^n b^n \mid n \geq 0\} \notin \text{REG}$

- Für jede Zahl  $l \geq 0$  enthält  $L$  das Wort  $x = a^l b^l$  mit  $|x| = 2l \geq l$
- Für jede Zerlegung  $x = uvw$  von  $x = a^l b^l$  mit
  - ❶  $v \neq \varepsilon$ist die Bedingung
  - ❸  $uv^i w \in L$für alle  $i \geq 2$  verletzt

# Kontraposition des Pumping-Lemmas

Um also  $L \notin \text{REG}$  zu zeigen, genügt es,

- für jede Zahl  $l \geq 0$  ein Wort  $x \in L$  der Länge  $|x| \geq l$  zu finden, so dass
- für jede Zerlegung  $x = uvw$  mindestens eine der folgenden drei Bedingungen verletzt ist:
  - ①  $v \neq \varepsilon$ ,
  - ②  $|uv| \leq l$  oder
  - ③  $uv^i w \in L$  für alle  $i \geq 0$

Beispiel:  $L = \{a^{n^2} \mid n \geq 0\} \notin \text{REG}$

- Für jede Zahl  $l \geq 0$  enthält  $L$  das Wort  $x = a^{l^2}$  mit  $|x| = l^2 \geq l$
- Für jede Zerlegung  $x = uvw$  mit  $|u| = r$ ,  $|v| = s$ ,  $|w| = t$  und
  - ①  $v \neq \varepsilon$  (d.h.  $s \geq 1$ ) sowie
  - ②  $|uv| \leq l$  (d.h.  $r + s \leq l$ )
 ist die Bedingung

③  $uv^2w \in L$

verletzt:

$$l^2 < \underbrace{l^2 + s}_{|uv^2w|} \leq l^2 + l < (l+1)^2$$



# Kontraposition des Pumping-Lemmas

Um also  $L \notin \text{REG}$  zu zeigen, genügt es,

- für jede Zahl  $l \geq 0$  ein Wort  $x \in L$  der Länge  $|x| \geq l$  zu finden, so dass
- für jede Zerlegung  $x = uvw$  mindestens eine der folgenden drei Bedingungen verletzt ist:
  - ①  $v \neq \varepsilon$ ,
  - ②  $|uv| \leq l$  oder
  - ③  $uv^i w \in L$  für alle  $i \geq 0$

Beispiel:  $L = \{a^p \mid p \text{ prim}\} \notin \text{REG}$

- Für jede Zahl  $l \geq 0$  enthält  $L$  ein Wort  $x$  mit  $|x| = p \geq l$
- Für jede Zerlegung  $x = uvw$  mit  $|v| = s$  und
  - ①  $v \neq \varepsilon$  (d.h.  $s \geq 1$ )
 ist die Bedingung
  - ③  $uv^i w \in L$
 wegen

$$|uv^i w| = p + (i - 1)s$$

für  $i = p + 1$  verletzt:

$$|uv^{p+1} w| = p + ps = p(s + 1)$$

## Bemerkung

- Mit dem Pumping-Lemma können nicht alle Sprachen  $L \notin \text{REG}$  als nicht regulär nachgewiesen werden, da seine Umkehrung falsch ist

- Betrachte die Sprache

$$L = \{a^i b^j c^k \mid i = 0 \text{ oder } j = k\}$$

- $L$  hat die Pumpingzahl  $l = 1$ , da jedes Wort  $x \in L$  mit Ausnahme von  $\varepsilon$  „gepumpt“ werden kann
- Allerdings ist  $L$  nicht regulär (siehe Übungen)

Mit Grammatiken lassen sich Sprachen oft sehr kompakt und elegant beschreiben. Implizit haben wir diese Methode bei der Definition der regulären Ausdrücke bereits benutzt.

### Beispiel

- Sei  $\Sigma = \{a_1, \dots, a_k\}$  ein Alphabet
- Dann lässt sich jeder reguläre Ausdruck über  $\Sigma$  aus dem Symbol  $R$  unter (eventuell mehrfacher) Anwendung folgender Ersetzungsregeln erzeugen:

$$R \rightarrow \emptyset$$

$$R \rightarrow \epsilon$$

$$R \rightarrow a_i, i = 1, \dots, k$$

$$R \rightarrow RR$$

$$R \rightarrow (R|R)$$

$$R \rightarrow (R)^*$$

# Definition einer Grammatik

## Definition

Eine **Grammatik** ist ein 4-Tupel  $G = (V, \Sigma, P, S)$ , wobei

- $V$  eine endliche Menge von **Variablen** (auch **Nichtterminalsymbole** genannt),
- $\Sigma$  das **Terminalalphabet**,
- $P \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$  eine endliche Menge von **Regeln** (oder **Produktionen**) und
- $S \in V$  die **Startvariable** ist

## Bemerkung

- $P$  ist also eine binäre Relation auf  $(V \cup \Sigma)^*$
- Für  $(u, v) \in P$  schreiben wir auch kurz  $u \rightarrow_G v$  bzw.  $u \rightarrow v$ , wenn die benutzte Grammatik aus dem Kontext ersichtlich ist
- Regeln der Form  $\varepsilon \rightarrow v$  sind nicht erlaubt

# Die von einer Grammatik erzeugte Sprache

- Ein Wort  $\beta \in (V \cup \Sigma)^*$  ist aus einem Wort  $\alpha \in (V \cup \Sigma)^+$  **in einem Schritt ableitbar** (kurz:  $\alpha \Rightarrow_G \beta$ ), falls eine Regel  $u \rightarrow_G v$  und Wörter  $l, r \in (V \cup \Sigma)^*$  existieren mit
 
$$\alpha = lur \text{ und } \beta = lvr$$
- Um aus einem Ableitungsschritt  $\alpha = lur \Rightarrow_G lvr = \beta$  eindeutig die benutzte Regel  $u \rightarrow_G v$  und das ersetzte Teilwort  $u$  von  $\alpha$  rekonstruieren zu können, notieren wir ihn meist in der Form  $\underline{lur} \Rightarrow_G lvr$
- Falls  $G$  aus dem Kontext ersichtlich ist, schreiben wir für  $\Rightarrow_G$  auch einfach  $\Rightarrow$
- Da  $\Rightarrow$  eine binäre Relation auf  $(V \cup \Sigma)^*$  ist, bezeichnet
  - $\Rightarrow^m$  das  $m$ -fache Relationenprodukt von  $\Rightarrow$  und
  - $\Rightarrow^*$  die reflexive, transitive Hülle von  $\Rightarrow$
- Die durch  $G$  **erzeugte Sprache** ist  $L(G) = \{x \in \Sigma^* \mid S \Rightarrow_G^* x\}$
- Ein Wort  $\alpha \in (V \cup \Sigma)^*$  mit  $S \Rightarrow_G^* \alpha$  heißt **Satzform** von  $G$

## Beispiel

- Sei  $\Sigma = \{0, 1\}$  und sei

$$G = (\{R\}, \Sigma', P, R)$$

die Grammatik für die Sprache  $RA_{\Sigma}$  aller regulären Ausdrücke über dem Alphabet  $\Sigma$  mit dem Terminalalphabet

$$\Sigma' = \Sigma \cup \{\emptyset, \epsilon, (, ), |, * \} = \{0, 1, \emptyset, \epsilon, (, ), |, * \}$$

und den Regeln

$$P: R \rightarrow 0, 1, \emptyset, \epsilon$$

$$R \rightarrow RR, (R|R), (R)^*$$

- In  $G$  lässt sich der reguläre Ausdruck  $(01)^*(\epsilon|\emptyset)$  in 8 Schritten ableiten:

$$\begin{aligned} \underline{R} &\Rightarrow \underline{RR} \Rightarrow (\underline{R})^* R \Rightarrow (\underline{RR})^* \underline{R} \Rightarrow (\underline{RR})^* (\underline{R|R}) \\ &\Rightarrow (\underline{0R})^* (\underline{R|R}) \Rightarrow (\underline{01})^* (\underline{R|R}) \Rightarrow (\underline{01})^* (\underline{\epsilon|R}) \Rightarrow (\underline{01})^* (\underline{\epsilon|\emptyset}) \end{aligned}$$



Man unterscheidet vier Typen von Grammatiken  $G = (V, \Sigma, P, S)$

## Definition

- ①  $G$  heißt vom Typ 3 oder regulär, falls für alle Regeln  $u \rightarrow v$  gilt:

$$u \in V \text{ und } v \in \Sigma V \cup \Sigma \cup \{\varepsilon\}$$

(d.h. alle Regeln haben die Form  $A \rightarrow aB$ ,  $A \rightarrow a$  oder  $A \rightarrow \varepsilon$ )

- ②  $G$  heißt vom Typ 2 oder kontextfrei, falls für alle Regeln  $u \rightarrow v$  gilt:

$$u \in V \quad \text{(d.h. alle Regeln haben die Form } A \rightarrow v \text{)}$$

- ③  $G$  heißt vom Typ 1 oder kontextsensitiv, falls für alle Regeln  $u \rightarrow v$  gilt:

$$|v| \geq |u| \quad \text{(mit Ausnahme der } \varepsilon\text{-Sonderregel, s. unten)}$$

- ④ Jede Grammatik ist automatisch vom Typ 0

## Die $\varepsilon$ -Sonderregel

In einer kontextsensitiven Grammatik ist auch die Regel  $S \rightarrow \varepsilon$  zulässig, falls das Startsymbol  $S$  nicht auf der rechten Seite einer Regel vorkommt

## Beispiel

- Sei  $G = (\{R\}, \{0, 1, \emptyset, \epsilon, (, ), |, * \}, P, R)$  wieder die Grammatik für die Sprache aller regulären Ausdrücke über  $\Sigma = \{0, 1\}$  mit den Regeln

$$P: \begin{aligned} R &\rightarrow 0, 1, \emptyset, \epsilon \\ R &\rightarrow RR, (R|R), (R)^* \end{aligned}$$

- Da  $G$  eine korrekte Grammatik ist, ist  $G$  vom Typ 0
- $G$  ist auch kontextsensitiv, da die linke Seite jeder Regel die Länge 1 und jede rechte Seite mindestens die Länge 1 hat (man beachte, dass  $\epsilon$  ein Terminal ist)
- Da auf der linken Seite jeder Regel eine einzelne Variable steht, ist  $G$  sogar kontextfrei
- Offenbar ist  $G$  aber keine reguläre Grammatik, da zwar die vier Regeln  $R \rightarrow 0, 1, \emptyset, \epsilon$  die geforderte Form haben, nicht jedoch die drei Regeln  $R \rightarrow RR, (R|R), (R)^*$



- Eine Sprache heißt vom Typ  $i$  bzw. **kontextfrei** oder **kontextsensitiv**, falls sie von einer entsprechenden Grammatik erzeugt wird
- Damit erhalten wir die neuen Sprachklassen

$$\text{CFL} = \{L(G) \mid G \text{ ist eine kontextfreie Grammatik}\}$$

und

*(context free languages)*

$$\text{CSL} = \{L(G) \mid G \text{ ist eine kontextsensitive Grammatik}\}$$

*(context sensitive languages)*

- Da die Klasse der Typ 0 Sprachen mit der Klasse der **rekursiv aufzählbaren** Sprachen übereinstimmt, bezeichnen wir diese Sprachklasse mit

$$\text{RE} = \{L(G) \mid G \text{ ist eine Grammatik}\}$$

*(recursively enumerable languages)*

- Wir werden bald beweisen, dass die Sprachklassen

$$\text{REG} \subset \text{CFL} \subset \text{CSL} \subset \text{RE}$$

eine Hierarchie bilden (d.h. die Inklusionen sind echt), die so genannte **Chomsky-Hierarchie**

- Zunächst rechtfertigen wir jedoch die Bezeichnung **regulär** für die regulären Grammatiken und für die von ihnen erzeugten Sprachen

## Satz

$$\text{REG} = \{L(G) \mid G \text{ ist eine reguläre Grammatik}\}$$

Beweis von  $\text{REG} \subseteq \{L(G) \mid G \text{ ist eine reguläre Grammatik}\}$

- Sei  $M = (Z, \Sigma, \delta, q_0, E)$  ein DFA
- Wir konstruieren eine reguläre Grammatik  $G$  mit  $L(G) = L(M)$
- Betrachte die Grammatik  $G = (V, \Sigma, P, S)$  mit  $V = Z$ ,  $S = q_0$  und

$$P = \{q \rightarrow ap \mid \delta(q, a) = p\} \cup \{q \rightarrow \varepsilon \mid q \in E\}$$

Beweis von  $\text{REG} \subseteq \{L(G) \mid G \text{ ist eine reguläre Grammatik}\}$ 

- Betrachte die Grammatik  $G = (V, \Sigma, P, S)$  mit  $V = Z$ ,  $S = q_0$  und

$$P = \{q \rightarrow ap \mid \delta(q, a) = p\} \cup \{q \rightarrow \varepsilon \mid q \in E\}$$

- Dann gilt für alle Wörter  $x = x_1 \dots x_n \in \Sigma^*$ :

$$x \in L(M) \Leftrightarrow \exists q_1, \dots, q_{n-1} \in Z \exists q_n \in E :$$

$$\delta(q_{i-1}, x_i) = q_i \text{ für } i = 1, \dots, n$$

$$\Leftrightarrow \exists q_1, \dots, q_n \in V :$$

$$q_{i-1} \rightarrow x_i q_i \text{ für } i = 1, \dots, n \text{ und } q_n \rightarrow \varepsilon$$

$$\Leftrightarrow \exists q_1, \dots, q_n \in V :$$

$$q_0 \Rightarrow^i x_1 \dots x_i q_i \text{ für } i = 1, \dots, n \text{ und } q_n \rightarrow \varepsilon$$

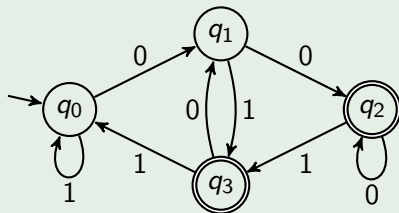
$$\Leftrightarrow x \in L(G)$$



## Reguläre Grammatiken

## Beispiel

- Für den DFA



erhalten wir die Grammatik  $G = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, P, q_0)$  mit

$$\begin{array}{ll}
 P: & q_0 \rightarrow 0q_1, 1q_0 & q_1 \rightarrow 0q_2, 1q_3 \\
 & q_2 \rightarrow 0q_2, 1q_3, \varepsilon & q_3 \rightarrow 0q_1, 1q_0, \varepsilon
 \end{array}$$

- Der akzeptierenden Rechnung

$q_0, q_1, q_2, q_3$

bei Eingabe  $x = 001$  entspricht dann folgende Ableitung:

$$\underline{q_0} \Rightarrow 0\underline{q_1} \Rightarrow 00\underline{q_2} \Rightarrow 001\underline{q_3} \Rightarrow 001$$

## Reguläre Grammatiken

- Offensichtlich lässt sich obige Konstruktion einer Grammatik  $G$  aus einem DFA  $M$  umdrehen, falls  $G$  keine Regeln der Form  $A \rightarrow a$  enthält
- Zu jeder solchen regulären Grammatik  $G$  erhalten wir dann einen äquivalenten NFA
- Für den Beweis der Rückrichtung genügt es daher, alle Regeln der Form  $A \rightarrow a$  zu eliminieren

### Lemma

Zu jeder regulären Grammatik  $G = (V, \Sigma, P, S)$  gibt es eine äquivalente reguläre Grammatik  $G'$ , die keine Regeln der Form  $A \rightarrow a$  hat

### Beweis

Betrachte die Grammatik  $G' = (V', \Sigma, P', S)$  mit

$$V' = V \cup \{X_{neu}\} \text{ und}$$

$$P' = \{A \rightarrow aX_{neu} \mid A \rightarrow_G a\} \cup \{X_{neu} \rightarrow \varepsilon\} \cup P \setminus (V \times \Sigma)$$

## Beispiel

- Betrachte die Grammatik  $G = (\{A, B, C\}, \{a, b\}, P, A)$  mit

$$P: A \rightarrow aB, bC, \varepsilon$$

$$B \rightarrow aC, bA, b$$

$$C \rightarrow aA, bB, a$$

- Wir ersetzen die Regeln  $B \rightarrow b$  und  $C \rightarrow a$  durch die Regeln  $B \rightarrow bD$  und  $C \rightarrow aD$  und fügen die Regel  $D \rightarrow \varepsilon$  hinzu
- Damit erhalten wir die Grammatik  $G' = (\{A, B, C, D\}, \{a, b\}, P', A)$  mit

$$P': A \rightarrow aB, bC, \varepsilon$$

$$B \rightarrow aC, bA, bD$$

$$C \rightarrow aA, bB, aD$$

$$D \rightarrow \varepsilon$$

Beweis von  $\{L(G) \mid G \text{ ist eine reguläre Grammatik}\} \subseteq \text{REG}$ 

- Sei  $G = (V, \Sigma, P, S)$  eine reguläre Grammatik, die keine Regeln der Form  $A \rightarrow a$  enthält
- Drehen wir obige Konstruktion einer Grammatik aus einem DFA um, so erhalten wir den NFA

$$M = (Z, \Sigma, \Delta, \{S\}, E)$$

mit  $Z = V$ ,  $\Delta(A, a) = \{B \mid A \rightarrow_G aB\}$  und  $E = \{A \mid A \rightarrow_G \varepsilon\}$

- Genau wie oben folgt dann  $L(M) = L(G)$





## Beispiel (Fortsetzung)

Die Grammatik  $G' = (\{A, B, C, D\}, \{a, b\}, P', A)$  mit

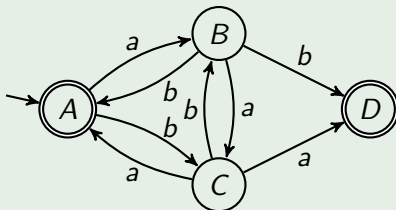
$$P' : A \rightarrow aB, bC, \varepsilon$$

$$B \rightarrow aC, bA, bD$$

$$C \rightarrow aA, bB, aD$$

$$D \rightarrow \varepsilon$$

führt auf den NFA



## Korollar

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

- $L$  ist regulär (d.h. 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  $L = L(\gamma)$
- die Nerode-Relation  $\sim_L$  von  $L$  hat einen endlichen Index
- es gibt eine reguläre Grammatik  $G$  mit  $L = L(G)$