

# Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik  
Humboldt-Universität zu Berlin

WS 2013/14

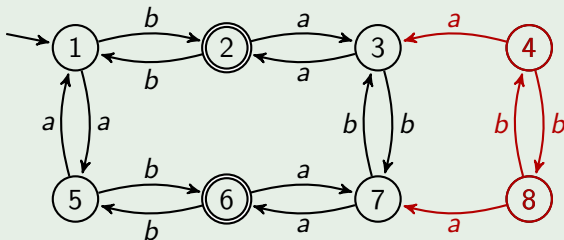
# Minimierung von DFAs

## 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 vom Startzustand aus **unerreichbaren Zustände** entfernt werden.



## 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 vom Startzustand aus 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:  $p \sim q$ ), wenn  $L(M_p) = L(M_q)$  ist.
- Offensichtlich ist  $\sim$  eine Äquivalenzrelation auf  $Z$ .

# Minimierung von DFAs

## Idee

Verschmelze jeden Zustand  $z$  mit allen äquivalenten Zuständen  $z' \sim z$  zu einem neuen Zustand.

## Notation

- Für die durch  $z$  repräsentierte Äquivalenzklasse

$$[z]_{\sim} = \{z' \in Z \mid z' \sim z\} = \{z' \in Z \mid L(M_{z'}) = L(M_z)\}$$

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

- Für eine Teilmenge  $Q \subseteq Z$  bezeichne

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

die Menge aller Äquivalenzklassen  $\tilde{q}$ , die mind. ein  $q \in Q$  enthalten.

- Dann führt obige Idee auf folgenden DFA:

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

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

Hierzu genügt es, herauszufinden, ob zwei Zustände  $p$  und  $q$  von  $M$  äquivalent sind oder nicht?

- Sei  $A\Delta B = (A \setminus B) \cup (B \setminus A)$  die **symmetrische Differenz** von  $A$  und  $B$ .
- Die Inäquivalenz  $p \not\sim q$  ist also gleichbedeutend mit  $L(M_p)\Delta L(M_q) \neq \emptyset$ .
- Wir nennen ein Wort  $x \in L(M_p)\Delta L(M_q)$  einen **Unterscheider** zwischen  $p$  und  $q$ .
- Offenbar unterscheidet  $\varepsilon$  Zustände  $p \in E$  von Zuständen  $q \in Z \setminus E$ .
- Falls  $x$  die Zustände  $\delta(p, a)$  und  $\delta(q, a)$  unterscheidet, so unterscheidet  $ax$  die Zustände  $p$  und  $q$ , d.h.  $\delta(p, a) \not\sim \delta(q, a) \Rightarrow p \not\sim q$ .
- Wenn also  $D$  nur inäquivalente Zustandspaare enthält, so trifft dies auch auf die Menge

$$D' = \{\{p, q\} \mid \exists a \in \Sigma : \{\delta(p, a), \delta(q, a)\} \in D\}$$

zu.

## 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

- Zuerst müssen wir zeigen, dass  $\delta'$  wohldefiniert ist, also  $\delta'(\tilde{q}, a)$  nicht von der Wahl des Repräsentanten  $q$  für die Äquivalenzklasse  $\tilde{q}$  abhängt.
- Hierzu ist die Implikation  $p \sim q \Rightarrow \delta(p, a) \sim \delta(q, a)$  zu zeigen.
- Diese ist wiederum äquivalent zur Kontraposition  $\delta(p, a) \not\sim \delta(q, a) \Rightarrow p \not\sim q$ , die wir bereits gezeigt haben.

# Minimierung von DFAs

## 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 (Fortsetzung)

- Als nächstes zeigen wir, dass  $L(M') = L(M)$  ist.
- Sei  $x = x_1 \dots x_n \in \Sigma^*$  eine Eingabe und seien  $q_0, q_1, \dots, q_n$  die von  $M(x)$  durchlaufenen 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.  $M'$  durchläuft 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  $\tilde{q}_n \in \tilde{E}$ , d.h. es gilt

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

## Beweis (Schluss)

- Noch z.z.: die Anzahl  $\|\tilde{Z}\|$  der Zustände von  $M'$  ist minimal.
- Wegen  $\|\tilde{Z}\| \leq \|Z\|$  ist  $M'$  sicher dann minimal, wenn  $M$  minimal ist.
- Es reicht also zu zeigen, dass  $\|\tilde{Z}\|$  nur von der Sprache  $L(M)$  abhängt.
- Wegen  $p \sim q \Leftrightarrow L(M_p) = L(M_q)$  gilt  $\|\tilde{Z}\| = \|\{L(M_z) \mid z \in Z\}\|$ .
- Sei  $L = L(M)$  und für  $x \in \Sigma^*$  sei  $L_x$  die Sprache  $\{y \in \Sigma^* \mid xy \in L\}$ .
- Dann gilt  $\{L_x \mid x \in \Sigma^*\} = \{L(M_z) \mid z \in Z\}$ :
  - ⊆: Klar, da  $L_x = L(M_z)$  für  $z = \hat{\delta}(q_0, x)$  ist.
  - ⊇: Auch klar, da jedes  $z \in Z$  über ein  $x \in \Sigma^*$  erreichbar ist.
- Also hängt  $\|\tilde{Z}\| = \|\{L_x \mid x \in \Sigma^*\}\|$  nur von  $L$  ab. □

## Bemerkung

Der Beweis zeigt, dass folgende Äquivalenzrelation  $R_L$  endlichen Index hat:

$$x R_L y \Leftrightarrow L_x = L_y.$$



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

Hierzu genügt es, zu entscheiden, ob zwei Zustände  $p$  und  $q$  von  $M$  äquivalent sind oder nicht?

- Offenbar unterscheidet  $\varepsilon$  Zustände  $p \in E$  von Zuständen  $q \in Z \setminus E$ .
- Falls  $x$  die Zustände  $\delta(p, a)$  und  $\delta(q, a)$  unterscheidet, so unterscheidet  $ax$  die Zustände  $p$  und  $q$ , d.h.  $\delta(p, a) \neq \delta(q, a) \Rightarrow p \neq q$ .
- Wenn also  $D$  nur inäquivalente Zustandspaare enthält, so trifft dies auch auf die Menge

$$D' = \{ \{p, q\} \mid \exists a \in \Sigma : \{\delta(p, a), \delta(q, a)\} \in D \}$$

zu.

## Idee

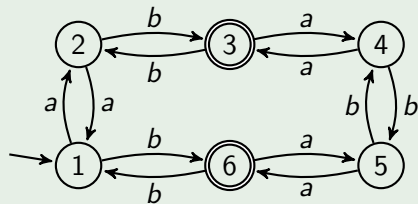
- Berechne ausgehend von  $D_0 = \{\{p, q\} \mid p \in E, q \notin E\}$  mittels
$$D_{i+1} = D_i \cup \{\{p, q\} \mid \exists a \in \Sigma : \{\delta(p, a), \delta(q, a)\} \in D_i\}$$
eine Folge  $D_0 \subseteq D_1 \subseteq D_2 \subseteq \dots$  von Mengen mit inäquivalenten Zustandspaaren.
- Da es nur endlich viele Zustandspaare gibt, muss es ein  $j$  geben mit  $D_{j+1} = D_j$ .
- Für die Menge  $D = D_j$  gilt dann
$$p \not\sim q \Leftrightarrow \{p, q\} \in D \quad (\text{siehe Übungen}).$$
- Folglich gilt
$$\tilde{Z} = \{z' \in Z \mid \{z, z'\} \notin D\}.$$

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

```
1  Input: DFA  $M = (Z, \Sigma, \delta, q_0, E)$ 
2  entferne alle nicht erreichbaren Zustände
3   $D' := \{\{z, z'\} \mid z \in E, z' \notin E\}$ 
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 für jeden Zustand
     $z \in Z$  gilt:  $\tilde{z} = \{z' \in Z \mid \{z, z'\} \notin D\}$ 
```

## Beispiel

Betrachte den DFA  $M$



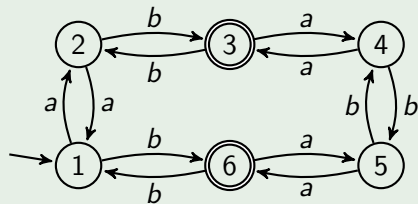
2					
3	$\epsilon$	$\epsilon$			
4			$\epsilon$		
5			$\epsilon$		
6	$\epsilon$	$\epsilon$		$\epsilon$	$\epsilon$
	1	2	3	4	5

Dann enthält  $D_0$  die Paare

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

Algorithmus für die Konstruktion von  $M'$ 

## 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

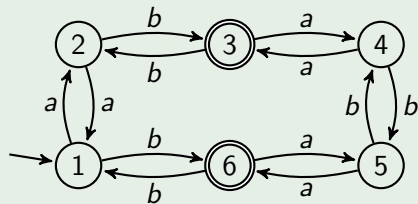
Wegen

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

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

Algorithmus für die Konstruktion von  $M'$ 

## 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

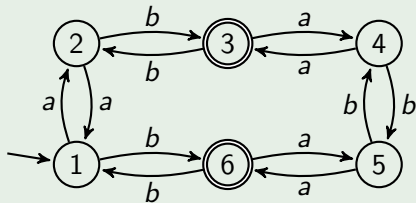
Da nun jedoch die verbliebenen Paare  $\{1, 2\}$ ,  $\{3, 6\}$ ,  $\{4, 5\}$  wegen

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

nicht zu  $D_1$  hinzugefügt werden können, ist  $D_2 = D_1 = D$ .

Algorithmus für die Konstruktion von  $M'$ 

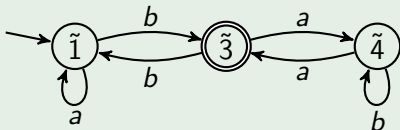
## Beispiel

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  $\tilde{4} = \{4, 5\}$ :



## Bemerkung

- $M'$  erreicht nach Lesen von  $x$  den Zustand  $\widehat{\delta}(q_0, x)$ . Wegen

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

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.

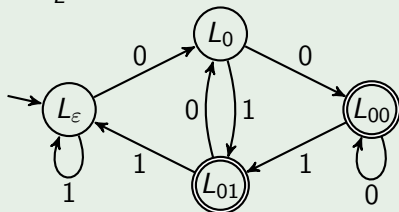


## Beispiel

- Betrachte die Sprache  $L = \{x_1 \dots x_n \in \{0, 1\}^* \mid x_{n-1} = 0\}$ .
- Dann hat  $M_L$  die folgenden Sprachen als Zustände:

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

- Graphische Darstellung von  $M_L$ :



## Der Satz von Myhill und Nerode

- Notwendig und hinreichend für die Existenz von  $M_L$  ist, dass die Menge  $Z_L = \{L_x \mid x \in \Sigma^*\}$  endlich ist.
- $L$  ist also genau dann regulär, wenn die Äquivalenzrelation  $R_L$  einen endlichen Index hat. Zur Erinnerung:

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

- Ist  $M$  ein DFA mit einer minimalen Anzahl von Zuständen, so haben die Zustände von  $M'$  die Form  $\tilde{q} = \{q\}$ , d.h.  $M$  ist isomorph zu  $M'$ .
- Da  $M'$  wiederum isomorph zu  $M_L$  ist, ist jeder minimale DFA  $M$  mit  $L(M) = L$  isomorph zu  $M_L$ , d.h. für jede reguläre Sprache  $L$  gibt es bis auf Isomorphie nur einen Minimal-DFA.

## Satz (Myhill und Nerode)

- 1  $REG = \{L \mid index(R_L) < \infty\}$ .
- 2 Sei  $L$  regulär und sei  $index(R_L)$  der Index von  $R_L$ . Dann gibt es für  $L$  bis auf Isomorphie genau einen Minimal-DFA. Dieser hat  $index(R_L)$  Zustände.

## Der Äquivalenzklassen-DFA $M_{R_L}$ für $L$

- Zwei Eingaben  $x$  und  $y$  überführen den DFA  $M_L$  genau dann in denselben Zustand, wenn  $L_x = L_y$  ist (also  $xR_L y$  gilt).
- Die Zustände von  $M_L$  können daher anstelle von  $L_x$  auch mit den Äquivalenzklassen  $[x]$  von  $R_L$  (bzw. mit geeigneten Repräsentanten) benannt werden.
- Der resultierende Minimal-DFA  $M_{R_L}$  wird auch als **Äquivalenzklassen-automat** bezeichnet:

$$M_{R_L} = (Z, \Sigma, \delta, [\varepsilon], E) \text{ mit } Z = \{[x] \mid x \in \Sigma^*\} \text{ und } E = \{[x] \mid x \in L\}.$$

- Für die Konstruktion von  $\delta$  genügt es, ausgehend von  $r_1 = \varepsilon$  eine Folge von Wörtern  $r_1, \dots, r_k$  mit  $[r_i] \neq [r_j]$  zu bestimmen, so dass zu jedem  $r_i$  und jedem Zeichen  $a \in \Sigma$  ein  $r_j$  existiert mit  $r_i a \in [r_j]$ .
- In diesem Fall ist dann  $\delta([r_i], a) = [r_i a] = [r_j]$ .
- Die Konstruktion von  $M_{R_L}$  erfordert meist weniger Aufwand als die von  $M_L$ , da die Bestimmung der Sprachen  $L_x$  entfällt.

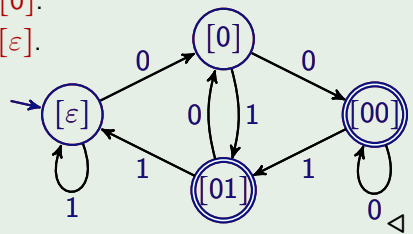
# Direkte Konstruktion von $M_{R_L}$ aus $L$

## Beispiel

Für die Sprache  $L = \{x_1 \dots x_n \in \{0, 1\}^* \mid x_{n-1} = 0\}$  lässt sich der Äquivalenzklassen-DFA  $M_{R_L}$  ausgehend von  $r_1 = \varepsilon$  wie folgt konstruieren:

- 1 Wegen  $r_1 0 = 0 \notin [\varepsilon]$  ist  $r_2 = 0$  und  $\delta([\varepsilon], 0) = [0]$ .
- 2 Wegen  $r_1 1 = 1 \in [\varepsilon]$  ist  $\delta([\varepsilon], 1) = [\varepsilon]$ .
- 3 Wegen  $r_2 0 = 00 \notin [\varepsilon] \cup [0]$  ist  $r_3 = 00$  und  $\delta([0], 0) = [00]$ .
- 4 Wegen  $r_2 1 = 01 \notin [\varepsilon] \cup [0] \cup [00]$  ist  $r_4 = 01$  und  $\delta([0], 1) = [01]$ .
- 5 Wegen  $r_3 0 = 000 \in [00]$  ist  $\delta([00], 0) = [00]$ .
- 6 Wegen  $r_3 1 = 001 \in [01]$  ist  $\delta([00], 1) = [01]$ .
- 7 Wegen  $r_4 0 = 010 \in [0]$  ist  $\delta([01], 0) = [0]$ .
- 8 Wegen  $r_4 1 = 011 \in [\varepsilon]$  ist  $\delta([01], 1) = [\varepsilon]$ .

$r$	$\varepsilon$	0	00	01
$[r0]$	$[0]$	$[00]$	$[00]$	$[0]$
$[r1]$	$[\varepsilon]$	$[01]$	$[01]$	$[\varepsilon]$



## Korollar

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

- $L$  ist regulär,
- es gibt einen DFA  $M$  mit  $L = L(M)$ ,
- es gibt einen NFA  $N$  mit  $L = L(N)$ ,
- es gibt einen regulären Ausdruck  $\gamma$  mit  $L = L(\gamma)$ ,
- die Äquivalenzrelation  $R_L$  hat endlichen Index.

Wir können also beweisen, dass eine Sprache  $L$  **nicht** regulär ist, indem wir unendlich viele Wörter finden, die paarweise inäquivalent bzgl.  $R_L$  sind.

## Satz

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

## Beweis

Die Wörter  $a^i$ ,  $i \geq 0$ , sind bzgl.  $R_L$  paarweise inäquivalent.

Für  $i \neq j$  gilt nämlich  $\neg a^i R_L a^j$ , da

$$b^i \in L_{a^i} \Delta L_{a^j}$$

enthalten ist. □

## Frage

Wie lässt sich möglichst einfach 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 **Pumping-Zahl** 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 Pumping-Zahl  $l = 3$ ).

- 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“.
- Sei

$$l = \begin{cases} 1 + \max_{x \in L} |x|, & L \neq \emptyset, \\ 0, & \text{sonst.} \end{cases}$$

- Dann lässt sich jedes Wort  $x \in L$  der Länge  $\geq l$  „pumpen“, da solche Wörter gar nicht existieren. Also hat  $L$  eine Pumping-Zahl  $\leq l$ .
- Zudem gibt es im Fall  $l > 0$  ein Wort  $x \in L$  der Länge  $l - 1$ , das sich nicht „pumpen“ lässt.
- Somit ist die Pumping-Zahl von  $L$  gleich  $l$ .

## Satz (Pumping-Lemma für reguläre Sprachen)

Zu jeder regulären Sprache  $L$  gibt es eine Zahl  $l$ , 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 **Pumping-Zahl** von  $L$  genannt.

# Das Pumping-Lemma

## Beweis

- Sei  $M = (Z, \Sigma, \delta, q_0, E)$  ein DFA 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.  $v \neq \epsilon$ ),  $k = |uv| \leq l$ .
- 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. □

# 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

Die Sprache

$$L = \{a^n b^n \mid n \geq 0\}$$

ist nicht regulär:

- 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
  - ①  $v \neq \varepsilon$
 ist die Bedingung
  - ③  $uv^i w \in L$
 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  $L$  ein Wort  $x$  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

$$\textcircled{3} \quad uv^2w \in L$$

verletzt, da  $r + 2s + t = l^2 + s$  keine Quadratzahl ist:

$$l^2 < l^2 + s \leq l^2 + l < (l+1)^2.$$

# 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:
  - ❶  $v \neq \varepsilon$ ,
  - ❷  $|uv| \leq l$  oder
  - ❸  $uv^i w \in L$  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
  - ❶  $v \neq \varepsilon$  (d.h.  $s \geq 1$ )
 ist die Bedingung
  - ❸  $uv^i w \in L$
 wegen
 
$$|uv^i w| = p + (i - 1)s$$
 für  $i = p + 1$  verletzt, da dann
 
$$|uv^i w| = p + ps = p(s + 1)$$
 ist.

## 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

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

- Da jedes Wort  $x \in L$  mit Ausnahme von  $\varepsilon$  „gepumpt“ werden kann, hat  $L$  die Pumping-Zahl 1.
- Allerdings ist  $L$  nicht regulär (siehe Übungen).



Eine elegante Methode, Sprachen zu beschreiben, sind Grammatiken. Implizit haben wir hiervon bei der Definition der regulären Ausdrücke schon Gebrauch gemacht.

### Beispiel

Die Sprache  $RA_{\Sigma}$  aller regulären Ausdrücke über einem Alphabet  $\Sigma = \{a_1, \dots, a_k\}$  lässt sich aus dem Symbol  $R$  unter Anwendung folgender Regeln erzeugen:

$$R \rightarrow \emptyset,$$

$$R \rightarrow \epsilon,$$

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

$$R \rightarrow RR,$$

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

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

## 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

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.

# 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

$$\alpha = lur \text{ und } \beta = lvr.$$

Hierfür schreiben wir auch  $lur \Rightarrow_G lvr$ .

- Eine Folge  $\sigma = (l_0, u_0, r_0), \dots, (l_m, u_m, r_m)$  von Tripeln  $(l_i, u_i, r_i)$  heißt **Ableitung von  $\beta$  aus  $\alpha$** , falls gilt:
  - $l_0 u_0 r_0 = \alpha$ ,  $l_m u_m r_m = \beta$  und
  - $l_i \underline{u_i} r_i \Rightarrow l_{i+1} u_{i+1} r_{i+1}$  für  $i = 0, \dots, m-1$ .

Die **Länge** der Ableitung  $\sigma$  ist  $m$  und wir notieren  $\sigma$  auch in der Form

$$l_0 \underline{u_0} r_0 \Rightarrow l_1 \underline{u_1} r_1 \Rightarrow \dots \Rightarrow l_{m-1} \underline{u_{m-1}} r_{m-1} \Rightarrow l_m u_m r_m.$$

- 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  $G$ .

## Zur Erinnerung:

- $\Rightarrow^*$  bezeichnet die reflexive, transitive Hülle der Relation  $\Rightarrow$ , d.h.  $\alpha \Rightarrow^* \beta$  bedeutet, dass es ein  $n \geq 0$  gibt mit  $\alpha \Rightarrow^n \beta$ .  
Hierzu sagen wir auch,  $\beta$  ist aus  $\alpha$  (in  $n$  Schritten) ableitbar.
- $\Rightarrow^n$  bezeichnet das  $n$ -fache Produkt der Relation  $\Rightarrow$ , d.h. es gilt  $\alpha \Rightarrow^n \beta$ , falls Wörter  $\alpha_0, \dots, \alpha_n$  existieren mit
  - $\alpha_0 = \alpha, \alpha_n = \beta$  und
  - $\alpha_i \Rightarrow \alpha_{i+1}$  für  $i = 0, \dots, n-1$ .

## Beispiel

- Wir betrachten nochmals die Grammatik

$$G = (\{R\}, \Sigma', P, R) \text{ mit } \Sigma' = \Sigma \cup \{\emptyset, \epsilon, (, ), |, * \}$$

für die Sprache aller regulären Ausdrücke über  $\Sigma$  mit den Regeln

$$P: R \rightarrow \emptyset, \epsilon, a, a \in \Sigma$$

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

- Der reguläre Ausdruck  $(01)^*(\epsilon|\emptyset)$  über  $\Sigma = \{0, 1\}$  lässt sich in  $G$  aus dem Startsymbol  $R$  wie folgt ableiten:

$$\begin{aligned} \underline{R} &\Rightarrow \underline{RR} \Rightarrow (\underline{R})^* R \Rightarrow (\underline{RR})^* \underline{R} \Rightarrow (\underline{RR})^* (\underline{R|R}) \\ &\Rightarrow (\underline{0R})^* (\underline{R|R}) \Rightarrow (0\underline{1})^* (\underline{R|R}) \Rightarrow (01)^* (\underline{\epsilon|R}) \Rightarrow (01)^* (\underline{\epsilon|\emptyset}) \end{aligned}$$

# Die Chomsky-Hierarchie

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:

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

- 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. Aber nur, wenn das Startsymbol  $S$  in keiner Regel rechts vorkommt.

## Beispiel

- Wir betrachten nochmals die Grammatik

$$G = (\{R\}, \Sigma \cup \{\emptyset, \epsilon, (, ), |, * \}, P, R)$$

für die Sprache aller regulären Ausdrücke über  $\Sigma$  mit den Regeln

$$P: R \rightarrow \emptyset, \epsilon, a, a \in \Sigma$$

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

- Da auf der linken Seite jeder Regel eine einzelne Variable steht, ist  $G$  kontextfrei.
- Offenbar ist  $G$  aber keine reguläre Grammatik, da zwar die  $\|\Sigma\| + 2$  Regeln  $R \rightarrow \emptyset, \epsilon, a, a \in \Sigma$ , die geforderte Form haben, nicht jedoch die drei Regeln  $R \rightarrow RR, (R|R), (R)^*$ .

- Eine Sprache heißt vom Typ  $i$  bzw. regulär, 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  $x = x_1 \dots x_n \in \Sigma^*$ :

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

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

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

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

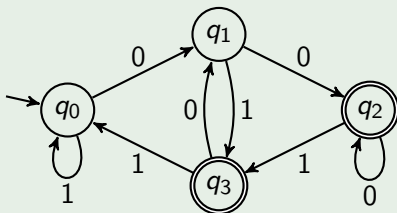
$$\Leftrightarrow \exists q_1, \dots, q_n \in V:$$

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

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

## Beispiel

Für den DFA



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

$$\begin{aligned}
 P: \quad & q_0 \rightarrow 1q_0, 0q_1, \\
 & q_1 \rightarrow 0q_2, 1q_3, \\
 & q_2 \rightarrow 0q_2, 1q_3, \varepsilon, \\
 & q_3 \rightarrow 0q_1, 1q_0, \varepsilon.
 \end{aligned}$$

- Offensichtlich lässt sich obige Konstruktion einer Grammatik  $G$  aus einem DFA  $M$  umdrehen, falls  $G$  keine Regeln der Form  $A \rightarrow a$  enthält.
- Für den Beweis der Rückrichtung genügt es daher, alle Regeln dieser Form 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
$$P: \begin{aligned} A &\rightarrow aB, bC, \varepsilon, \\ B &\rightarrow aC, bA, b, \\ C &\rightarrow aA, bB, a. \end{aligned}$$
- 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
$$P': \begin{aligned} A &\rightarrow aB, bC, \varepsilon, \\ B &\rightarrow aC, bA, bD, \\ C &\rightarrow aA, bB, aD, \\ D &\rightarrow \varepsilon. \end{aligned}$$



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

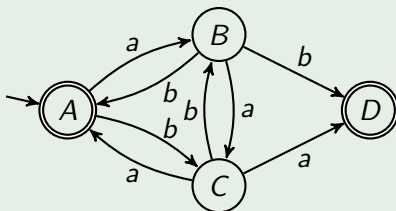
$P' : A \rightarrow aB, bC, \varepsilon,$

$B \rightarrow aC, bA, bD,$

$C \rightarrow aA, bB, aD,$

$D \rightarrow \varepsilon.$

führt auf den NFA





## Korollar

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

- $L$  ist regulär,
- 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 Äquivalenzrelation  $R_L$  hat endlichen Index,
- es gibt eine reguläre Grammatik  $G$  mit  $L = L(G)$ .