

# Automaten und Formale Sprachen

Franz Binder

Institut für Algebra  
Johannes Kepler Universität Linz

Vorlesung im 2006W

<http://www.algebra.uni-linz.ac.at/Students/Win/ml>

Endliche Automaten

Deterministische  
Automaten

Reguläre Sprachen  
und Automaten

Eigenschaften  
regulärer Sprachen

Durch eine  
Grammatik definierte  
Sprachen

Endliche Automaten

Deterministische Automaten

Reguläre Sprachen und Automaten

Eigenschaften regulärer Sprachen

Durch eine Grammatik definierte Sprachen

Endliche Automaten

Deterministische  
Automaten

Reguläre Sprachen  
und Automaten

Eigenschaften  
regulärer Sprachen

Durch eine  
Grammatik definierte  
Sprachen

$$\delta: \Sigma \rightarrow (Q \rightarrow Q \rightarrow \Omega)$$

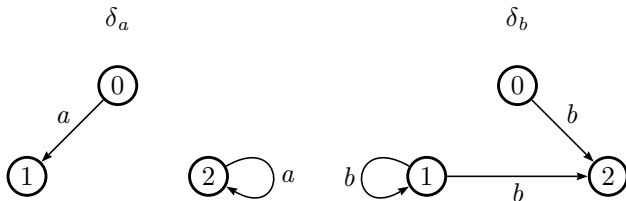
## Zustandsdiagramm

## Endliche Automaten

Deterministische  
AutomatenReguläre Sprachen  
und AutomatenEigenschaften  
regulärer SprachenDurch eine  
Grammatik definierte  
Sprachen

$$\delta: \Sigma \rightarrow (Q \rightarrow Q \rightarrow \Omega)$$

## BEISPIEL



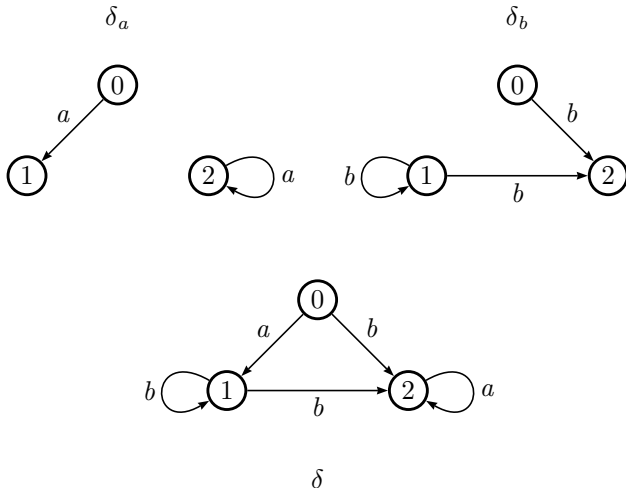
## Zustandsdiagramm

## Endliche Automaten

Deterministische  
AutomatenReguläre Sprachen  
und AutomatenEigenschaften  
regulärer SprachenDurch eine  
Grammatik definierte  
Sprachen

$$\delta: \Sigma \rightarrow (Q \rightarrow Q \rightarrow \Omega)$$

## BEISPIEL



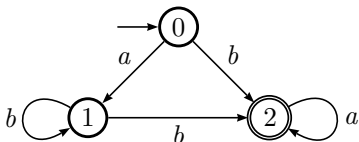
## DEFINITION

Ein **nicht-deterministischer, endlicher Automat** besteht aus

- ▶ einer endlichen Menge  $\Sigma$ , dem **Eingabealphabet**;
- ▶ einer endlichen Menge  $Q$  von **Zuständen (states)**;
- ▶ einer entscheidbaren **Zustandsüberföhrungsrelation**  
 $\delta: \Sigma \rightarrow (Q \rightarrow Q \rightarrow \mathbb{B})$ ;
- ▶ einer Teilmenge  $I \subseteq Q$  von **Initialzuständen**;
- ▶ einer Teilmenge  $F \subseteq Q$  von **Finalzuständen**.

Diese Konstruktion wird dann auch mit dem Quintupel  $(\Sigma, Q, \delta, I, F)$  bezeichnet.

## Der Automat



erkennt die Wörter:  $b$ ,  $ab$ ,  $abbbb$ ,  $baaaa$ ,  $abbbbbbaaaaa$ ;  
nicht aber:  $aa$   $abab$ ,  $bb$ .

$(Q \rightarrow Q \rightarrow \Omega)$  bildet mit dem Relationenprodukt ein Monoid.

$$\delta: \Sigma \rightarrow (Q \rightarrow Q \rightarrow \Omega)$$

wird fortgesetzt zu einem Homomorphismus:

$$\delta^*: \Sigma^* \rightarrow (Q \rightarrow Q \rightarrow \Omega)$$

## DEFINITION

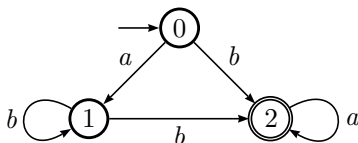
Die **Sprache** eines Automaten

$$L(\mathcal{A}) := \{ u: \Sigma^* \mid \exists_{i: I} \exists_{j: J} i \xrightarrow{\delta_u^*} j \}$$

besteht aus allen Wörtern, die einen Übergang von einem Initialzustand in einen Finalzustand erlauben.

Man sagt auch, der Automat **erkennt** oder **akzeptiert** die Sprache  $L(\mathcal{A})$ , bzw. jedes Wort in dieser Sprache.

## Der Automat



akzeptiert die Sprache, die durch den folgenden regulären Ausdruck

$$(b \mid ab^*b)a^*$$

beschrieben wird.

Dieser Automat arbeitet nicht-deterministisch.

## DEFINITION

Ein **deterministischer, endlicher Automat** besteht aus

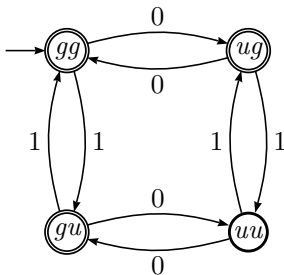
- ▶ einer endlichen Menge  $Q$  von **Zuständen**;
- ▶ einer endlichen Menge  $\Sigma$ , dem **Eingabealphabet**;
- ▶ einer **Zustandsüberföhrungsfunktion**  $\delta: \Sigma \rightarrow (Q \rightarrow Q)$ ;
- ▶ einem Initialzustand  $q_0: Q$ ;
- ▶ einer Teilmenge  $F \subseteq Q$  von **Finalzuständen**.

Diese Konstruktion wird dann auch mit dem Quintupel  $(\Sigma, Q, \delta, q_0, F)$  bezeichnet.

# Deterministischer Automat

## EXAMPLE

### Der Automat



ist deterministisch und akzeptiert die Sprache bestehend aus allen 0-1-Folgen, welche eine gerade Anzahl von Nullern oder Einsen aufweisen.

## DEFINITION

Ein **partieller** (deterministischer) Automat hat zu jedem Input höchstens einen Folgezustand.

Ein partieller Automat kann leicht vervollständigt werden, indem man einen neuen Zustand  $\omega$  (weder initial noch final), der immer dann als Folgezustand verwendet wird, wenn sonst keiner zur Verfügung steht.

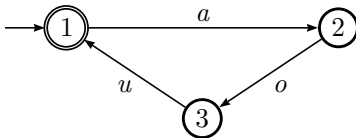
## DEFINITION

Ein **partieller** (deterministischer) Automat hat zu jedem Input höchstens einen Folgezustand.

Ein partieller Automat kann leicht vervollständigt werden, indem man einen neuen Zustand  $\omega$  (weder initial noch final), der immer dann als Folgezustand verwendet wird, wenn sonst keiner zur Verfügung steht.

## BEISPIEL

Ein partieller deterministischer Automat über dem Alphabet  $\{ a, o, u \}$ :



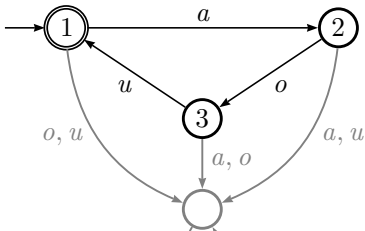
## DEFINITION

Ein **partieller** (deterministischer) Automat hat zu jedem Input höchstens einen Folgezustand.

Ein partieller Automat kann leicht vervollständigt werden, indem man einen neuen Zustand  $\omega$  (weder initial noch final), der immer dann als Folgezustand verwendet wird, wenn sonst keiner zur Verfügung steht.

## BEISPIEL

Ein partieller deterministischer Automat über dem Alphabet  $\{ a, o, u \}$ :



# Deterministisch äquivalent zu Nicht-deterministisch

## SATZ

*Zu jedem nicht-deterministischen Automaten gibt es einen deterministischen Automaten, der dieselbe Sprache erkennt.*

## BEWEIS.

Man betrachte die Teilmengen von  $Q$  als Zustände:

$$\bar{\delta}: \Sigma \rightarrow \mathcal{P}(Q) \rightarrow \mathcal{P}(Q)$$



# Deterministisch äquivalent zu Nicht-deterministisch

## SATZ

*Zu jedem nicht-deterministischen Automaten gibt es einen deterministischen Automaten, der dieselbe Sprache erkennt.*

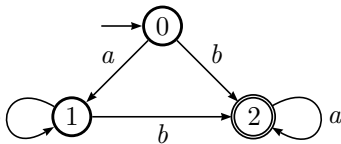
## BEWEIS.

Man betrachte die Teilmengen von  $Q$  als Zustände:

$$\bar{\delta}: \Sigma \rightarrow \mathcal{P}(Q) \rightarrow \mathcal{P}(Q)$$



## BEISPIEL



$\bar{\delta}$	$a$	$b$
0	1	2
1		12
2	2	
12	2	12

# Deterministisch äquivalent zu Nicht-deterministisch

## SATZ

*Zu jedem nicht-deterministischen Automaten gibt es einen deterministischen Automaten, der dieselbe Sprache erkennt.*

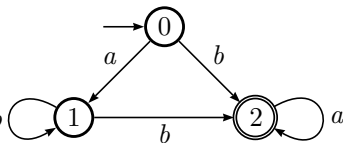
## BEWEIS.

Man betrachte die Teilmengen von  $Q$  als Zustände:

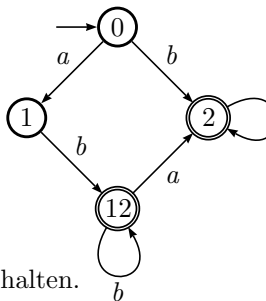
$$\bar{\delta}: \Sigma \rightarrow \mathcal{P}(Q) \rightarrow \mathcal{P}(Q)$$

□

## BEISPIEL



$\bar{\delta}$	$a$	$b$
0	1	2
1		12
2	2	
12	2	12



Initialzustand:  $I$ .

Finalzustände: Alle, die ein Element von  $J$  enthalten.

Betrachte nur erreichbare Zustandsmengen.

## DEFINITION

Seien  $\mathcal{A}_1, \mathcal{A}_2$  zwei Automaten über demselben Eingabealphabet.

**Parallelschaltung** :  $\mathcal{A}_1 + \mathcal{A}_2$

**Serienschaltung** :  $\mathcal{A}_1\mathcal{A}_2$

**Rückkoppelung** :  $\mathcal{A}^*$

## SATZ

*Seien  $\mathcal{A}_1, \mathcal{A}_2$  endliche Automaten. Dann gilt*

$$L(\mathcal{A}_1\mathcal{A}_2) = L(\mathcal{A}_1)L(\mathcal{A}_2)$$

$$L(\mathcal{A}_1 + \mathcal{A}_2) = L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$$

$$L(\mathcal{A}_1^*) = L(\mathcal{A}_1)^*.$$

## FOLGERUNG

*Zu jeder regulären Sprache gibt es einen erkennenden Automaten.*

## Vom Automaten zum regulären Ausdruck

## SATZ

*Jede von einem endlichen Automaten erkannte Sprache ist regulär.*

## BEWEIS.

$R_{i,j}^Y$  bestehe aus all jenen Wörtern, die vom Zustand  $i$  in den Zustand  $j$  führen, und dabei *zwischen* nur in Zustände aus der Menge  $Y$  führen. Dann gilt

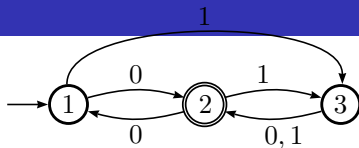
$$L(\mathcal{A}) = \bigcup_{j \in F} R_{0,j}^Q.$$

Es reicht also zu zeigen, daß jedes  $R_{i,j}^Y$  regulär ist.

Rekursion:

$$R_{i,j}^{Y \cup \{k\}} = R_{i,j}^Y \cup R_{i,k}^Y (R_{k,k}^Y)^* R_{k,j}^Y,$$

Und die  $R_{i,j}^\emptyset$  sind endliche Teilmengen von  $\Sigma$ . □



$$r_{1,2}^{\{1,2,3\}} = r_{1,2}^{\{1,2\}} \mid r_{1,3}^{\{1,2\}} (r_{3,3}^{\{1,2\}})^* r_{3,2}^{\{1,2\}}$$

$$r_{1,2}^{\{1,2\}} = r_{1,2}^{\{1\}} \mid r_{1,2}^{\{1\}} (r_{2,2}^{\{1\}})^* r_{2,2}^{\{1\}}$$

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

$$= 0(\epsilon \mid 00)^* = 0(00)^*$$

$$r_{1,3}^{\{1,2\}} = r_{1,3}^{\{1\}} \mid r_{1,2}^{\{1\}} (r_{2,2}^{\{1\}})^* r_{2,3}^{\{1\}}$$

$$= 1 \mid 0(00)^*(1 \mid 01) = 0^*1$$

$$r_{3,3}^{\{1,2\}} = r_{3,3}^{\{1\}} \mid r_{3,2}^{\{1\}} (r_{2,2}^{\{1\}})^* r_{2,3}^{\{1\}}$$

$$= \epsilon \mid (0 \mid 1)(00)^*(1 \mid 01) = \epsilon \mid (0 \mid 1)0^*1$$

$$r_{3,2}^{\{1,2\}} = r_{3,2}^{\{1\}} (r_{2,2}^{\{1\}})^*$$

$$= (0 \mid 1)(00)^*$$

$$r_{1,2}^{\{1,2,3\}} = 0(00)^* \mid 0^*1((0 \mid 1)0^*1)^*(0 \mid 1)(00)^*$$

Endliche Automaten

Deterministische  
AutomatenReguläre Sprachen  
und AutomatenEigenschaften  
regulärer SprachenDurch eine  
Grammatik definierte  
Sprachen

# Abschlußeigenschaften regulärer Sprachen

## SATZ

*Die Klasse der regulären Sprachen ist gegenüber allen booleschen Mengenoperationen (also auch Durchschnitt und Komplement)*

## BEWEIS.

Es ist nicht offensichtlich, wie ein regulärer Ausdruck für das Komplement einer regulärer Sprache ausschauen soll. Dafür kann man zu jedem Automaten  $\mathcal{A}$  ganz einfach jenen konstruieren, der alle Wörter erkennt, die  $\mathcal{A}$  nicht erkennt: man verwendet einfach  $Q \setminus F$  als Finalzustände. Für den Durchschnitt kann man dann etwa ein De Morgan Gesetz verwenden:

$$L_1 \cap L_2 = \mathcal{C}(\mathcal{C}L_1 \cup \mathcal{C}L_2).$$



Ferner ist die Klasse der regulären Sprachen gegenüber Spiegelung, homomorphen Bildern und Substitutionen abgeschlossen. Für die Spiegelung muß man lediglich den regulären Ausdruck spiegeln.

# Minimierung endlicher Automaten

## DEFINITION

Ein vollständiger deterministischer endlicher Automat  $\mathcal{A}$  heißt **minimal** wenn jeder andere Automat, der dieselbe Sprache erkennt, mindestens soviel Zustände wie  $\mathcal{A}$  hat.

## DEFINITION

**Äquivalente Zustände:**  $i \sim j \iff \forall_{u: \Sigma^*} (\delta_u^*(i) \in F \iff \delta_u^*(j) \in F)$

## SATZ

*Zu jeder regulären Sprache gibt es, bis auf Isomorphie, genau einen diese erkennenden minimalen Automaten.*

## BEWEIS.

Äquivalente Zustände lassen sich identifizieren.

Zu jedem erreichbaren Zustand  $k$  gibt es einen Inputstring  $u_k: \Sigma^*$ , sodaß  $\delta_{u_k}^*(q_0) = k$ .

Ist ein  $\mathcal{A}'$  ein weiterer Automat, der dieselbe Sprache erkennt, dann entspricht der Zustand  $\delta'_{u_k}(q'_0)$  in  $\mathcal{A}'$  dem Zustand  $k$  in  $\mathcal{A}$ .

Gilt  $\delta_u^*(q_0) = \delta_v^*(q_0)$ , dann ist  $\delta'_u(q'_0) \sim \delta'_v(q'_0)$ . □

## SATZ

*Alle folgenden Probleme sind für reguläre Sprachen entscheidbar:*

1.  $L = \emptyset$ ;
2.  $L$  ist endlich;
3.  $w \in L$ ;
4.  $L_1 = L_2$ ;
5.  $L_1 \subseteq L_2$ .

## BEWEIS.

- 3 Festzustellen, ob ein Wort zu einer Sprache gehört oder nicht, ist gerade die Aufgabe, die ein erkennender Automat (optimal) beherrscht.
- 4 Man kann von beiden einen minimalen Automaten berechnen und feststellen, ob diese isomorph sind. Alternative: Zurückführung auf Durchschnittsberechnung und Punkt 1.
- 5  $L_1 \subseteq L_2$  gilt genau dann wenn  $L_1 \cup L_2 = L_2$  (oder wenn  $L_1 \cap \complement L_2 = \emptyset$ ).

# Vervielfachen von Wortteilen

## PUMPING LEMMA

Sei  $L$  eine reguläre Sprache.

- ▶ Dann gibt es  $n \in \mathbb{N}$ , sodaß
- ▶ jedes  $z \in L$  mit  $|z| \geq n$
- ▶ derart in  $z = uvw$  zerlegt werden kann, daß gilt:
  - ▶  $|v| \geq 1$ ;
  - ▶  $|uv| \leq n$ ;
  - ▶  $uv^k w \in L$ , für alle  $k \in \mathbb{N}$ .

Jedes ausreichend lange Wort einer regulären Sprache läßt sich „aufpumpen“.

## BEWEIS.

Deterministischer endlicher Automat.

- ▶ Sei  $n = |Q|$ .
- ▶ Wenn  $|z| \geq n$ , werden mindestens  $n + 1$  Zustände durchlaufen.
- ▶ Sei  $j$  der erste wiederholte Zustand.
- ▶ Zerlege  $z = uvw$  gemäß:  
 $i \xrightarrow{u} j \xrightarrow{v} j \xrightarrow{w} f$ .

Dann gilt:

- ▶  $|v| \geq 1$ ;
- ▶  $|uv| \leq n$ ;
- ▶ Jedes  $uv^k w$  führt nach  $f$ .

## DEFINITION

Eine **Grammatik**  $\mathcal{G} = (T, N, S, \mathcal{P})$  besteht aus

- ▶ einem Alphabet  $\Sigma$  von **Terminalzeichen**;
- ▶ einem Alphabet  $N$  von **Nicht-Terminalzeichen**;
- ▶ einem Startsymbol  $S \in N$ ;
- ▶ einer Menge von Produktionsregeln der Form

$$l \Rightarrow r,$$

mit  $l \in (N \cup \Sigma)^* \setminus \Sigma^*$  und  
 $r \in (N \cup \Sigma)^*$ .

## DEFINITION

Sei  $\mathcal{G} = (\Sigma, N, S, \mathcal{P})$  eine Grammatik, und  $u, w \in (N \cup \Sigma)^*$ .

- ▶ Dann ist  $w$  aus  $u$  **ableitbar** (Schreibweise:  $u \rightarrow^* w$ ) wenn es
  - ▶ Zerlegungen  $u = ale$  und  $w = are$  gibt,
  - ▶ sodaß  $l \Rightarrow r$  eine Produktionsregel der Grammatik ist.
- ▶ Die von  $\mathcal{G}$  **erzeugte Sprache** ist definiert durch

$$L(\mathcal{G}) = \{u \in \Sigma^* \mid S \rightarrow^* u\}.$$

# Kontextfreie Grammatik

## DEFINITION

- ▶ Eine Grammatik heißt **kontextfrei** wenn jede Produktionsregel die Form

$$A \Rightarrow r$$

hat, wobei

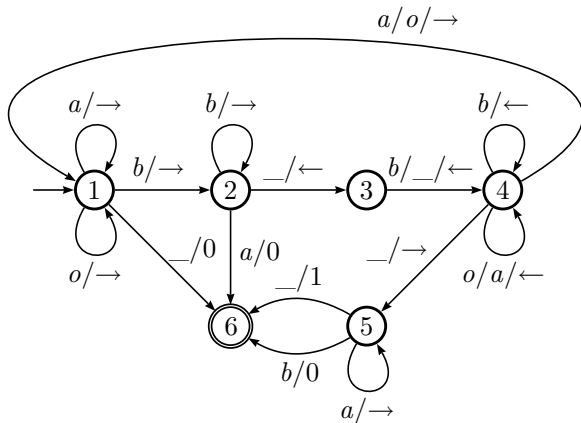
- ▶  $A$  ein Nichtterminalsymbol;
- ▶  $r$  beliebig.
- ▶ Sie heißt **rechtslinear** wenn darüberhinaus  $r$  die Form  $x$  oder  $xB$  hat, wobei  $x$  terminal und  $B$  nichtterminal ist.

## DEFINITION

Eine *Sprache* heißt **kontextfrei** bzw **rechtslinear** wenn sie durch eine kontextfreie bzw. rechtslineare Grammatik definierbar ist.

## SATZ

*Kontextfreie Sprachen werden durch Kellerautomaten erkannt.*



erkennt  $\{a^n b^{2^n} \mid n: \mathbb{N}\}$

## Endliche Automaten

Deterministische  
AutomatenReguläre Sprachen  
und AutomatenEigenschaften  
regulärer SprachenDurch eine  
Grammatik definierte  
Sprachen

Operation	reg	ktf	kts	rek	ra
Verkettung	ja	ja	ja	ja	ja
Iteration	ja	ja	ja	ja	ja
Vereinigung	ja	ja	ja	ja	ja
Durchschnitt	ja	nein	ja	ja	ja
Komplement	ja	nein	ja	ja	nein
$w \in L$	ja	ja	ja	ja	nein
$L = \emptyset$	ja	ja	nein	nein	nein
$L$ endlich	ja	ja	nein	nein	nein
$L_1 = L_2$	ja	nein	nein	nein	nein
$L_1 \subseteq L_2$	ja	nein	nein	nein	nein