

# Grundlagen der Theoretischen Informatik

## 3. Endliche Automaten (III)

2.05.2019

Viorica Sofronie-Stokkermans

e-mail: [sofronie@uni-koblenz.de](mailto:sofronie@uni-koblenz.de)

# Organisatorisches

---

- **1. Teilklausur:**

Mittwoch, 19.06.2019, 14:00-15:00

Registration opens soon.

# Übersicht

---

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational = Reguläre Ausdrücke

# Automaten mit $\varepsilon$ -Kanten

---

## Vom NDEA zum Automaten mit $\varepsilon$ -Kanten

Bisher (NDEA): Kanten mit einem **Buchstaben** beschriftet

Jetzt (**Automat mit  $\varepsilon$ -Kanten**): Kanten mit einem **Wort** beschriftet  
Es darf auch das leere Wort  $\varepsilon$  sein!

# Automaten mit $\varepsilon$ -Kanten

---

## Vom NDEA zum Automaten mit $\varepsilon$ -Kanten

Bisher (NDEA): Kanten mit einem **Buchstaben** beschriftet

Jetzt (**Automat mit  $\varepsilon$ -Kanten**): Kanten mit einem **Wort** beschriftet  
Es darf auch das leere Wort  $\varepsilon$  sein!

## Ein Automat mit $\varepsilon$ -Kanten kann ...

- in einem Schritt ein ganzes Wort verarbeiten
- einen Zustandsübergang machen, ohne dabei einen Buchstaben zu lesen

# Automaten mit $\varepsilon$ -Kanten: Definition

---

**Definiton.** Ein **Automat mit  $\varepsilon$ -Kanten ( $\varepsilon$ -NDEA)**  $A$  ist ein Tupel

$$\mathcal{A} = (K, \Sigma, \Delta, I, F)$$

Dabei ist

- $K$  eine endliche Menge von Zuständen,
- $\Sigma$  ein endliches Alphabet,
- $\Delta \subseteq (K \times \Sigma^*) \times K$  eine (endliche) Übergangsrelation,
- $I \subseteq K$  eine Menge von Startzuständen,
- $F \subseteq K$  eine Menge von finalen Zuständen

# Automaten mit $\varepsilon$ -Kanten: Übergangsrelation

**Definition. (Erweiterung von  $\Delta$  zu  $\Delta^*$ )**

$$\Delta^* \subseteq (K \times \Sigma^*) \times K$$

ist (ähnlich wie für NDEAs) definiert durch:

$$\Delta^*((q, \varepsilon), q') \quad \underline{\text{gdw}} \quad q' = q \quad \text{oder} \quad \Delta((q, \varepsilon), q')$$

$$\begin{aligned} \Delta^*((q, w_1 w_2), q') \quad \underline{\text{gdw}} \quad & \exists q'' \in K \\ & \Delta^*((q, w_1), q'') \vee \Delta((q, w_1), q'') \\ & \wedge \\ & \Delta^*((q'', w_2), q') \vee \Delta((q'', w_2), q') \end{aligned}$$

# Automaten mit $\varepsilon$ -Kanten: Akzeptierte Sprache

---

**Definition.** Die von einem Automaten mit  $\varepsilon$ -Kanten

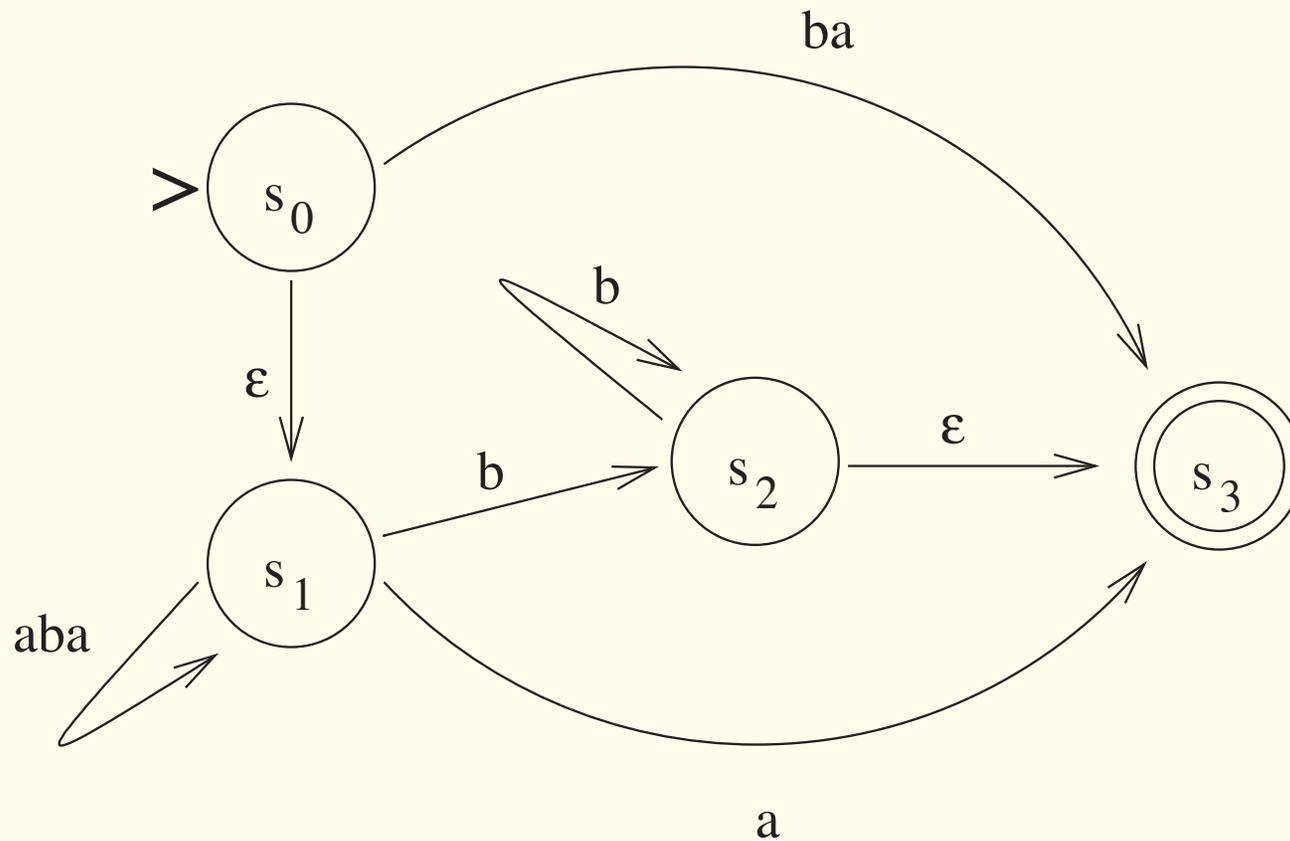
$$\mathcal{A} = (K, \Sigma, \Delta, I, F)$$

akzeptierte Sprache ist

$$L(\mathcal{A}) := \{ w \in \Sigma^* \mid \exists s_0 \in I \exists q \in F \Delta^*( (s_0, w) q ) \}$$

# Automaten mit $\varepsilon$ -Kanten: Beispiel

## Beispiel



Akzeptiert:  $\{aba\}^* \{b\} \{b\}^* + \{aba\}^* \{a\} + \{ba\}$

# Gleichmächtigkeit von Automaten mit $\varepsilon$ -Kanten und NDEAs

---

## Theorem ( $\varepsilon$ -NDEA gleich mächtig wie NDEA)

Zu jedem Automaten mit  $\varepsilon$ -Kanten  $\mathcal{A}$  existiert ein indeterminierter endlicher Automat  $\mathcal{A}'$  mit

$$L(\mathcal{A}) = L(\mathcal{A}')$$

# Gleichmächtigkeit: Beweis

---

Transformation von  $\mathcal{A}$  in einen NDEA ohne  $\varepsilon$ -Kanten

## 1. Ersetze Übergänge:

- mit nur einem Buchstaben markiert  $\rightarrow$  **beibehalten**
- mit einem Wort  $w$  markiert ( $|w| = n$ )  $\rightarrow$  **ersetze durch  $n$  Übergänge**  
(verwende  $n - 1$  neue, zusätzlicher Zustände)
- $\varepsilon$ -Übergänge  $\rightarrow$  statt diesen

$$\Delta( (q, a), q'' )$$

für jedes Paar

$$\Delta( (q, a), q' ) \quad \text{und} \quad \Delta^*( (q', \varepsilon), q'' )$$

# Gleichmächtigkeit von Automaten mit $\varepsilon$ -Kanten und NDEAs

---

Beweis (Fortsetzung)

Transformation von  $\mathcal{A}$  in einen NDEA ohne  $\varepsilon$ -Kanten

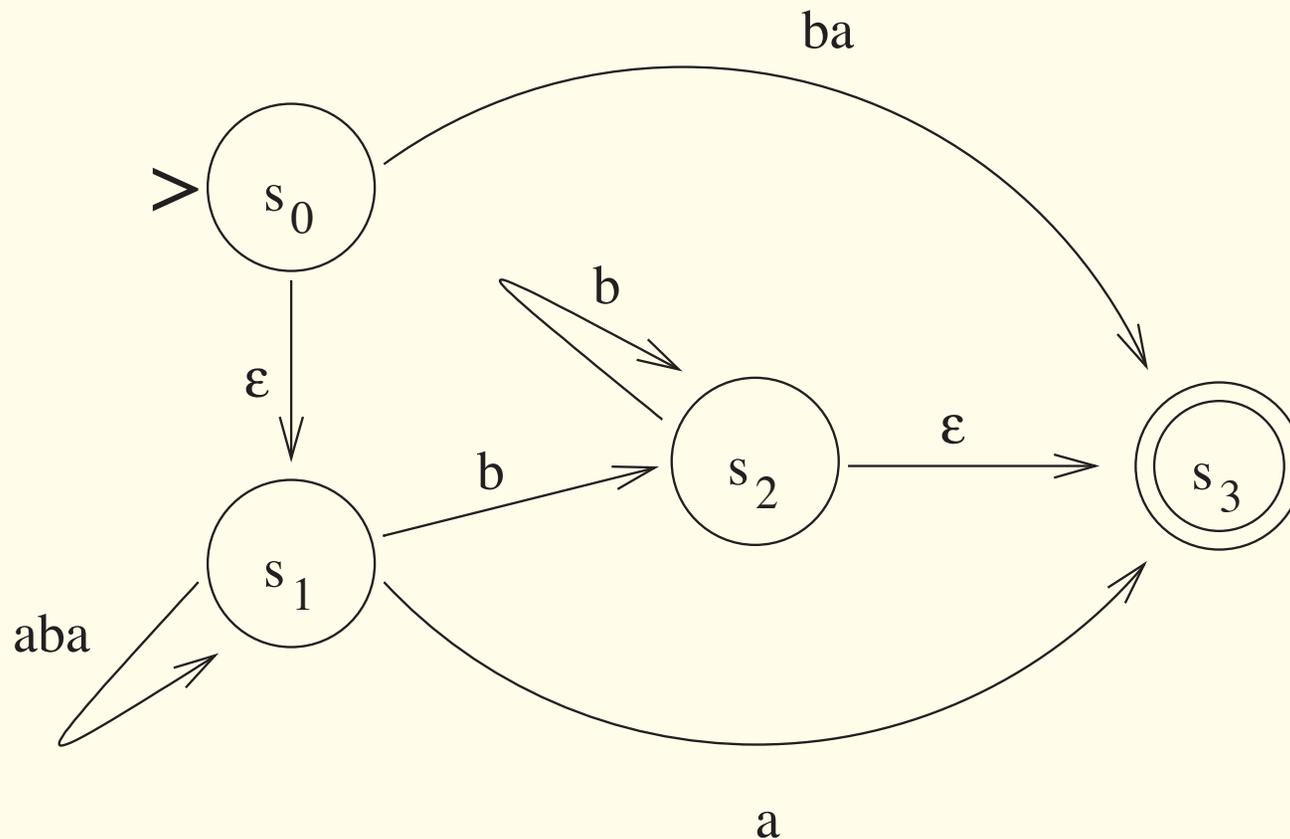
## 2. Zusätzliche Initialzustände:

Falls  $q \in I$  und  $\Delta^*((q, \varepsilon), q')$ , dann auch  $q' \in I$

## 3. Finalzustände bleiben unverändert

# Gleichmächtigkeit: Beispiel

Beispiel. Der Automat mit  $\varepsilon$ -Kanten ...



Akzeptiert:  $\{aba\}^* \{b\} \{b\}^* + \{aba\}^* \{a\} + \{ba\}$

# Gleichmächtigkeit: Beispiel

---

## 1. Ersetze Übergänge:

- Die mit einem Buchstaben markierten Übergänge: beibehalten.  
 $s_1 \xrightarrow{b} s_2$ ,  $s_2 \xrightarrow{b} s_2$  und  $s_1 \xrightarrow{a} s_3$  bleiben.
- Übergang  $s_1 \xrightarrow{aba} s_1$ :  
Neue Zustände:  $p_{(aba,1)}$ ,  $p_{(aba,2)}$ .  
Übergang  $s_1 \xrightarrow{aba} s_1$  ersetzt durch  
 $s_1 \xrightarrow{a} p_{(aba,1)} \xrightarrow{b} p_{(aba,2)} \xrightarrow{a} s_1$ .
- Übergang  $s_0 \xrightarrow{ba} s_3$ :  
Neuer Zustand:  $p_{(ba,1)}$ .  
Übergang  $s_0 \xrightarrow{ba} s_3$  ersetzt durch  $s_0 \xrightarrow{b} p_{(ba,1)} \xrightarrow{a} s_3$ .

# Gleichmächtigkeit: Beispiel

---

## 1. Ersetze Übergänge:

(Fortsetzung)

Statt  $\varepsilon$ -Übergänge:

- $\Delta((s_1, b), s_2)$  und  $\Delta((s_2, \varepsilon), s_3)$  : neuer Übergang:  $\Delta((s_1, b), s_3)$ .
- $\Delta((s_2, b), s_2)$  und  $\Delta((s_2, \varepsilon), s_3)$  : neuer Übergang:  $\Delta((s_2, b), s_3)$ .

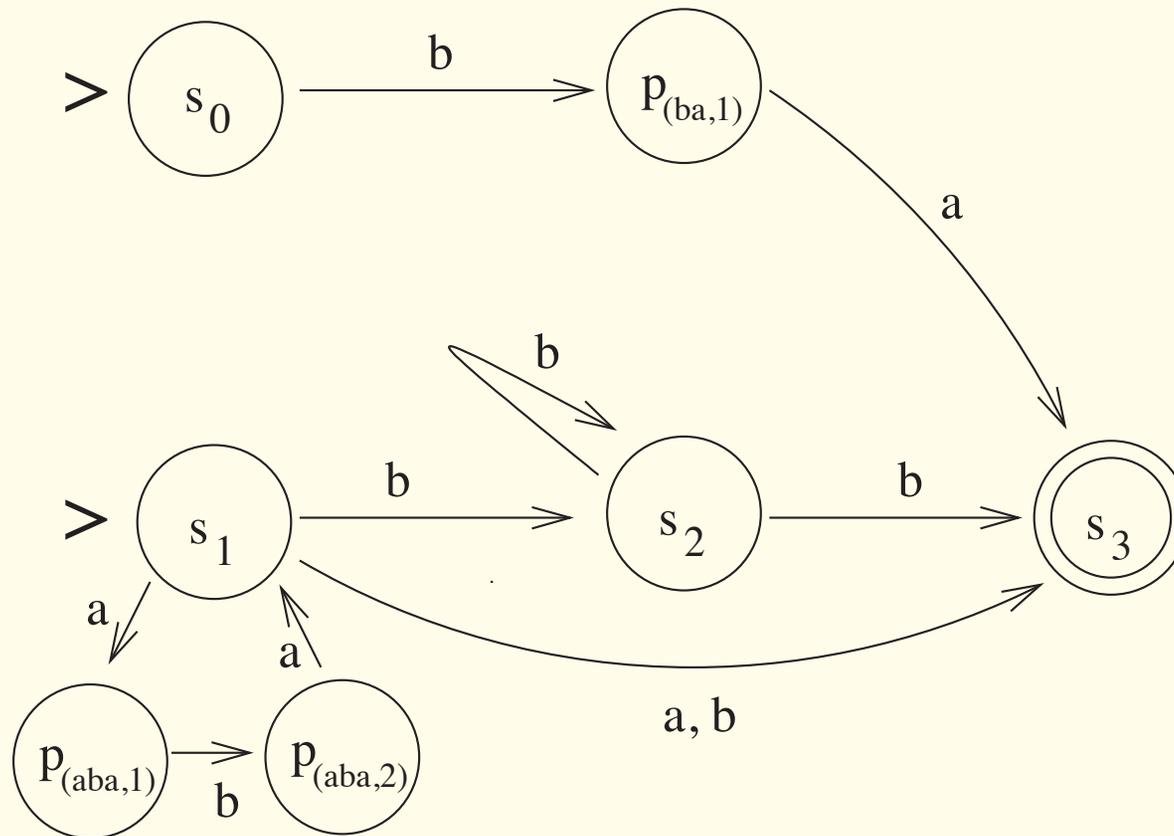
## 2. Zusätzliche Initialzustände:

Da  $s_0 \in I$  und  $\Delta((s_0, \varepsilon), s_1)$ , dann auch  $s_1 \in I$ .

## 3. Finalzustände bleiben unverändert

# Gleichmächtigkeit: Beispiel

Beispiel. ... wird transformiert in den äquivalenten NDEA



# Gleichmächtigkeit von Automaten mit $\varepsilon$ -Kanten und NDEAs

---

## Theorem ( $\varepsilon$ -NDEA gleich mächtig wie NDEA)

Zu jedem Automaten mit  $\varepsilon$ -Kanten  $\mathcal{A}$  existiert ein indeterminierter endlicher Automat  $\mathcal{A}'$  mit

$$L(\mathcal{A}) = L(\mathcal{A}')$$

# Übersicht

---

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational = Reguläre Ausdrücke

# Jetzt

---

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational = Reguläre Ausdrücke

# Satz von Kleene

---

**Theorem (Satz von Kleene:  $RAT = \mathcal{L}_3$ )**

Eine Sprache  $L$  ist rational gdw  $L$  ist regulär.

Merke:

**$L$  ist rational heißt:** es gibt einen endlichen Automaten, der  $L$  akzeptiert

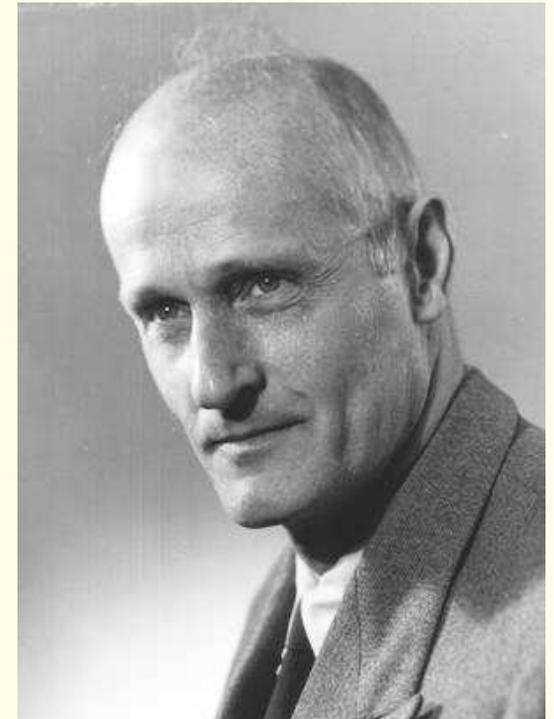
**$L$  ist regulär heißt:** es gibt eine rechtslineare Grammatik für  $L$

# Satz von Kleene

---

Stephen Cole Kleene (1909 – 1994)

- Mathematiker und Logiker
- Promovierte bei Church in Princeton (wie Turing und viele andere)
- Professor an der Universität von Wisconsin
- Einer der Begründer der theoretischen Informatik
- Unter anderem:  
Erfinder der regulären Ausdrücke



# Satz von Kleene

---

**Theorem (Satz von Kleene:  $RAT = \mathcal{L}_3$ )**

Eine Sprache  $L$  ist rational gdw  $L$  ist regulär.

Beweis:

„ $\Rightarrow$ “ zu zeigen:

**Wenn eine Sprache  $L$  von einem endlichen Automaten  $\mathcal{A}$  akzeptiert wird, ist sie regulär (wird von einer rechtslinearen Grammatik akzeptiert).**

Sei also  $L = L(\mathcal{A})$  für einen endlichen Automaten  $\mathcal{A} = (K, \Sigma, \delta, s_0, F)$

Dazu konstruieren wir eine Grammatik  $G = (V, T, R, S)$ :

**Automat  $\mathcal{A}$ :** in **Zustand  $q$** , **liest  $a$** , geht in **Zustand  $q'$**

**Grammatik:** **Variable  $q$** , **erzeugt  $a$**  neue **Variable  $q'$**

# Satz von Kleene

---

**Beweis** (Fortsetzung) Formale Definition der Grammatik:

$$V := K$$

$$T := \Sigma$$

$$S := s_0$$

$$R := \{ q \rightarrow aq' \mid \delta(q, a) = q' \} \cup \\ \{ q \rightarrow \varepsilon \mid q \in F \}$$

**Durch Induktion über die Länge eines Wortes  $w$ :**

$$S \xRightarrow*_G wq \quad \underline{\text{gdw}} \quad \delta^*(s_0, w) = q$$

$$\text{Daraus: } S \xRightarrow*_G w \quad \underline{\text{gdw}} \quad \exists q \in F (S \xRightarrow*_G wq \xRightarrow w)$$

$$\underline{\text{gdw}} \quad \exists q \in F (\delta^*(s_0, w) = q)$$

$$\underline{\text{gdw}} \quad w \in L(\mathcal{A})$$

# Satz von Kleene

---

Beweis (Fortsetzung) “ $\Leftarrow$ ” zu zeigen:

**Wenn eine Sprache  $L$  regulär ist**

**(sie wird von einer rechtslinearen Grammatik akzeptiert),**

**dann gibt es einen endlichen Automaten  $\mathcal{A}$ , der sie akzeptiert.**

Sei also  $L = L(G)$  für eine rechtslineare Grammatik  $G = (V, T, R, S)$

Dazu konstruieren wir einen  $\varepsilon$ -NDEA  $\mathcal{A} = (K, \Sigma, \Delta, I, F)$  mit:

$$K := V \cup \{q_{stop}\} \quad (q_{stop} \text{ neu})$$

$$I := \{S\}$$

$$\Sigma := T$$

$$F := \{q_{stop}\}$$

# Satz von Kleene

---

Beweis (Fortsetzung)

Definition von  $\Delta$ :

$$\begin{aligned}\Delta( (X, u), X' ) & :\iff_{def} X \rightarrow uX' \in R \\ \Delta( (X, u), q_{stop} ) & :\iff_{def} X \rightarrow u \in R\end{aligned}$$

für  $X, X' \in K$  und  $u \in \Sigma^*$

Durch Induktion über die Länge einer Ableitung:

$$S \xRightarrow*_G w \quad \underline{\text{gdw}} \quad \Delta^*( (S, w), q_{stop} ) \quad \underline{\text{gdw}} \quad w \in L(\mathcal{A})$$

Wegen Gleichmächtigkeit von  $\epsilon$ -NDEA- mit DEA-Automaten gibt es dann auch einen determinierten endlichen Automaten, der  $L$  akzeptiert.

□

# Satz von Kleene

## Beispiel:

Grammatik  $G$  mit Regeln

$S \rightarrow abaS$

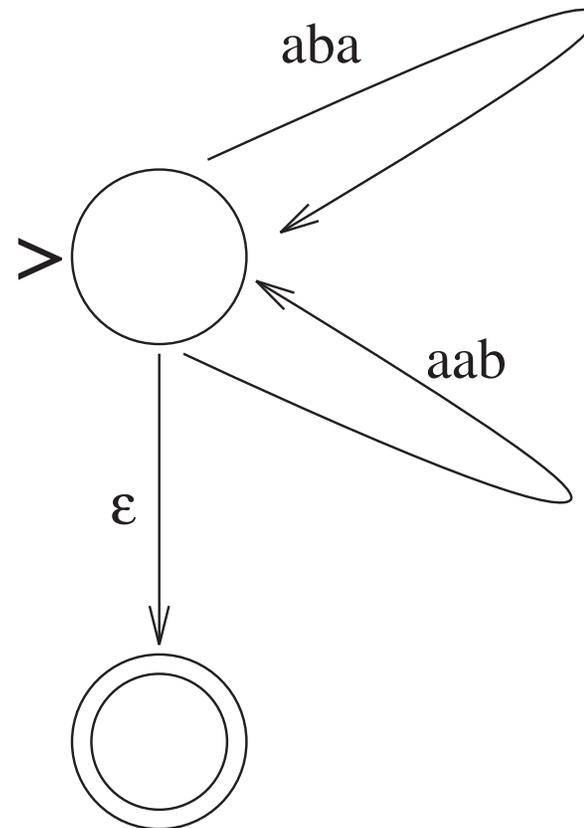
$S \rightarrow aabS$

$S \rightarrow \varepsilon$

Sprache

$$L(G) = \{aba, aab\}^*$$

$\varepsilon$ -NDEA:



# Überblick

---

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping-Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational = Reguläre Ausdrücke

# Pumping-Lemma

---

## “Aufpumpbarkeit” (informell)

Lange Wörter  $x \in L$  lassen sich zerlegen

$$x = uvw \quad |v| \geq 1$$

so dass

$$u \underbrace{vv \dots v}_i w = uv^m w$$

wieder in  $L$  liegt (für alle  $m \geq 1$ )

# Pumping-Lemma

---

## Pumping-Lemma (informell)

Zu jeder regulären Sprache  $L$  gibt es ein  $n \in \mathbb{N}$ , so dass alle Wörter

$$w \in L \quad \text{mit } |w| \geq n$$

aufgepumpt werden können

# Pumping-Lemma

---

## Pumping-Lemma (informell)

Zu jeder regulären Sprache  $L$  gibt es ein  $n \in \mathbb{N}$ , so dass alle Wörter

$$w \in L \quad \text{mit } |w| \geq n$$

aufgepumpt werden können

## Anwendung

- Wichtige Information über die Struktur regulärer Sprachen
- Nachweis der Nicht-Regularität von Sprachen:  
Wenn das Pumping-Lemma für eine Sprache nicht gilt,  
dann kann sie nicht regulär sein

# Pumping-Lemma: Intuition

---

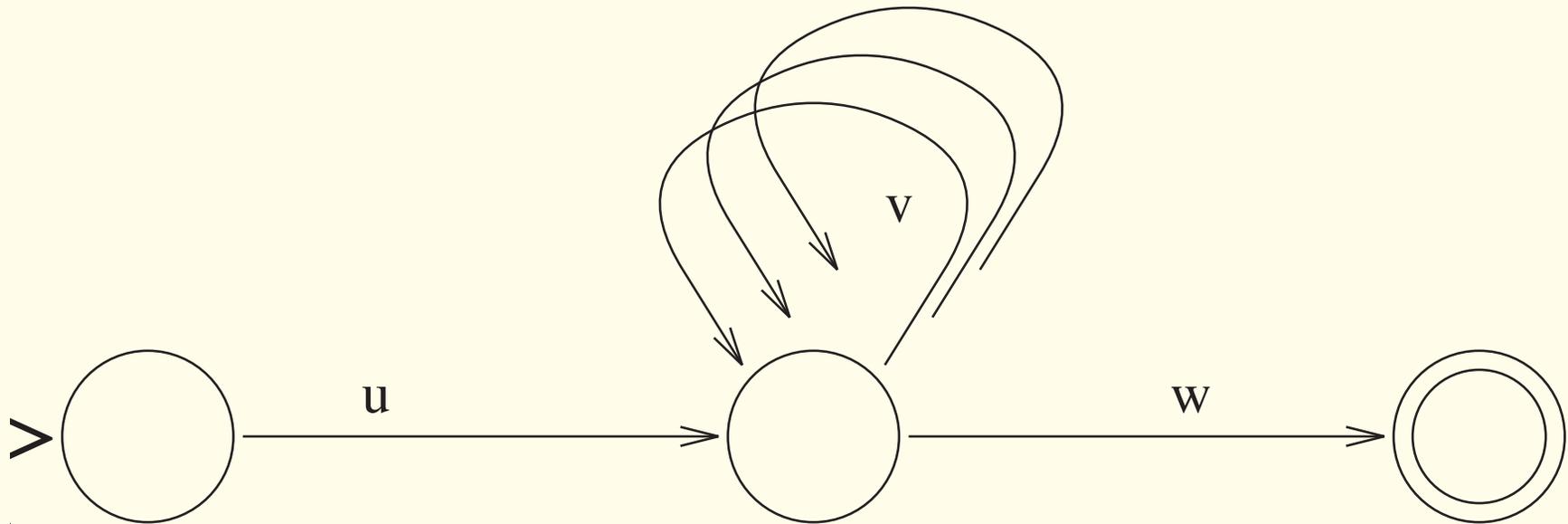
## Warum gilt das Pumping-Lemma?

- Zu regulärer Sprache  $L$  gibt es einen DEA, der  $L$  akzeptiert
- Dieser hat endliche Zustandsmenge  $K$ .  
Sei  $m := |K|$ .
- Wenn  $|w| > |K|$ , dann muss beim Akzeptieren von  $w$  eine Schleife durchlaufen werden.
- Die Schleife kann auch mehrfach durchlaufen werden.
- Das Teilwort  $v$ , das der Schleife entspricht, kann aufgepumpt werden.

# Pumping-Lemma: Intuition

---

Abstrakt gesehen



# Pumping-Lemma: Formal

---

## Theorem (Pumping-Lemma für $\mathcal{L}_3$ -Sprachen)

Sei  $L \in \mathbf{RAT}$ . Dann existiert ein  $n \in \mathbb{N}$ , so dass:

Für alle

$$x \in L \quad \text{mit} \quad |x| \geq n$$

existiert eine Zerlegung

$$x = uvw \quad u, v, w \in \Sigma^*$$

mit

- $|v| \geq 1$
- $|v| < n$
- $uv^m w \in L$  für alle  $m \in \mathbb{N}$

# Pumping-Lemma: Beweis

---

Beweis: folgt nach Beispielen

# Pumping-Lemma: Umkehrung

---

## Korollar.

Wenn für eine Sprache das Pumping-Lemma **nicht** gilt, dann ist sie **nicht** regulär.

## Vorsicht

- Es gibt nicht-reguläre Sprachen, für die das Pumping-Lemma gilt.  
(Beispiel:  $L = \{a^m b^n c^n \mid m, n \geq 1\} \cup \{b^m c^n \mid m, n \geq 0\}$ )
- Daraus, dass das Pumping-Lemma für eine Sprache gilt, folgt **nicht**, dass sie regulär ist.

# Pumping-Lemma: Anwendung der Umkehrung

---

## Beispiel

Folgende Sprachen sind **nicht regulär**:

1.  $L_1 := \{a^i b a^i \mid i \in \mathbb{N}\}$

2.  $L_2 := \{a^p \mid p \text{ ist Primzahl}\}$

# Pumping-Lemma: Anwendung der Umkehrung

---

Beweis der Nichtregularität von  $L_1$

Zu

$$L_1 := \{a^i ba^i \mid i \in \mathbb{N}\}$$

**Annahme:**  $L_1$  ist regulär.

Dann gilt für  $L_1$  das Pumping-Lemma.

Sei  $n$  die Zahl aus dem Pumping-Lemma.

Dann muss sich das Wort

$$a^n ba^n \in L_1$$

aufpumpen lassen (da  $|a^n ba^n| \geq n$ ).

Sei  $a^n ba^n = uvw$  eine passende Zerlegung laut Lemma.

# Pumping-Lemma: Anwendung der Umkehrung

---

Beweis der Nichtregularität von  $L_1$  (Forts.)

**1. Fall:**  $u = a^k, v = a^j, w = a^i ba^n$  mit  $i, k \geq 0, j > 0$  und  $k + j + i = n$ .

Einmal aufpumpen ( $m = 2$ ) ergibt:

$$uv^2w = a^k a^{2j} a^i ba^n = a^{k+2j+i} ba^n = a^{n+j} ba^n \notin L_1$$

**Widerspruch zum Lemma!**

# Pumping-Lemma: Anwendung der Umkehrung

---

Beweis der Nichtregularität von  $L_1$  (Forts.)

**1. Fall:**  $u = a^k, v = a^j, w = a^i ba^n$  mit  $i, k \geq 0, j > 0$  und  $k + j + i = n$ .

Einmal aufpumpen ( $m = 2$ ) ergibt:

$$uv^2w = a^k a^{2j} a^i ba^n = a^{k+2j+i} ba^n = a^{n+j} ba^n \notin L_1$$

**Widerspruch zum Lemma!**

**2. Fall:**  $u = a^n ba^i, v = a^j, w = a^k$

**Widerspruch zum Lemma!** (analog zu Fall 1)

# Pumping-Lemma: Anwendung der Umkehrung

---

Beweis der Nichtregularität von  $L_1$  (Forts.)

**1. Fall:**  $u = a^k, v = a^j, w = a^i ba^n$  mit  $i, k \geq 0, j > 0$  und  $k + j + i = n$ .

Einmal aufpumpen ( $m = 2$ ) ergibt:

$$uv^2w = a^k a^{2j} a^i ba^n = a^{k+2j+i} ba^n = a^{n+j} ba^n \notin L_1$$

**Widerspruch zum Lemma!**

**2. Fall:**  $u = a^n ba^i, v = a^j, w = a^k$

**Widerspruch zum Lemma!** (analog zu Fall 1)

**3. Fall:**  $u = a^k, v = a^j ba^i, w = a^l$  mit  $k + j = i + l = n$  und  $i, j, k, l \geq 0$

Einmal aufpumpen ( $m = 2$ ) ergibt:

$$uv^2w = a^k a^j ba^i a^j ba^i a^l = a^{k+j} ba^{i+j} ba^{i+l} \notin L_1$$

**Widerspruch zum Lemma!**

# Pumping-Lemma: Anwendung der Umkehrung

---

Beweis der Nichtregularität von  $L_1$  (Forts.)

**Also:** Annahme falsch.

**Also:**  $L_1$  nicht regulär.  $\square$