Grundlagen der Theoretischen Informatik

3. Endliche Automaten (III)

5.05.2021

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Bis 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

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:

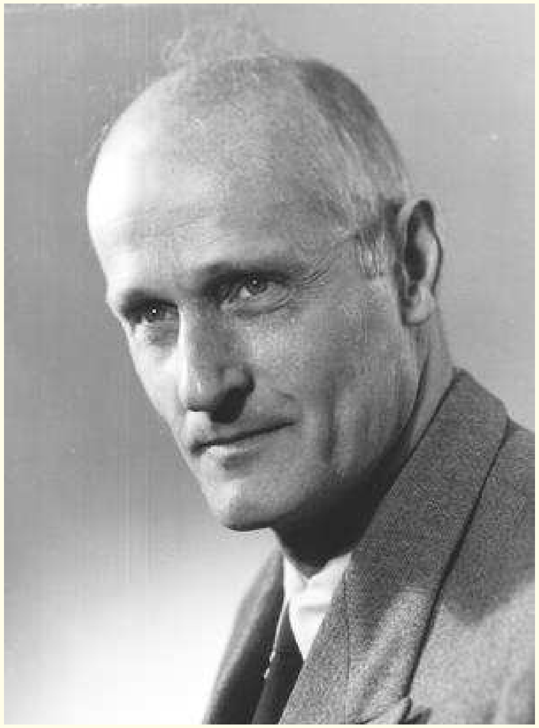
L **ist rational heißt:** es gibt einen endlichen Automaten, der L akzeptiert

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 <u>gdw</u> 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:

$$\begin{aligned}
V &:= K \\
T &:= \Sigma \\
S &:= s_0 \\
R &:= \{\, q \to aq' \mid \delta(q,a) = q' \,\} \cup \\
&\ \{\, q \to \varepsilon \mid q \in F \,\}
\end{aligned}$$

Satz von Kleene

Beweis (Fortsetzung) Formale Definition der Grammatik:

$$\begin{aligned} V &:= K \\ T &:= \Sigma \\ S &:= s_0 \\ R &:= \{\, q \to aq' \mid \delta(q,a) = q' \,\} \cup \\ & \{\, q \to \varepsilon \mid q \in F \,\} \end{aligned}$$

Durch Induktion über die Länge eines Wortes w:

$$S \Longrightarrow_G^* wq \qquad \underline{\text{gdw}} \qquad \delta^*(s_0, w) = q$$

Daraus: $S \Longrightarrow_G^* w$ $\underline{\text{gdw}}$ $\exists q \in F \; \big(S \Longrightarrow_G^* wq \Longrightarrow w\big)$

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

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

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 ε-NDEA $\mathcal{A} = (K, \Sigma, \Delta, I, F)$ mit:

$$\begin{array}{rcll} K & := & V \cup \{q_{stop}\} & \quad (q_{stop} \text{ neu}) \\ I & := & \{S\} & \\ \Sigma & := & T & \\ F & := & \{q_{stop}\} & \end{array}$$

Satz von Kleene

Beweis (Fortsetzung)

Definition von Δ:

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

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

Satz von Kleene

Beweis (Fortsetzung)

Definition von Δ:

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

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

Grammatik G mit Regeln

$$\begin{aligned} S \quad &\to \quad abaS \\ S \quad &\to \quad aabS \\ S \quad &\to \quad \varepsilon \end{aligned}$$

Satz von Kleene

Beweis (Fortsetzung)

Definition von Δ:

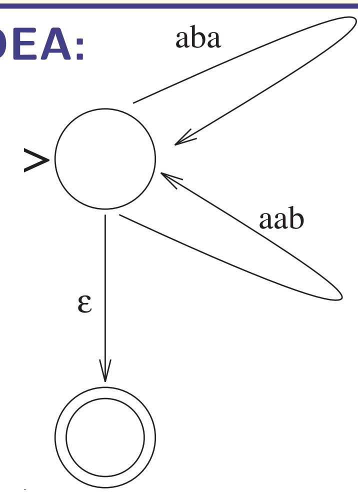
$$\begin{aligned} \Delta(\ (X,u),\ X'\ ) \quad &:\Longleftrightarrow_{def} \quad X \to uX' \in R \\ \Delta(\ (X,u),\ q_{stop}\ ) \quad &:\Longleftrightarrow_{def} \quad X \to u \in R \end{aligned}$$

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

Grammatik G mit Regeln

$$\begin{aligned} S &\to abaS \\ S &\to aabS \\ S &\to \varepsilon \end{aligned}$$

ε-NDEA:



Satz von Kleene

Beweis (Fortsetzung)

Definition von Δ:

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

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

Durch Induktion über die Länge einer Ableitung:

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

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

$\square$

Satz von Kleene

Beispiel:

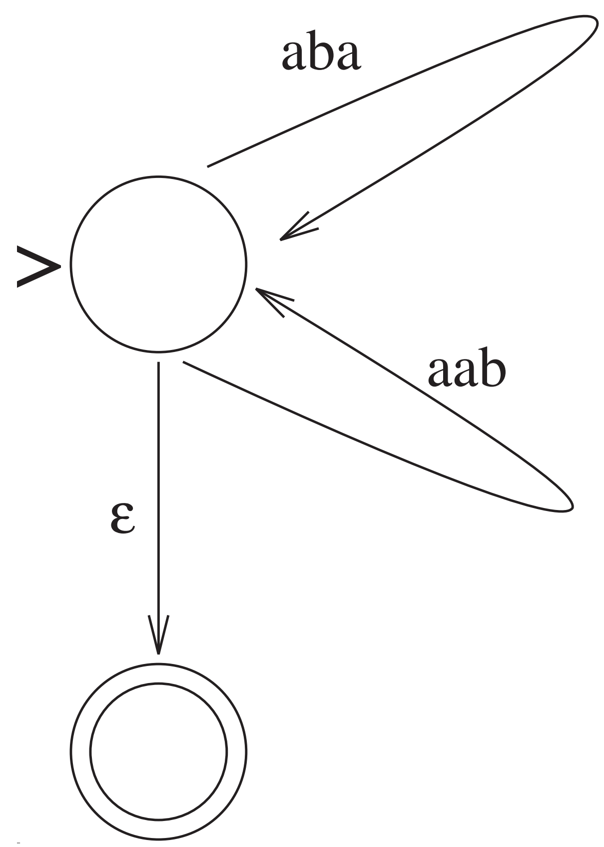
Grammatik G mit Regeln

$$
\begin{array}{lcl}
S & \rightarrow & abaS \\
S & \rightarrow & aabS \\
S & \rightarrow & \varepsilon
\end{array}
$$

Sprache

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

ε-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

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

so dass

$$u \underbrace{vv \ldots v}_{i \text{ mal}} w = uv^m w$$

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

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

aufgepumpt werden können

Pumping-Lemma

Pumping-Lemma (informell)

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

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

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

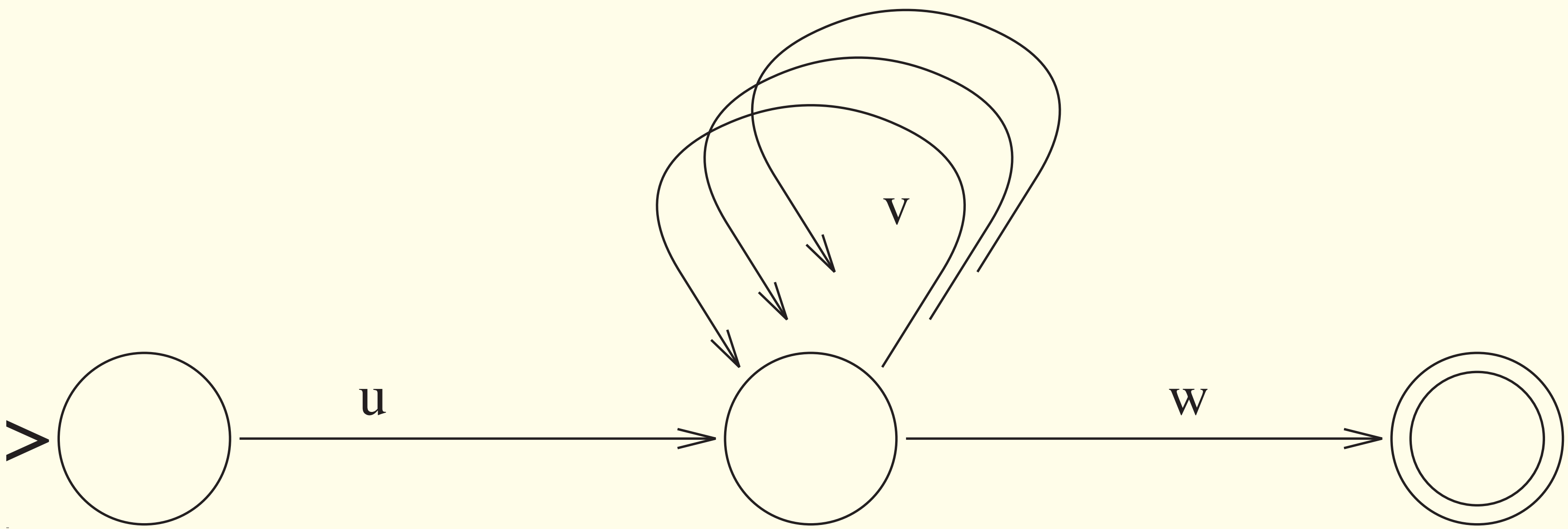
Pumping-Lemma: Intuition

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 $L \in$ **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

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

mit

- $|v| \geq 1$
- $|v| < n$
- $uv^m w \in L \quad$ 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.) $\underbrace{aa\ldots aa}_{n}\,b\,\underbrace{aa\ldots aa}_{n} = uvw$

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.) $\underbrace{aa\ldots aa}_{n}\, b\, \underbrace{aa\ldots aa}_{n} = uvw$

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.) $\underbrace{aa\ldots aa}_{n}\,b\,\underbrace{aa\ldots aa}_{n} = uvw$

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.) $\underbrace{aa\dots aa}_{n}\, b\, \underbrace{aa\dots aa}_{n} = uvw$

Also: Annahme falsch.

Also: L_1 nicht regulär. $\square$

Pumping-Lemma: Anwendung der Umkehrung

Beweis der Nichtregularität von $L_2 := \{a^p \mid p$ **ist Primzahl** $\}$

Pumping-Lemma: Anwendung der Umkehrung

Beweis der Nichtregularität von L_2

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

Annahme: L_2 ist regulär.

Dann gilt für L_2 das Pumping-Lemma.

Sei n die Zahl aus dem Pumping-Lemma.

Dann muss sich jedes Wort

$$a^p \in L_2 \quad \text{mit} \quad p \geq n$$

aufpumpen lassen.

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

Pumping-Lemma: Anwendung der Umkehrung

Beweis der Nichtregularität von L_2 (Forts.)

Sei

$$a^p = uvw = a^i a^j a^k$$

also

$$i + j + k = p \geq n \qquad \text{und} \qquad 0 < j < n$$

Pumping-Lemma: Anwendung der Umkehrung

Beweis der Nichtregularität von L_2 (Forts.)

Fall 1: $i + k > 1$.

Pumpe $(i + k)$ mal:

$$uv^{i+k}w = a^i a^{j(i+k)} a^k$$

Nach Pumping-Lemma liegt dieses Wort in L_2, d. h.

$$i + j(i + k) + k \quad \text{prim}$$

Aber **Widerspruch**:

$$\begin{aligned} i + j(i + k) + k &= i + ij + jk + k \\ &= i(1 + j) + (j + 1)k \\ &= i(1 + j) + k(1 + j) \\ &= (i + k)(1 + j) \end{aligned}$$

Pumping-Lemma: Anwendung der Umkehrung

Beweis der Nichtregularität von L_2 (Forts.)

Fall 2: $i + k = 1$.

Pumpe $(j+2)$ mal:

$$uv^{j+2}w = a^i a^{j(j+2)} a^k$$

Nach Pumping-Lemma liegt dieses Wort in L_2, d. h.

$$i + j(j+2) + k \quad \text{prim}$$

Aber **Widerspruch!**:

$$\begin{aligned} i + j(j+2) + k &= 1 + j(j+2) \\ &= 1 + 2j + j^2 \\ &= (1+j)^2 \end{aligned}$$

Also: Annahme falsch. L_2 nicht regulär. $\square$

Pumping-Lemma: Beweis

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

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

mit

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

Pumping-Lemma: Beweis

Beweis. Sei L eine reguläre Sprache.

1. Fall: L ist endlich.

Sei w_{max} das längste Wort in L.

Wir setzen

$$n := |w_{max}| + 1$$

Dann gibt es keine Wörter $x \in L$, für die $|x| \geq n$ gilt.

Also gilt dann das Pumping-Lemma trivialerweise.

Pumping-Lemma: Beweis

Beweis (Fortsetzung) Sei L eine reguläre Sprache.

2. Fall: L ist unendlich.

Sei

$$\mathcal{A} = (K, \Sigma, \delta, s_0, F)$$

ein endlicher Automat (DEA), der L akzeptiert.

Wir setzen

$$n := |K| + 1$$

Pumping-Lemma: Beweis

Beweis (Fortsetzung)

Wir betrachten ein beliebiges Wort

$$x \;=\; x_1x_2\ldots x_t \;\in\; L \quad \text{mit} \quad |x| = t \geq n, \quad x_i \in \Sigma$$

Zu zeigen: x lässt sich aufpumpen.

Seien

$$q_0, q_1, \ldots, q_t \in K,$$

die Zustände, die beim Akzeptieren von x durchlaufen werden:

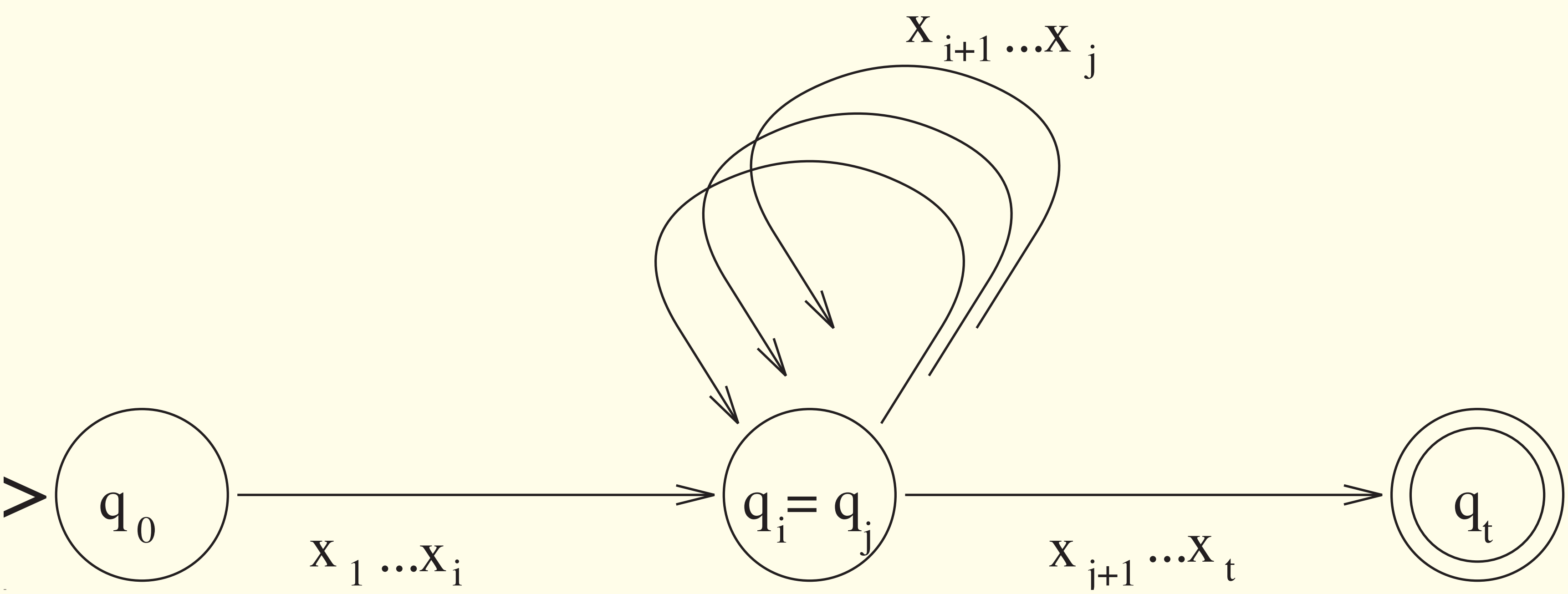
$$q_0 = s_0, \quad q_t \in F, \quad \delta(q_i, x_{i+1}) = q_{i+1} \;\; \forall 0 \leq i \leq t-1$$

Pumping-Lemma: Beweis

Beweis (Fortsetzung)

Da $t \geq |K| + 1$, muss es $0 \leq i < j \leq t - 1$ geben mit

- $q_i = q_j$
- $|j - i| \leq |K|$



Pumping-Lemma: Beweis

Beweis (Fortsetzung)

Wähle nun:

$$\left.\begin{array}{ll} u & x_1 \dots x_i \\ v & x_{i+1} \dots x_j \\ w & x_{j+1} \dots x_t \end{array}\right\} \qquad x = uvw \text{ mit } 1 \leq |v| < n.$$

Damit:

- Für alle $m \geq 0$ gibt es Wege

$$q_0, \dots, q_{i-1}, \underbrace{q_i, \dots, q_j = q_i, \dots, q_j = \cdots = q_i \dots, q_j}_{m \text{ mal}}, q_{j+1}, \dots, q_t$$

- Also: $uv^m w$ wird von $\mathcal{A}$ akzeptiert.
- Also: $uv^m w \in L$ $\square$

Pumping-Lemma: Stärkere Variante

Theorem (Pumping-Lemma für $\mathcal{L}_3$-Sprachen, stärkere Variante)
Sei $L \in \textbf{RAT}$. Dann existiert ein $n \in \mathbb{N}$, so dass:

Für alle

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

existiert eine Zerlegung

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

mit

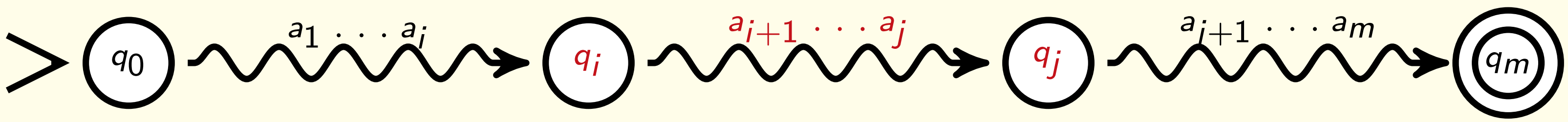
- $|v| \geq 1$
- $|uv| < n$ (statt $|v| < n$)
- $uv^m w \in L$ für alle $m \in \mathbb{N}$

Pumping-Lemma für $\mathcal{L}_3$

Beweis (Fortsetzung) (Fall L unendlich)

Betrachte Zustände $q_0, q_1, \ldots, q_{n-1}$, wobei $n - 1 = |K|$

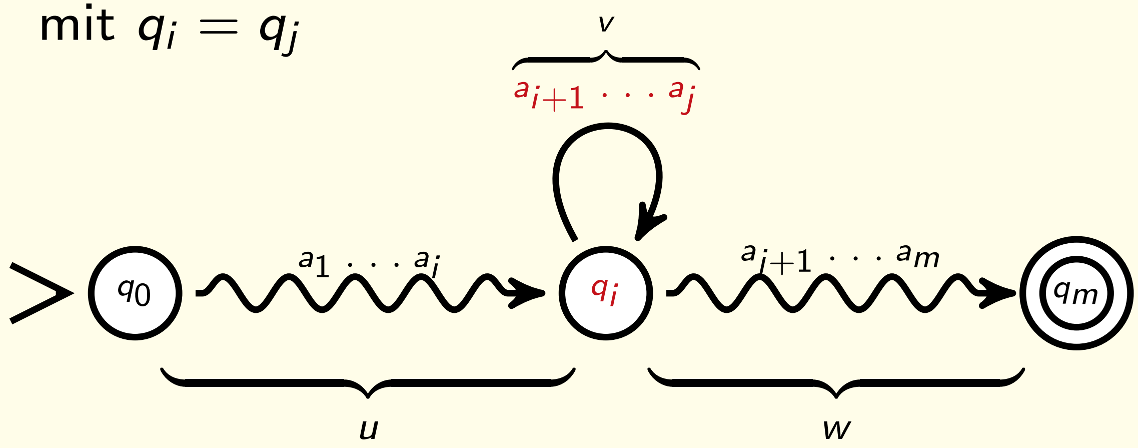
$\mapsto \exists i, j \;\; 0 \leq i < j \leq n - 1 \;\;$ mit $q_i = q_j$



Pumping-Lemma für $\mathcal{L}_3$

Betrachte Zustände $q_0, q_1, \ldots, q_{n-1}$, wobei $n - 1 = |K|$

$\mapsto \exists i, j \;\; 0 \leq i < j \leq n - 1 \;\;$ mit $q_i = q_j$

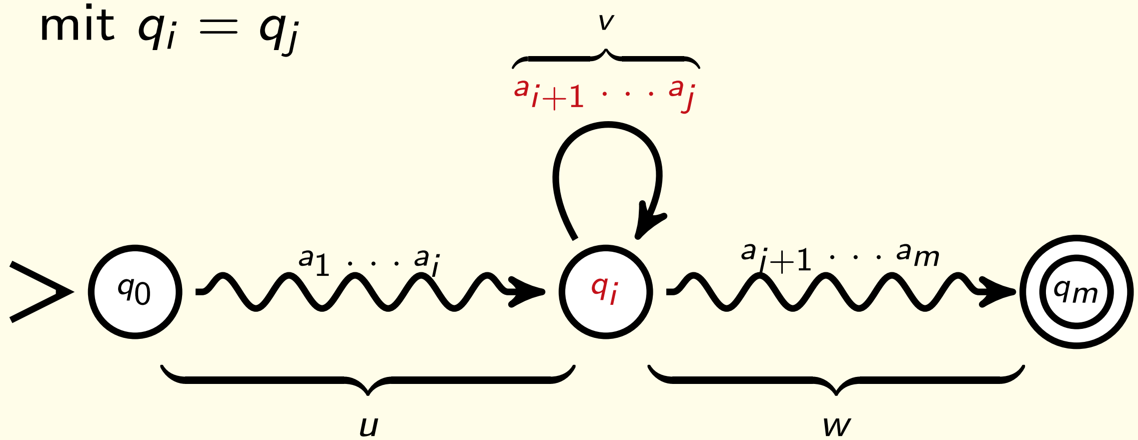


Damit: • $|v| \geq 1$, $|uv| \leq n - 1 = |K|$

Pumping-Lemma für $\mathcal{L}_3$

Betrachte Zustände $q_0, q_1, \ldots, q_{n-1}$, wobei $n - 1 = |K|$

$\mapsto \exists i, j \ \ 0 \leq i < j \leq n - 1$ mit $q_i = q_j$



Damit:

- $|v| \geq 1$, $|uv| \leq n - 1 = |K|$
- Für alle $k \geq 0$ gibt es Wege

  $$q_0 \xrightarrow{u} \underbrace{q_i \xrightarrow{v} q_i \xrightarrow{v} \ldots q_i \xrightarrow{v} q_i}_{k \text{ mal}} \xrightarrow{w} q_m$$

- Also: uv^kw wird von $\mathcal{A}$ akzeptiert, d.h. $uv^kw \in L$

Pumping-Lemma: Stärkere Variante

Theorem (Pumping-Lemma für $\mathcal{L}_3$-Sprachen, stärkere Variante)
Sei $L \in \textbf{RAT}$. Dann existiert ein $n \in \mathbb{N}$, so dass:

Für alle

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

existiert eine Zerlegung

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

mit

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

Pumping-Lemma: Anwendung der stärkeren Variante

Beispiel (Palindrome)

Die Sprache der Palindrome

$$L = \{ww^R \mid w \in \{a, b\}^*\}$$

ist nicht regulär

- Beweis gelingt **nicht** mit der schwächeren Variante des PL (die schwächere Version gilt für die Sprache)
- Beweis **gelingt** mit der stärkeren Varianten des PL

Pumping-Lemma: Anwendung der stärkeren Variante

Beispiel (Palindrome)

Die Sprache der Palindrome

$$L = \{ww^R \mid w \in \{a, b\}^*\}$$

ist nicht regulär

Annahme: L ist regulär. Dann gilt für L_1 das Pumping-Lemma. Sei n die Zahl aus dem Pumping-Lemma.

Dann muss sich das Wort $a^n bba^n \in L$ aufpumpen lassen (da $|a^n bba^n| \geq n$).

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

Da $|uv| < n$, ist $u = a^i, v = a^j, w = a^k bba^n$, wobei $j > 0$ und $i + j + k = n$

Aber dann $uv^0 w = a^{i+k} bba^n \notin L$, da $i + k < n$. Widerspruch.

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