

Vorlesung
Grundlagen der Theoretischen Informatik /
Einführung in die Theoretische Informatik I

Bernhard Beckert

Institut für Informatik



Sommersemester 2007

Dank

Diese Vorlesungsmaterialien basieren ganz wesentlich auf den Folien zu den Vorlesungen von

Katrin Erk (gehalten an der Universität Koblenz-Landau)

Jürgen Dix (gehalten an der TU Clausthal)

Ihnen beiden gilt mein herzlicher Dank.

– *Bernhard Beckert, April 2007*

Satz von Kleene

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:	Endvariable q,	erzeugt a	neue Endvariable q'

Satz von Kleene

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:	Endvariable q,	erzeugt a	neue Endvariable q'

Satz von Kleene

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: **Endvariable q** , **erzeugt a** neue **Endvariable 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 :

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

Daraus:

$$\begin{array}{l} S \xRightarrow{*}_G w \quad \underline{\text{gdw}} \quad \exists q \in F (S \xRightarrow{*}_G wq \Rightarrow w) \\ \underline{\text{gdw}} \quad \exists q \in F (\delta(s_0, w) = q) \\ \underline{\text{gdw}} \quad w \in L(\mathcal{A}) \end{array}$$

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 :

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

Daraus:

$$\begin{array}{l} S \xRightarrow{*}_G w \quad \underline{\text{gdw}} \quad \exists q \in F (S \xRightarrow{*}_G wq \Rightarrow w) \\ \underline{\text{gdw}} \quad \exists q \in F (\delta(s_0, w) = q) \\ \underline{\text{gdw}} \quad w \in L(\mathcal{A}) \end{array}$$

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 :

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

Daraus:

$$S \Longrightarrow_G^* w \quad \underline{\text{gdw}} \quad \exists q \in F (S \Longrightarrow_G^* wq \Longrightarrow w) \\ \underline{\text{gdw}} \quad \exists q \in F (\delta(s_0, w) = q) \\ \underline{\text{gdw}} \quad w \in L(\mathcal{A})$$

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 ε -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)

„ \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:

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

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

$$\Sigma := T$$

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

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:

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

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

$$\Sigma := T$$

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

Satz von Kleene

Beweis (Fortsetzung)

Definition von Δ :

$$\Delta((X, u), X') : \iff_{def} X \rightarrow uX' \in R$$

$$\Delta((X, u), q_{stop}) : \iff_{def} X \rightarrow u \in R$$

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

Durch Induktion über die Länge einer Ableitung:

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

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

Satz von Kleene

Beweis (Fortsetzung)

Definition von Δ :

$$\Delta((X, u), X') : \iff_{def} X \rightarrow uX' \in R$$

$$\Delta((X, u), q_{stop}) : \iff_{def} X \rightarrow u \in R$$

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



Satz von Kleene

Beweis (Fortsetzung)

Definition von Δ :

$$\Delta((X, u), X') : \iff_{def} X \rightarrow uX' \in R$$

$$\Delta((X, u), q_{stop}) : \iff_{def} X \rightarrow u \in R$$

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



Beispiel 14.2

ε -NDEA:

Grammatik G mit Regeln

$$S \rightarrow abaS$$

$$S \rightarrow aabS$$

$$S \rightarrow \varepsilon$$

Sprache

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

Beispiel 14.2

ε -NDEA:

Grammatik G mit Regeln

$$S \rightarrow abaS$$

$$S \rightarrow aabS$$

$$S \rightarrow \varepsilon$$

Sprache

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

Satz von Kleene

Beispiel 14.2

ϵ -NDEA:

Grammatik G mit Regeln

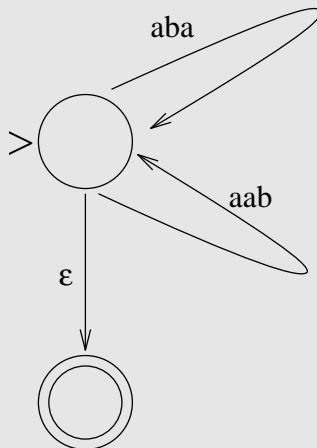
$S \rightarrow abaS$

$S \rightarrow aabS$

$S \rightarrow \epsilon$

Sprache

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



Teil III

- 1 Determinierte endliche Automaten (DEAs)
- 2 Indeterminierte endliche Automaten (NDEAs)
- 3 Automaten mit epsilon-Kanten
- 4 Endliche Automaten akzeptieren genau die Typ-3-Sprachen**
- 5 Pumping-Lemma
- 6 Wortprobleme
- 7 Rational = Regulär

Teil III

- 1 Determinierte endliche Automaten (DEAs)
- 2 Indeterminierte endliche Automaten (NDEAs)
- 3 Automaten mit epsilon-Kanten
- 4 Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- 5 Pumping-Lemma**
- 6 Wortprobleme
- 7 Rational = Regulär

Pumping-Lemma

„Aufpumpbarkeit“ (informell)

Lange Wörter $x \in L$ lassen sich zerlegen

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

so dass

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

i mal

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

Pumping-Lemma

„Aufpumpbarkeit“ (informell)

Lange Wörter $x \in L$ lassen sich zerlegen

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

so dass

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

i mal

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

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

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

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

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 auf mehrfach durchlaufen werden.
- Das Teilwort v , das der Schleife entspricht kann aufgepumpt werden.

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 auf mehrfach durchlaufen werden.
- Das Teilwort v , das der Schleife entspricht kann aufgepumpt werden.

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 auf mehrfach durchlaufen werden.
- Das Teilwort v , das der Schleife entspricht kann aufgepumpt werden.

Warum gilt das Pumping-Lemma?

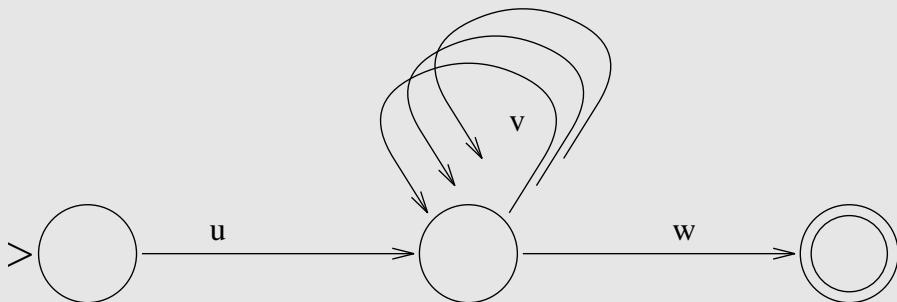
- Zu regulärer Sprache L gibt es einen DEA, der L akzeptiert
- Dieser hat endliche Zustandsmenge K .
Sei $m := |K|$.
- Wenn $|w| > |K|$, dann muss beim Akzeptieren von w eine Schleife durchlaufen werden.
- **Die Schleife kann auf mehrfach durchlaufen werden.**
- Das Teilwort v , das der Schleife entspricht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

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 auf mehrfach durchlaufen werden.
- **Das Teilwort v , das der Schleife entspricht kann aufgepumpt werden.**

Pumping-Lemma: Intuition

Abstrakt gesehen



Pumping-Lemma: Formal

Theorem 15.1 (Pumping-Lemma für L_3 -Sprachen)

Sei $L \in \mathbf{RAT}$.

Dann existiert ein $n \in \mathbb{N}$, so dass:

Für alle

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

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: Formal

Theorem 15.1 (Pumping-Lemma für 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: Formal

Theorem 15.1 (Pumping-Lemma für 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: Formal

Theorem 15.1 (Pumping-Lemma für 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: Formal

Theorem 15.1 (Pumping-Lemma für 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: Formal

Theorem 15.1 (Pumping-Lemma für 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

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 Pumping-Lemma trivialerweise.

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 Pumping-Lemma trivialerweise.

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 Pumping-Lemma trivialerweise.

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 Pumping-Lemma trivialerweise.

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 Pumping-Lemma trivialerweise.

Beweis (Fortsetzung)

2. Fall: L ist unendlich.

Sei

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

ein endlicher Automat (DEA), der L akzeptiert.

Wir setzen

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

Beweis (Fortsetzung)

2. Fall: L ist unendlich.

Sei

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

ein endlicher Automat (DEA), der L akzeptiert.

Wir setzen

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

Beweis (Fortsetzung)

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

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

Zu zeigen: x lässt sich aufpumpen.

Seien

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

die Zustände, die beim Akzeptieren von x durchlaufen werden:

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

Pumping-Lemma: Beweis

Beweis (Fortsetzung)

Wir betrachten ein beliebiges Wort

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

Zu zeigen: x lässt sich aufpumpen.

Seien

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

die Zustände, die beim Akzeptieren von x durchlaufen werden:

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

Pumping-Lemma: Beweis

Beweis (Fortsetzung)

Wir betrachten ein beliebiges Wort

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

Zu zeigen: x lässt sich aufpumpen.

Seien

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

die Zustände, die beim Akzeptieren von x durchlaufen werden:

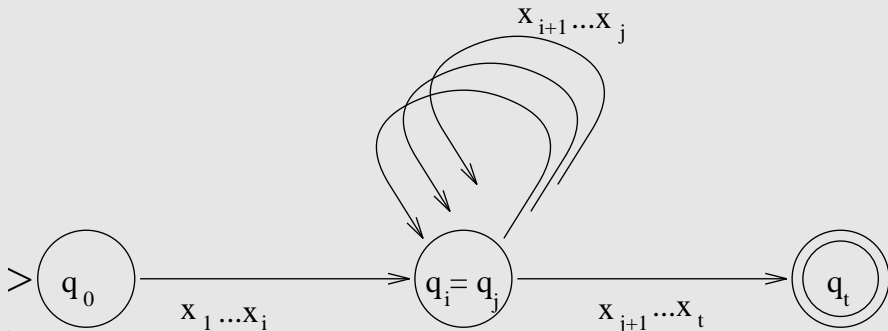
$$q_0 = s_0, \quad q_t \in F, \quad \delta(q_i, x_{i+1}) = q_{i+1} \quad \forall 0 \leq i \leq t-1$$

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:

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

Damit:

- Für alle $m \geq 0$ gibt es Wege

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

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

Pumping-Lemma: Beweis

Beweis (Fortsetzung)

Wähle nun:

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

Damit:

- Für alle $m \geq 0$ gibt es Wege

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

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

Pumping-Lemma: Beweis

Beweis (Fortsetzung)

Wähle nun:

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

Damit:

- Für alle $m \geq 0$ gibt es Wege

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

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

- Also: $uv^m w \in L$

Pumping-Lemma: Beweis

Beweis (Fortsetzung)

Wähle nun:

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

Damit:

- Für alle $m \geq 0$ gibt es Wege

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

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



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.
- Daraus, dass das Pumping-Lemma für eine Sprache gilt, folgt **nicht**, dass sie regulär ist.

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.
- Daraus, dass das Pumping-Lemma für eine Sprache gilt, folgt **nicht**, dass sie regulär ist.

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.
- Daraus, dass das Pumping-Lemma für eine Sprache gilt, folgt **nicht**, dass sie regulär ist.