Informatik ᗡ: Einführung in die Theoretische Informatik

VO 6
Beschreibungsäquivalenz (Nachtrag)

Pumping Lemma

Komplexität und Abschlusseigenschaften

Prof. Dr. Markus Chimani

Theoretische Informatik, Uni Osnabrück

Sommersemester 2013

Alles das gleiche!

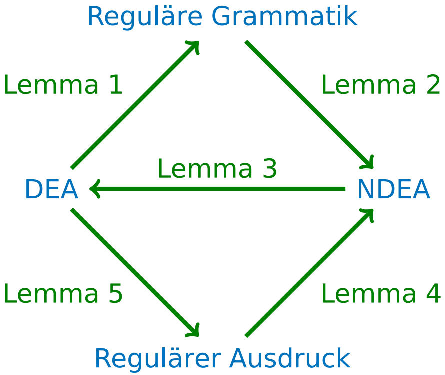
Theorem. Reguläre Grammatiken, reguläre Ausdrücke, DEAen und NDEAen erlauben alle **genau** die selben Sprachen zu beschreiben (nämlich die regulären Sprachen).

Beweis.
Ringschluss in 5 Lemmata:

Seien X und Y jew. eine der vier Beschreibungsarten.

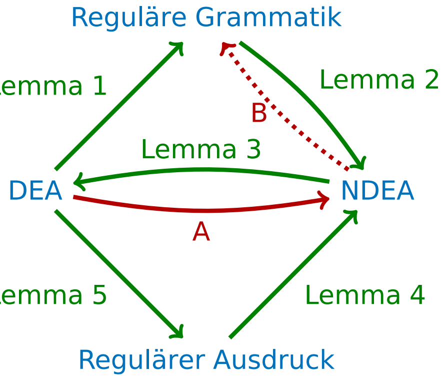
Lemma X $\rightarrow$ Y: Nimm eine beliebige Instanz der Art X. Wir erstellen eine Instanz der Art Y, die die selbe Sprache beschreibt.

$\Rightarrow$ Y kann mind. so viele Sprachen beschreiben wie X.



□

Apropos...



A Ist trivial, da DEA $\subset$ NDEA.

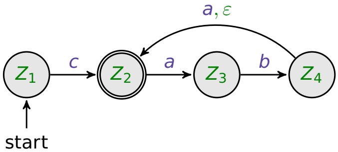
B Kann man für NDEAen **die keine ε-Übergänge enthalten** analog zu Lemma 1 machen.

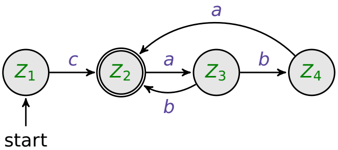
Reguläre Sprachen & endliche Automaten

Beispiele zum Äquivalenzbeweis (Wiederholung des Tafelbilds)

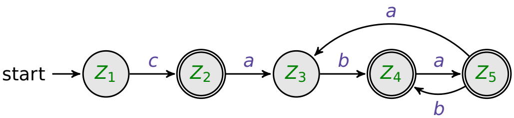
Beispielsprache

- **Reg. Ausdruck:** $c\,(ab \mid aba)^*$
- **Reg. Grammatik:**
$$S \rightarrow c \mid cA$$
$$A \rightarrow aB$$
$$B \rightarrow b \mid bA \mid bC$$
$$C \rightarrow a \mid aA$$
- **NDEA** (mit und ohne ε)

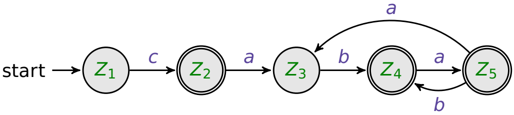




- **DEA**



Lemma 1: DEA $\rightarrow$ Reguläre Grammatik



Durch den Beweis generiert:

$$A_1 \rightarrow cA_2 \mid c$$
$$A_2 \rightarrow aA_3$$
$$A_3 \rightarrow bA_4 \mid b$$
$$A_4 \rightarrow aA_5 \mid a$$
$$A_5 \rightarrow bA_4 \mid aA_3$$

Handgestrickt:

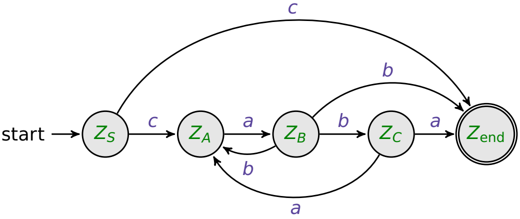
$$S \rightarrow c \mid cA$$
$$A \rightarrow aB$$
$$B \rightarrow b \mid bA \mid bC$$
$$C \rightarrow a \mid aA$$

Lemma 2: Reguläre Grammatik $\to$ NDEA

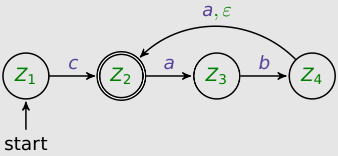
$$S \to c \mid cA \qquad A \to aB$$

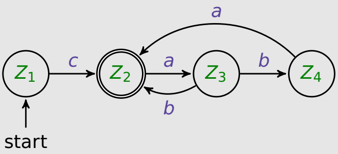
$$B \to b \mid bA \mid bC \qquad C \to a \mid aA$$

Durch den Beweis generiert:

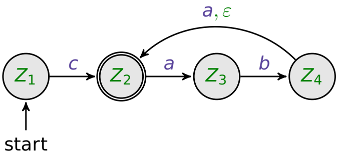


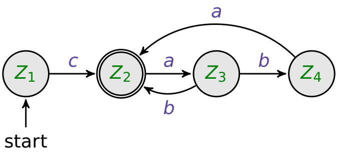
Handgestrickt:



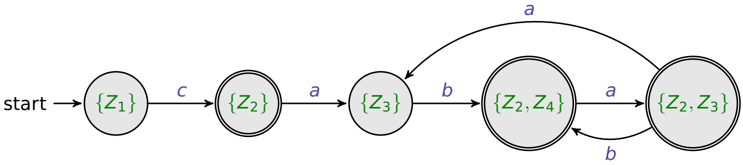


Lemma 3: NDEA $\to$ DEA

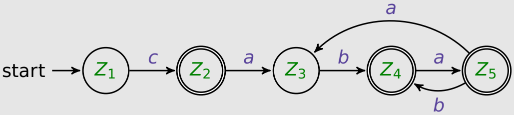




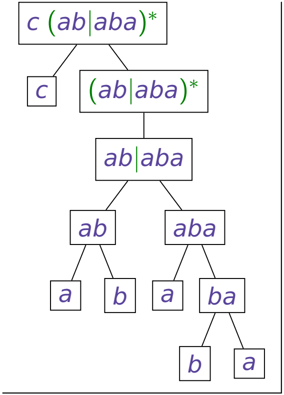
Durch den Beweis generiert:



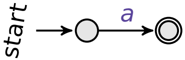
Handgestrickt:



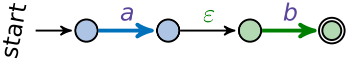
Lemma 4: RegEx → NDEA



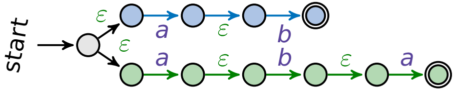
a: (*b*, *c* analog)



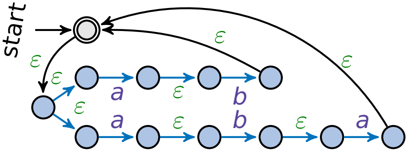
ab: (*aba* analog)



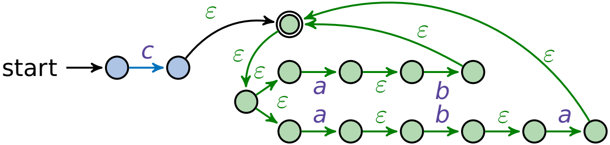
ab|*aba*:



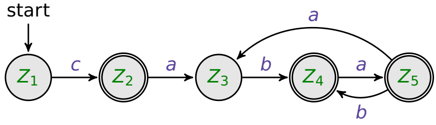
(*ab*|*aba*)*:



c(*ab*|*aba*)*:



Lemma 5: DEA $\rightarrow$ RegEx



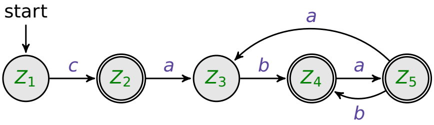
0,1,[2]	1	2	3	4	5
1	ε	c	$[ca]$	$\emptyset$	$\emptyset$
2	$\emptyset$	ε	a	$\emptyset$	$\emptyset$
3	$\emptyset$	$\emptyset$	ε	b	$\emptyset$
4	$\emptyset$	$\emptyset$	$\emptyset$	ε	a
5	$\emptyset$	$\emptyset$	a	b	ε

3	1	2	3	4	5
1	ε	c	ca	cab	$\emptyset$
2	$\emptyset$	ε	a	ab	$\emptyset$
3	$\emptyset$	$\emptyset$	ε	b	$\emptyset$
4	$\emptyset$	$\emptyset$	$\emptyset$	ε	a
5	$\emptyset$	$\emptyset$	a	$b\|ab$	ε

4	1	2	3	4	5
1	ε	c	ca	cab	$caba$
2	$\emptyset$	ε	a	ab	aba
3	$\emptyset$	$\emptyset$	ε	b	ba
4	$\emptyset$	$\emptyset$	$\emptyset$	ε	a
5	$\emptyset$	$\emptyset$	a	$b\|ab$	$x := \varepsilon\|(b\|ab)a$

5	1	2	3	4	5
1	ε	c	$ca\|cabax^*a$	$cab\|cabax^*(b\|ab)$	$cabax^*$
2	$\emptyset$	ε	$a\|abax^*a$	$ab\|abax^*(b\|ab)$	$abax^*$
3	$\emptyset$	$\emptyset$	$\varepsilon\|bax^*a$	$b\|bax^*(b\|ab)$	bax^*
4	$\emptyset$	$\emptyset$	ax^*a	$\varepsilon\|ax^*(b\|ab)$	ax^*
5	$\emptyset$	$\emptyset$	x^*a	$x^*(b\|ab)$	x^*

Lemma 5: DEA → RegEx



0,1,[2]	1	2	3	4	5
1	ε	c	[ca]	$\emptyset$	$\emptyset$
2	$\emptyset$	ε	a	$\emptyset$	$\emptyset$
3	$\emptyset$	$\emptyset$	ε	b	$\emptyset$
4	$\emptyset$	$\emptyset$	$\emptyset$	ε	a
5	$\emptyset$	$\emptyset$	a	b	ε

3	1	2	3	4	5

4	1	2	3	4	5
					$caba$
					aba
					ba
					a
					$x := \varepsilon \vert (b \vert ab)a$

Startzustand: 1, Endzustände: 2,4,5

RegEx, durch den Beweis generiert:

$$c \mid cab \mid caba(\varepsilon \mid (b \mid ab)a)^*(b \mid ab) \mid caba(\varepsilon \mid (b \mid ab)a)^*$$

RegEx, handgestrickt:

$$c\,(ab \mid aba)^*$$

5	1	2	3	4	5
1	ε	c	$ca \vert cabax^*a$	$cab \vert cabax^*(b \vert ab)$	$cabax^*$
2	$\emptyset$	ε	$a \vert abax^*a$	$ab \vert abax^*(b \vert ab)$	$abax^*$
3	$\emptyset$	$\emptyset$	$\varepsilon \vert bax^*a$	$b \vert bax^*(b \vert ab)$	bax^*
4	$\emptyset$	$\emptyset$	ax^*a	$\varepsilon \vert ax^*(b \vert ab)$	ax^*
5	$\emptyset$	$\emptyset$	x^*a	$x^*(b \vert ab)$	x^*

Reguläre Sprachen & endliche Automaten

Pumping Lemma

Nicht-Regularität

Gegeben: Eine Sprache L.

Frage: Ist L regulär?

Falls ja:
Beweis durch reg. Grammatik, reg. Ausdruck, DEA oder NDEA, die/der L beschreibt.

Falls nein:
Wie beweist man, dass eine Sprache **nicht** regulär ist?

- **Pumping Lemma** $\rightarrow$ jetzt
- Myhill-Nerode Theorem: Äquivalenzrelation und Minimalautomat (werden wir nicht besprechen)

Pumping Lemma

Pumping Lemma (für reguläre Sprachen).
Sei L eine reguläre Sprache. Es gibt eine Zahl $n := n(L)$ (d.h. in Abhängigkeit von L), so dass alle Wörter $z \in L$ mit $|z| \geq n$ sich zerlegen lassen als $z = u\,v\,w$ mit den Eigenschaften:

1 $|v| \geq 1$, **2** $|u\,v| \leq n$, **3** $u\,v^*\,w \in L$.

Beweis. $L \to \exists$ DEA $\mathcal{A}$ mit Zuständen $\mathcal{Z}$. Wähle $n := |\mathcal{Z}|$.

Bei $\mathcal{A}$-Abarbeitung eines Wortes z werden $|z| + 1$ Zustände abgelaufen (inkl. Startzustand). Da $|z| \geq n$: mindestens ein Zustand wird öfters (mind. 2x) besucht.

Wähle Zerlegung $z = uvw$ so, dass das man nach Lesen des letzten Symbols von u und von uv im gleichen Zustand (Z_i) ist. Dabei ist es trivial, **1** und **2** zu erfüllen.

Nun könnte man von Z_i aus auch mehrmals (inkl. 0-mal) v ablaufen, bevor man w abläuft $\to uv^*w \in L \to$ **3**. □

Anwendung des Pumping Lemmas

$\forall$ reg. Spr. L: $\exists n$: $\forall z \in L$ mit $|z| \geq n$: $\exists u\,v\,w = z$ mit:

1 $|v| \geq 1$, **2** $|u\,v| \leq n$, **3** $u\,v^*\,w \in L$.

Aufgabe: Sei $L = \{a^i b^i \mid i \geq 0\}$ die Sprache der Worte deren vordere Hälfte lauter a, und deren hintere Hälfte lauter b sind. Zeige, dass L nicht regulär ist.

Lösung: Beweis durch Widerspruch.

Nimm an, L wäre regulär, dann würde für L das Pumping Lemma gelten. Sei n die entsprechende Wortmindestgröße.

Betrachte das Wort $z = a^n b^n \in L$.

- $\exists$ Zerteilung $z = uvw$ mit **1** – **3**.
  2: uv besteht nur aus a-Symbolen.
  1: $|v| = \ell \geq 1$.
- Da **3**: $uw = a^{n-\ell} b^n \in L \rightarrow$ Widerspruch. $\square$

Anwendung des Pumping Lemmas

$\forall$ reg. Spr. L: $\exists n$: $\forall z \in L$ mit $|z| \geq n$: $\exists u\,v\,w = z$ mit:

1 $|v| \geq 1$, **2** $|u\,v| \leq n$, **3** $u\,v^*\,w \in L$.

Aufgabe: Sei $L = \{a^{2^i} \mid i \geq 0\}$, d.h. Worte die aus „2er-Potenz" vielen „a"s bestehen. Zeige, dass L nicht regulär ist.

Lösung: Beweis durch Widerspruch.
Nimm an, L wäre regulär, dann würde für L das Pumping Lemma gelten. Sei n die entsprechende Wortmindestgröße.

Wähle ein k, so dass $2^k > n$. Betrachte das Wort $z = a^{2^k} \in L$.

- $\exists$ Zerteilung $z = uvw$ mit **1** – **3**.
  2: $|uv| \leq n \to |w| \geq 1$.
- Da **3**: $y = uv^2w \in L$: $|uv^2w| = |uvw| + |v| < 2|uvw|$
  $\to$ y ist länger als z, aber kürzer als nächste 2er-Potenz.
  $\to$ Widerspruch. $\square$

Grenzen des Pumping Lemmas

Beobachtung.
Es gibt nicht-reguläre Sprachen, für die das Pumping Lemma **nicht stark genug** ist, um die Nicht-Regularität zu beweisen.

Beispiel. Sei $L = \{c^j a^i b^i \mid i, j \geq 0\} \cup \{a^j b^i \mid i, j \geq 0\}$.
$\rightarrow$ L kann nicht regulär sein, da (in erster Teilmenge) nur ein paar c vor einer nicht-regulären Sprache stehen (siehe vorhin)

Pumping Lemma.
Betrachte bel. Wort $z = c^j a^i b^i$ oder $z = a^j b^i$ mit $|z| \geq n$ aus L.
Zerlegung: $u = \varepsilon$, $v = z[1]$, $w = [2 \ldots]$ (v ist das erstes Symbol von z, w ist der Rest). $\rightarrow$ Jedes $u\, v^*\, w$ liegt in L!
$\rightarrow$ Kein Widerspruch $\rightarrow$ Beweis funktioniert nicht.

$\Rightarrow$ Es gibt jedoch andere Methoden (statt dem Pumping Lemma), um den Beweis der Nicht-Regularität zu führen (siehe einschlägige Literatur).

Reguläre Sprachen & endliche Automaten

Komplexität und Abschlusseigenschaften

Komplexitäten

Sei L eine reguläre Sprache. Annahme: L ist als DEA gegeben (Zustände $\mathcal{Z}$, Startzustand Z_{start}, Endzustände $\mathcal{Z}_{\text{end}}$).

Wortproblem. Sei w ein Wort. Ist $w \in L$?
Laufe den DEA mit w als Eingabe ab.
Nach $\mathcal{O}(|w|)$ Schritten kennt man die Antwort.

Leerheitsproblem. Ist $L = \emptyset$?
$L \neq \emptyset \iff \exists Z' \in \mathcal{Z}_{\text{end}}$ mit einem Pfad $Z_{\text{start}} \rightsquigarrow Z'$.
$\to$ Mittels Tiefensuche in linearer Zeit $\mathcal{O}(|\mathcal{Z}|)$ entscheidbar.

Endlichkeitsproblem. Ist L eine endliche Menge?
L unendlich $\iff \exists Z_0 \in \mathcal{Z}, Z' \in \mathcal{Z}_{\text{end}}$ mit Pfaden $Z_{\text{start}} \rightsquigarrow Z_0$, $Z_0 \rightsquigarrow Z'$ und einem Kreis, der Z_0 enthält.
$\to$ Mittels Tiefensuche in linearer Zeit $\mathcal{O}(|\mathcal{Z}|)$ entscheidbar.

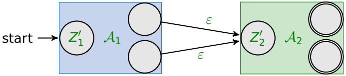
Abgeschlossenheit

Theorem.
Reguläre Sprachen sind abgeschlossen bezüglich Verkettung, Vereinigung, Schnitt und Komplementbildung.

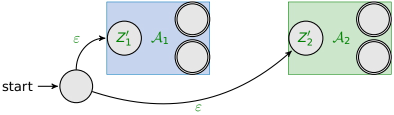
D.h. gegeben zwei reguläre Sprachen L_1, L_2. Die folgenden Sprachen sind auch regulär: $L_1 L_2$, $L_1 \cup L_2$, $L_1 \cap L_2$, $\Sigma^* \setminus L_1$.

Beweis. Seien $\mathcal{A}_1, \mathcal{A}_2$ die zugehörigen deterministische EAen.

Verkettung.



Vereinigung.



Schnitt.
$\rightarrow$ nächste Seite.

Komplementbildung.
Alle Nicht-Endzustände von $\mathcal{A}_1$ (inkl. Falle) werden Endzustände, und umgekehrt.

Abgeschlossenheit

Beweis (Schnitt).
$w \in L_1 \cap L_2 \iff \mathcal{A}_1$ **und** $\mathcal{A}_2$ akzeptieren w.
$\rightarrow$ Wir bauen einen (deterministischen) endlichen Automaten, der das gleichzeitige Ablaufen beider Automaten simuliert.

Seien $\mathcal{Z}, \mathcal{Y}$ die Zustandsmengen der Automaten.
Neuer Automat:

- Zustandsmenge $\mathcal{Z} \times \mathcal{Y}$
  (=alle Paare von $\mathcal{A}_1$-$\mathcal{A}_2$-Zuständen)
- Startzustand $(Z_{\text{start}}, Y_{\text{start}})$, wobei $Z_{\text{start}}, Y_{\text{start}}$ die Startzustände der einzelnen Automaten sind
- Übergänge $\delta(\,(Z, Y), a\,) := (\,\delta_1(Z, a), \delta_2(Y, a)\,)$ für alle $Z \in \mathcal{Z}, Y \in \mathcal{Y}, a \in \Sigma$
- Ein Zustand (Z, Y) ist ein Endzustand genau dann wenn $Z \in \mathcal{Z}_{\text{end}}$ und $Y \in \mathcal{Y}_{\text{end}}$ □

Reguläre Sprachen & endliche Automaten

Zusammenfassung

Reguläre Sprachen: Zusammenfassung

Reguläre Sprachen...

...werden beschrieben durch

- **Reguläre Grammatiken**
- **Reguläre Ausdrücke**
- **Deterministische Endliche Automaten**
- **Nicht-deterministische Endliche Automaten**

...erfüllen

- **Pumping Lemma**
- **Abgeschlossenheit** bzgl. Verkettung, Vereinigung, Schnitt und Komplementbildung

...erlauben

- das **effiziente Entscheiden** (in Linearzeit!) des Wort-, Leerheits- und Endlichkeitsproblems

Reguläre Sprachen: Zusammenfassung

Motivation:

- *Hopcroft-Motwani-Ullman:* 200 von 600 Seiten geschafft
- *Schöning:* 40 von 180 Seiten geschafft