

Vorkurs Theoretische Informatik

Einführung in reguläre Sprachen

Arbeitskreis Theo-Vorkurs

Freitag, 10. Oktober 2024

Fachgruppe Informatik Universität Stuttgart



1. Automaten: Weitere Beispiele

2. Grammatik und Automaten

3. Reguläre Ausdrücke

4. Wiederholung



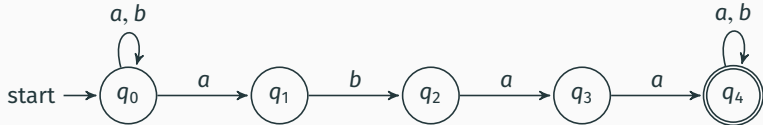
Automaten: Weitere Beispiele



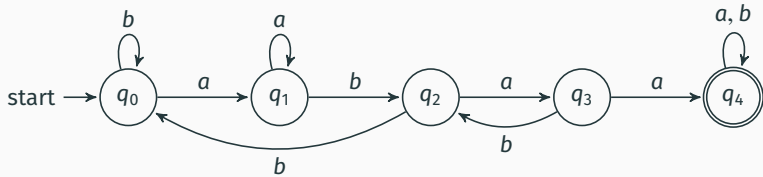
Beispiel NEA vs DEA

$$L = \{uabaav \mid u, v \in \{a, b\}^*\}$$

NEA:



DEA:



Aufgaben

Normal

Finde einen passenden DEA oder NEA für die folgenden Sprachen:

- $L_1 = \{w \in \{a, b\}^* \mid |w|_a \equiv 1 \pmod{3}\}$
- $L_2 = \{ua \mid u \in \{a, b\}^*\}$

Etwas Schwerer

Finde einen passenden NEA mit **einem** Startzustand für die folgende Sprache

- $L_3 = L_1 \cup L_2$

Sehr Schwer

Finde für L_3 einen passenden DEA



Grammatik und Automaten



Ein **DEA M** lässt sich beschreiben durch ein geordnetes 5-Tupel

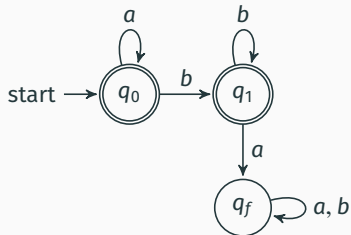
$M = (Z, \Sigma, \delta, z_0, E)$ mit:

- Z : Die Menge der Zustände
- Σ : Das Alphabet
- δ : Die Überföhrungsfunktion
- z_0 : Der Startzustand
- E : Die Menge der Endzustände



Automaten: Formal - Beispiel

$$L_1 = \{a^n b^m \mid n, m \in \mathbb{N}\}$$



$M = (Z, \Sigma, \delta, q_0, E)$ mit:

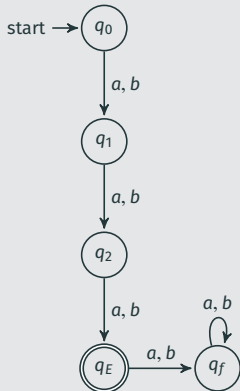
- $Z = \{q_0, q_1, q_f\}$
- $\Sigma = \{a, b\}$
- δ :
 - $\delta(q_0, a) = q_0$
 - $\delta(q_0, b) = q_1$
 - $\delta(q_1, a) = q_f$
 - $\delta(q_1, b) = q_1$
 - $\delta(q_f, a) = q_f$
 - $\delta(q_f, b) = q_f$
- $E = \{q_0, q_1\}$



Aufgabe

Normal

$$L_2 = \{w \in \{a, b\}^* \mid |w| = 3\}$$



Automaten und Grammatiken: Parallelen

Wir stellen einige Parallelen fest:

DEA	Reguläre Grammatiken
Wörter werden von links nach rechts gelesen	Wörter werden von links nach rechts erzeugt
Pro Schritt wird ein Buchstabe gelesen	Pro Schritt wird ein Buchstabe erzeugt
Ein Startzustand	Ein Startsymbol
Wenn beim Lesen des letzten Buchstabens in einen Endzustand übergegangen wird, wird akzeptiert	Beim Erzeugen des letzten Buchstabens wird keine neue Variable erzeugt
In jedem Schritt wird aus einem Zustand in genau einen Zustand übergegangen	In jedem Schritt wird aus einer Variable genau eine Variable erzeugt



Satz

Jede durch deterministische endliche Automaten erkennbare Sprache ist auch regulär (also Typ 3).

Der Beweis dieses Satzes findet sich im Anhang.
Ihr werdet ihn auch in der Vorlesung zeigen.



Reguläre Ausdrücke



Automaten und Mengenschreibweise sind oft nicht das optimale Mittel, eine Sprache zu beschreiben.

Die **regulären Ausdrücke** bieten uns eine Möglichkeit Sprachen schnell und intuitiv zu beschreiben.

Funktionsweise

1. Wörter können mit einem angegebenen Muster abgeglichen werden.
2. Lässt sich ein Wort durch das Muster beschreiben, ist es in der davon beschriebenen Sprache.



Induktive Definition der Syntax

- \emptyset und ε sind reguläre Ausdrücke.
- a ist ein regulärer Ausdruck (für alle $a \in \Sigma$).
- Wenn α und β reguläre Ausdrücke sind, dann sind $\alpha\beta$, $(\alpha \mid \beta)$ und $(\alpha)^*$ auch reguläre Ausdrücke.

Beispiele für die Syntax

Reguläre Ausdrücke können zum Beispiel so aussehen...

- $\gamma_1 = (a|b)baa(ab)^*$
- $\gamma_2 = ((\varepsilon|baa(b|bb))aa)^*$



Reguläre Ausdrücke und Sprachen

Ein regulärer Ausdruck γ beschreibt eine formale Sprache.

Diese formale Sprache schreiben wir als $L(\gamma) \subseteq \Sigma^*$.

Wir bauen reguläre Ausdrücke schrittweise auf und erklären dabei die Bedeutung der einzelnen Bausteine:

Bedeutung von regulären Ausdrücken

- Sei $\gamma = \emptyset$: Dann ist $L(\gamma) = \emptyset$
- Sei $\gamma = \varepsilon$: Dann ist $L(\gamma) = \{\varepsilon\}$
- Sei $\gamma = a$ für ein $a \in \Sigma$: Dann ist $L(\gamma) = \{a\}$



Bedeutung von regulären Ausdrücken

Bedeutung von regulären Ausdrücken

Sind α und β bereits reguläre Ausdrücke, so gilt:

- Sei $\gamma = (\alpha \mid \beta)$: Dann ist $L(\gamma) = L(\alpha) \cup L(\beta)$

Beispiel: $\gamma = (\alpha \mid \beta)$

- $L(a|b) = \{a, b\}$
- $L((aab|baba)|a) = \{aab, baba, a\}$



Bedeutung von regulären Ausdrücken

Bedeutung von regulären Ausdrücken

Sind α und β bereits reguläre Ausdrücke, so gilt:

- Sei $\gamma = (\alpha \mid \beta)$: Dann ist $L(\gamma) = L(\alpha) \cup L(\beta)$
- Sei $\gamma = (\alpha)^*$: Dann ist $L(\gamma) = L(\alpha)^*$

Beispiel: $\gamma = (\alpha)^*$

- $L((a)^*) = \{a\}^* = \{a^n \mid n \in \mathbb{N}\}$
- $L((abaa)^*) = \{abaa\}^* = \{w^n \mid n \in \mathbb{N}, w = abaa\}$
- $L((a|b)^*) = (L(a|b))^* = \{a, b\}^*$
- $L((a)^*|(b)^*) = L((a)^*) \cup L((b)^*) = \{a\}^* \cup \{b\}^*$



Bedeutung von regulären Ausdrücken

Bedeutung von regulären Ausdrücken

Sind α und β bereits reguläre Ausdrücke, so gilt:

- Sei $\gamma = (\alpha \mid \beta)$: Dann ist $L(\gamma) = L(\alpha) \cup L(\beta)$
- Sei $\gamma = (\alpha)^*$: Dann ist $L(\gamma) = L(\alpha)^*$
- Sei $\gamma = \alpha\beta$: Dann ist $L(\gamma) = \{uv \mid u \in L(\alpha) \wedge v \in L(\beta)\}$

Beispiel: $\gamma = \alpha\beta$

- $L(ababba) = \{ababba\}$
- $L((a)^*(b)^*) = \{a^n b^m \mid n, m \in \mathbb{N}\}$
- $L((bab \mid (ab)^*)abb) = \{bababb\} \cup \{(ab)^n abb \mid n \in \mathbb{N}\}$



Aufgaben

Finde einen regulären Ausdruck γ_i für die folgenden Sprachen.

Normal

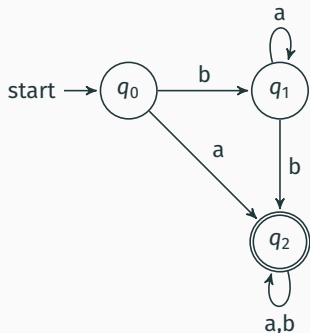
- $L(\gamma_1) = \{a^{2n} \mid n \in \mathbb{N}\}$
- $L(\gamma_2) = \{a^n b^m \mid n, m \in \mathbb{N}\}$
- $L(\gamma_3) = \{uv \mid u \in \{a, b\}^*, v \in \{c, d\}\}$
- $L(\gamma_4) = \{w \mid |w| = 3, w \in \{a, b, c\}^*\}$

Etwas Schwerer

- $L(\gamma_5) = \{a^n \mid n \equiv 1 \pmod{3}\}$
- $L(\gamma_6) = \{uv \mid u \in \{\blacktriangleleft, \blacktriangle, \blacktriangleright, \blacktriangledown\}^*, v \in \{\text{STOP}\}\}$
- $L(\gamma_7) = \{w \mid |w|_a = 3, |w|_b = 1, w \in \{a, b, c\}^*\}$



Gegeben sei folgender DEA M:



Welcher reguläre Ausdruck beschreibt

$T(M)$?

1. $(a|b(a)^*b)(a|b)^*$
2. $a(ab)^*$
3. $(a|b(a)^*b)(b)^*$
4. $(a|b)^*$



Aufgaben

Welche der folgenden Aussagen sind wahr/falsch? Begründe.

Normal

- $A_1 : L((a|a)^*) = L((aa)^*)$
- $A_2 : L((a|b)^*) = L((a)^* | (b)^*)$
- $A_3 : L((a(a|b)^*) | (b(a|b)^*)) = \{a, b\}^*$

Schwer

- $A_4 : L((\varepsilon|aaa)^* a (((aaa)^* | (\varepsilon)^* |aaa))) = \{a^n \mid n \equiv 1 \pmod{3}\}$

Sehr Schwer

- $A_5 : L((a|b)^*) = L(((a)^* b)^* (a)^*)$



Wiederholung



Tag 1: Mengen

- Was ist eine Menge?
- Wie kann man zwei Mengen verknüpfen?
- Wie schreibt man formal Mengen auf?



Tag 1: Formale Sprachen

- Was ist eine Formale Sprache?
- Was ist ein Alphabet?
- Wie zeigt man, dass zwei Sprachen äquivalent sind?



Aufgaben

Beantworte die folgenden Fragen

Normal

- Was ist der Unterschied zwischen Σ und Σ^* ?
- Welche besondere Eigenschaft besitzt das leere Wort ε ? Insbesondere:
Welchen Wert besitzt $|\varepsilon|$?
- Welchen Wert besitzt $|\varepsilon \cdot a^2 \cdot \varepsilon \cdot bab|_a$?



Aufgaben

Welche der folgenden Aussagen ist korrekt?

Normal bis etwas Schwerer

- $a \in \{a, b, c\}$
- $a \subseteq \{a, b, c\}$
- $\{a, b\} \in \{a, b, \{a, b\}, c\}$
- $\{a, b\} \subseteq \{a, b, \{a, b\}, c\}$
- $\{\{a, b\}\} \subseteq \{a, b, \{a, b\}, c\}$

Nenne jeweils 5 Wörter aus den folgenden Sprachen

Normal bis etwas Schwerer

- $L_1 = \{a^n b^m \mid n, m \in \mathbb{N}\}$
- $L_2 = \{a^n b^n \mid n \in \mathbb{N}\}$
- $L_3 = L_2 \setminus L_1$
- $L_4 = \{w \in \{a, b\}^* \mid |w|_a \equiv 3 \pmod{4}\}$



Aufgaben

Bestimme den Wahrheitswert der folgenden Aussagen

Normal

- $\neg(42 = 11) \wedge (|\varepsilon| > 0 \vee \emptyset^* = \{\varepsilon\})$
- $\forall w \in \{a, b\}^* : (|w|_a = 0 \implies w = \varepsilon)$
- $\forall w \in \{a, b\}^* : (|w|_a = 0 \implies ab = ba)$
- $(\forall w \in \{a, b\}^* : |w|_a = 0) \implies ab = ba$

Schwer

- $\forall x \in \mathbb{Z} : x \equiv 0 \pmod{2} \vee x \equiv 1 \pmod{4}$



Tag 2: Beweise

- Was ist ein direkter Beweis?
- Wie funktioniert die Kontraposition?
- Wie funktioniert ein Widerspruchsbeweis?



Aufgaben

Versuche dich an folgenden Beweisen:

Normal

Sei $L = \{a^n b^n \mid n \in \mathbb{N}\}$, dann gilt:

$$\forall w \in \{a, b\}^* : (w \in L \implies |w| \text{ ist gerade})$$

Wer noch nicht genug hat...

$$\forall x \in \mathbb{Z} : x \equiv 0 \pmod{2} \vee x^2 \equiv 1 \pmod{4}$$



Tag 3: Grammatiken

- Was sind Grammatiken?
- Was ist der Zusammenhang zwischen Grammatiken und Sprachen?
- Wie finde ich raus, ob ein Wort von einer Grammatik erzeugt wird?



Tag 4: Reguläre Grammatiken

- Wie sehen Produktionsregeln für reguläre Grammatiken aus?
- Bilden einer regulären Grammatik für gegebene reguläre Sprache



Tag 4: Automaten

- Was sind Automaten?
- Was macht einen deterministischen Automaten aus?
- Finden eines (deterministischen) Automaten für gegebene Sprache



Aufgaben

Gegeben sind formale Sprachen. Finde dafür:

- eine reguläre Grammatik
- einen Automaten (NEA oder DEA)
- einen regulären Ausdruck

Normal

- $L_1 = \{aa, bb\}$ über dem Alphabet $\Sigma_1 = \{a, b\}$

Etwas Schwerer

- $L_2 = \{(ab)^n x \mid n \in \mathbb{N} \wedge x \in \{c, d\}\}$ über $\Sigma_2 = \{a, b, c, d\}$

Hinweis: Achte auf formal korrekte Notation.



Heute: Repräsentationen regulärer Sprachen

- Welche Möglichkeiten gibt es, reguläre Sprachen zu beschreiben?
- Wie wandelt man Automaten zu einer äquivalenten Grammatik um?
- Was ist ein regulärer Ausdruck?



Heute: Reguläre Ausdrücke

- Wie funktioniert die Konkatination?
- Was bedeuten $(\alpha | \beta)$ und $(\alpha)^*$?
- Finden eines regulären Ausdrucks für gegebene reguläre Sprache



Abk.	Bedeutung	Was?!
\mathbb{N}	natürliche Zahlen (mit 0)	In der theoretischen Informatik enthält \mathbb{N} die 0: $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ (Auch \mathbb{N}_0)
\mathbb{Z}	ganze Zahlen	$\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
Σ	Sigma	mit diesem Zeichen wird oft das Alphabet (die Menge an verwendbaren Symbolen) repräsentiert
Σ^*	Sigma Stern	Menge aller Möglichkeiten Elemente aus Σ hintereinander zu schreiben
ε	leeres Wort	Wort (über bel. Alphabet) mit der Länge 0 ($ \varepsilon = 0$), in allen Σ^* enthalten.
\emptyset	$\{\}$	leere Menge
$a b$	teilt	a ist Teiler von b , d.h. a teilt b ohne Rest
:	sodass	z.B. $\forall a, b \in \mathbb{Z} : a b$
mod	modulo	$a \equiv b \pmod{n} \iff n (a - b)$, mit $a, b, n \in \mathbb{Z}$



Abk.	Bedeutung	Was?!
z.z. Sei	zu zeigen	Was zu beweisen ist bereits bekannte Objekte werden eingeführt und benannt
\exists	es gibt ein	
$\exists!$	es gibt genau ein	
x ist genau y	$x = y$	<i>genau</i> wird verwendet bei Äquiva- lenz
x ist eindeutig der, die, das	$\exists!x$	bestimmte Artikel weisen auf Ein- deutigkeit hin
gdw.	genau dann, wenn	Äquivalenz zwischen Aussagen



Abk.	Bedeutung	Was?!
A ist notwendig für B	$B \implies A$	A muss wahr sein, wenn B wahr ist
A ist hinreichend für B	$A \implies B$	B muss wahr sein, wenn A wahr ist
notwendig und hinreichend	$A \iff B$	genau dann, wenn



Abk.	Bedeutung	Was?!
∅	ohne Einschränkung	die Allgemeinheit der Aussage wird nicht durch getroffene Aussagen eingeschränkt
o.B.d.A.	ohne Beschränkung der Allgemeinheit	wie ∅
trivial	offensichtlich	Beweisschritte, welche keine weiter Begründung brauchen. (nicht verwenden!)
□	Mic Drop	Kommt am Ende eines erfolgreichen Beweises
q.e.d.	quod erat demonstrandum	Was zu beweisen war


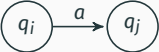




Gestalt	mögliches Vorgehen
nicht F	Zeige, dass F nicht gilt.
F und G	Zeige F und G in zwei getrennten Beweisen.
$F \implies G$	Füge F in die Menge der Annahmen hinzu und zeige G.
F oder G	Zeige: nicht $F \implies G$. (Alternativ zeige: nicht $G \implies F$)
$F \iff G$	Zeige: $F \implies G$ und $G \implies F$.
$\forall x \in A : F$	Sei x ein beliebiges Element aus A. Zeige dann F.
$\exists x \in A : F$	Sei x ein konkretes Element aus A. Zeige dann F.



Abk.	Bedeutung	Was?!
$A \subseteq B$	Teilmenge	Alle Elemente aus A sind auch in B enthalten. Dabei können die Mengen auch gleich sein.
$A \subsetneq B$	echte Teilmenge	Alle Elemente aus A sind auch in B enthalten. Jedoch enthält B noch Elemente, die nicht in A enthalten sind. \implies Mengen sind nicht gleich!
$A \subset B$	Teilmenge <i>oder</i> echte Teilmenge	Bei manchen Leuten \subseteq , bei manchen \subsetneq . Mehrdeutig, lieber nicht verwenden!



Abk.	Bedeutung	Was?!
start \rightarrow 	Startzustand	Hier fängt der Automat beim Lesen eines Wortes an
	Zustandsübergang	gibt an, welches Symbol eingelesen werden kann, um in den Folgezustand zu übergehen.
	Endzustand	Hier kann ein fertig gelesenes Wort akzeptiert werden.
	Fangzustand	wird benötigt, um Determinismus zu gewährleisten. In Graphiken oft nicht eingezeichnet, ist aber da. Malt den hin.



Abk.	Bedeutung	Was?!
$T(M)$	Sprache von Automat M	Die Sprache, die von einem Automat M erkannt wird
$L(G)$	Sprache von Grammatik G	Die Sprache, die von einer Grammatik G erzeugt wird
γ	kleines Gamma	oft Bezeichner für regulären Ausdruck
$L(\gamma)$	Sprache von reg. Ausdruck γ	Die Sprache, die von einem regulären Ausdruck γ erkannt wird



Anhang



Beweis: DEA zu Grammatik



Satz

Jede durch endliche Automaten erkennbare Sprache ist auch regulär (also Typ 3).

Sei $A \subseteq \Sigma^*$ eine Sprache und M ein DEA mit $\mathbf{T}(M) = A$.
(d.h. M erkennt die Sprache A)

Wir suchen eine Typ 3-Grammatik G mit $\mathbf{L}(G) = A$.
(d.h. die Grammatik G erzeugt die Sprache A)

Anmerkung

Wir beschränken uns auf DEAs; in der Vorlesung werdet ihr aber eine allgemeinere Äquivalenz zeigen.



Also: Zustände $\hat{=}$ Variablen; Übergänge $\hat{=}$ Produktionen

Sei also ein DEA $M = (Z, \Sigma, \delta, z_0, E)$ gegeben.

Wir definieren die Grammatik $G = (V, \Sigma, P, S)$ mit:

- Alphabet Σ
- Variablenmenge $V = Z$
- Startsymbol $S = z_0$
- Produktionsmenge P erzeugen wir aus δ



DEA zu Grammatik: Produktionsregeln

Für die Produktionsmenge P wandeln wir die Übergänge um.

Jedem δ -Übergang $\delta(z_1, a) = z_2$ ordnen wir folgende Regeln zu:

- $z_1 \rightarrow az_2$
- Und zusätzlich, falls $z_2 \in E$:
 $z_1 \rightarrow a$

Falls $z_0 \in E$, brauchen wir außerdem
 $z_0 \rightarrow \varepsilon$.



„ $\delta(z_1, a) = z_2$ “
wird zu

$$z_1 \rightarrow az_2$$

$$z_1 \rightarrow a$$



„ $z_0 \in E$ “
wird zu

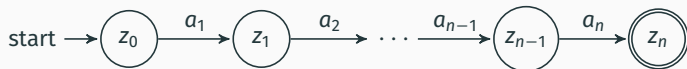
$$z_0 \rightarrow \varepsilon$$



DEA zu Grammatik: Korrektheit (Beweisidee)

Zu zeigen: $x \in T(M)$ gdw. $x \in L(G)$

Sei $x = a_1a_2 \dots a_{n-1}a_n$ ein Wort, das von M akzeptiert wird.



Mit den passenden Regeln (z.B.

$z_0 \rightarrow a_1z_1z_1 \rightarrow a_2z_2z_2 \rightarrow a_3z_3z_{n-2} \rightarrow a_{n-1}z_{n-1}z_{n-1} \rightarrow a_n$) lässt sich x ableiten:

$$z_0 \Rightarrow a_1z_1 \Rightarrow a_1a_2z_2 \Rightarrow \dots \Rightarrow a_1a_2 \dots a_{n-1}z_{n-1} \Rightarrow a_1a_2 \dots a_{n-1}a_n = x$$

Also können wir das Wort in der Grammatik ableiten.

Funktioniert die Argumentation auch andersrum? Ja!



Zu zeigen: $x \in T(M)$ **gdw.** $x \in L(G)$

Dabei gilt immer noch: $x = a_1 a_2 \dots a_{n-1} a_n$

Die folgenden Aussagen sind äquivalent:

- x wird von Automat M erkannt ($x \in T(M)$)
- Es gibt eine Folge von Zuständen $z_0, z_1, \dots, z_{n-1}, z_n$ mit: z_0 ist Startzustand, z_n ist Endzustand **und:** $\forall i \in \{1, \dots, n\} : \delta(z_{i-1}, a_i) = z_i$
- Es gibt Folge an Variablen z_0, z_1, \dots, z_{n-1} mit: z_0 ist Startvariable, $(z_{n-1} \rightarrow a_n) \in P$ **und:** $\forall i \in \{1, \dots, n-1\} : (z_{i-1} \rightarrow a_i z_i) \in P$
- Es gibt Folge an Variablen z_0, z_1, \dots, z_{n-1} mit: z_0 ist Startvariable **und:** $z_0 \Rightarrow a_1 z_1 \Rightarrow \dots \Rightarrow a_1 a_2 \dots a_{n-1} a_n$
- x wird von der Grammatik G produziert ($x \in L(G)$)

Also gilt die Äquivalenz.



- Unsere Folien sind frei!
- Jeder darf die Folien unter den Bedingungen der **GNU General Public License v3** (oder jeder späteren Version) weiterverwenden.
- Ihr findet den Quelltext unter
<https://www.github.com/FIUS/theo-vorkurs-folien>

