Vorkurs Theoretische Informatik

Einführung in reguläre Sprachen

Arbeitskreis Theoretische Informatik

Freitag, 05.10.2018

Fachgruppe Informatik

Übersicht

Chomsky-Hierachie

Manche Sprachen sind schwerer zu beschreiben als andere

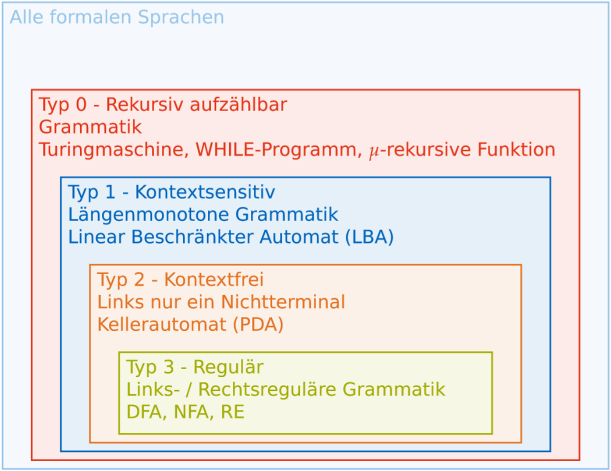
Wenn wir unsere Grammatiken einschränken, können wir nicht mehr alle Sprachen beschreiben.

Beispiel

Mit der Einschränkung
Alle Produktionsregeln müssen der Form $A \to a$ oder $A \to aB$ entsprechen, wobei $A, B \in V$ und $a \in \Sigma$.
können wir Sprachen wie $L_1 = \{a^n \mid n \in \mathbb{N}\}$ beschreiben,
aber nicht mehr Sprachen wie $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$.
Achtung: ist $\varepsilon \in L$, ist auch $S \to \varepsilon$ erlaubt, sofern S nicht auf der rechten Seite einer Produktion vorkommt.

$\leadsto$ Sprachen, die wir mit dieser starken Einschränkung beschreiben können, nennen wir *regulär* oder vom *Typ 3*.
Es gibt weitere Typen $\leadsto$ Mehr dazu in der Vorlesung

Manche Sprachen sind schwerer zu beschreiben als andere



Reguläre Grammatik

Aufgaben

Finde eine reguläre Grammatik für die folgenden Sprachen

Normal

- $L_1 = \{a^{2n} \mid n \in \mathbb{N}\}$
- $L_2 = \{a^n b^m \mid n, m \in \mathbb{N}\}$
- $L_3 = \{uv \mid u \in \{a, b\}^*,\ v \in \{c, d\}\}$
- $L_4 = \{w \mid |w| = 3, w \in \{a, b, c\}^*\}$

Etwas Schwerer

- $L_5 = \{a^n \mid n \equiv 1 \mod 3\}$
- $L_6 = \{uv \mid u \in \{◀, ▲, ▶, ▼\}^*,\ v \in \{\text{STOP}\}\}$
- $L_7 = \{w \in \{a, b, c\}^* \mid |w|_a = 3, |w|_b = 1\}$

Lösung

- $P_1 = \{S \to aA \mid \varepsilon,\ A \to aB \mid a,\ B \to aA\}$

Lösung

- $P_1 = \{S \to aA \mid \varepsilon,\ A \to aB \mid a,\ B \to aA\}$
- $P_2 = \{S \to aA \mid bB \mid b \mid a \mid \varepsilon,\ A \to aA \mid bB \mid b \mid a,\ B \to bB \mid b\}$

Lösung

- $P_1 = \{S \rightarrow aA \mid \varepsilon,\ A \rightarrow aB \mid a,\ B \rightarrow aA\}$
- $P_2 = \{S \rightarrow aA \mid bB \mid b \mid a \mid \varepsilon,\ A \rightarrow aA \mid bB \mid b \mid a,\ B \rightarrow bB \mid b\}$
- $P_3 = \{S \rightarrow aS \mid bS \mid c \mid d\}$

Lösung

- $P_1 = \{S \to aA \mid \varepsilon,\ A \to aB \mid a,\ B \to aA\}$
- $P_2 = \{S \to aA \mid bB \mid b \mid a \mid \varepsilon,\ A \to aA \mid bB \mid b \mid a,\ B \to bB \mid b\}$
- $P_3 = \{S \to aS \mid bS \mid c \mid d\}$
- $P_4 = \{S \to aA \mid bA \mid cA,\ A \to aB \mid bB \mid cB,\ B \to a \mid b \mid c\}$

Lösung

- $P_1 = \{S \to aA \mid \varepsilon,\ A \to aB \mid a,\ B \to aA\}$
- $P_2 = \{S \to aA \mid bB \mid b \mid a \mid \varepsilon,\ A \to aA \mid bB \mid b \mid a,\ B \to bB \mid b\}$
- $P_3 = \{S \to aS \mid bS \mid c \mid d\}$
- $P_4 = \{S \to aA \mid bA \mid cA,\ A \to aB \mid bB \mid cB,\ B \to a \mid b \mid c\}$
- $P_5 = \{S \to aA \mid a,\ A \to aB,\ B \to aS\}$

Lösung

- $P_1 = \{S \to aA \mid \varepsilon,\ A \to aB \mid a,\ B \to aA\}$
- $P_2 = \{S \to aA \mid bB \mid b \mid a \mid \varepsilon,\ A \to aA \mid bB \mid b \mid a,\ B \to bB \mid b\}$
- $P_3 = \{S \to aS \mid bS \mid c \mid d\}$
- $P_4 = \{S \to aA \mid bA \mid cA,\ A \to aB \mid bB \mid cB,\ B \to a \mid b \mid c\}$
- $P_5 = \{S \to aA \mid a,\ A \to aB,\ B \to aS\}$
- $P_6 = \{S \to \blacktriangleleft S \mid \blacktriangle S \mid \blacktriangleright S \mid \blacktriangledown S \mid \text{STOP}\}$

Lösung

- $$\begin{aligned} P_7 = \{S &\to cS \mid aA_1 \mid bB_0, \\ A_1 &\to cA_1 \mid aA_2 \mid bB_1, \\ A_2 &\to cA_2 \mid aA_3 \mid bB_2, \\ A_3 &\to cA_3 \mid bB_3 \mid b, \\ B_0 &\to cB_0 \mid aB_1, \\ B_1 &\to cB_1 \mid aB_2, \\ B_2 &\to cB_2 \mid aB_3 \mid a, \\ B_3 &\to cB_3 \mid cC \mid c, \\ C &\to cC \mid c\} \end{aligned}$$

Murmelpause

Automaten

Reguläre Sprachen anders beschreiben

Wir können reguläre Sprachen auch graphisch beschreiben.
Dafür nutzen wir endliche Automaten.
Ein Automat prüft Wörter und entscheidet, ob sie Teil der Sprache sind oder nicht.
$\rightsquigarrow$ Wir nennen das *akzeptieren*, bzw. nicht akzeptieren.

Funktionsweise

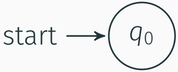
1. Ein Wort wird in den Automat eingegeben
2. Wort wird zeichenweise abgearbeitet
3. Nach jedem Zeichen wird der Automat in einen Zustand überführt, der bestimmt, wie fortgefahren wird
4. Befindet sich der Automat in einem *Endzustand*, sobald das Wort abgearbeitet wurde, akzeptiert der Automat das Wort.

Bestandteile eines endlichen Automaten

Der Automat kann als gerichteter Graph notiert werden.
Wir konstruieren ihn aus den folgenden Komponenten:

Startzustand

Im Startzustand wird das Wort eingegeben.

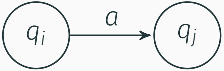


Bestandteile eines endlichen Automaten

Der Automat kann als gerichteter Graph notiert werden.
Wir konstruieren ihn aus den folgenden Komponenten:

Zustandsübergang

Wird das Zeichen auf dem Übergang *gelesen*, geht der Automat in den folgenden Zustand über.



Bestandteile eines endlichen Automaten

Der Automat kann als gerichteter Graph notiert werden.
Wir konstruieren ihn aus den folgenden Komponenten:

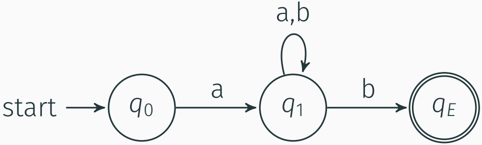
Endzustand

Falls sich der Automat in diesem Zustand befindet, und das Wort abgearbeitet ist, wird das Wort akzeptiert.

Anmerkung: Unter Umständen sind mehrere hiervon nötig

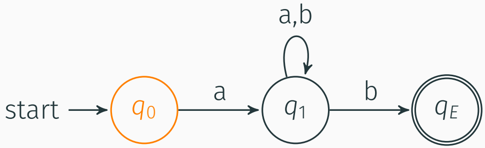
Beispiel

$L = \{axb \mid x \in \{a, b\}^*\}$



Beispiel

$L = \{axb \mid x \in \{a, b\}^*\}$

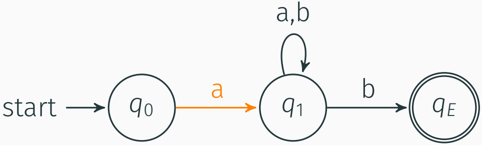


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a, b\}^*\}$

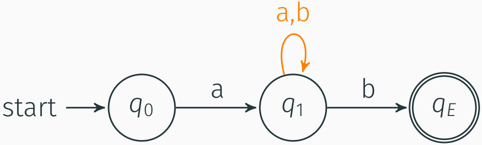


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a, b\}^*\}$

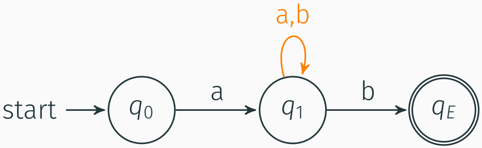


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a, b\}^*\}$

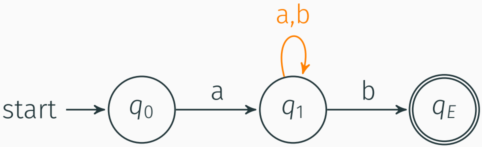


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a,b\}^*\}$

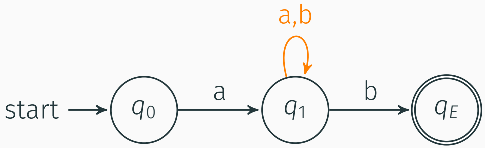


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a,b\}^*\}$

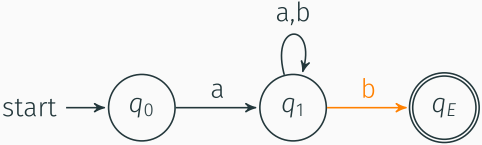


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a, b\}^*\}$

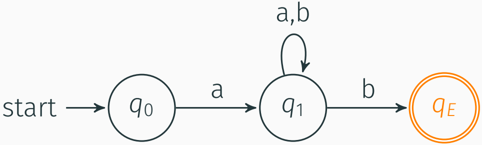


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a,b\}^*\}$



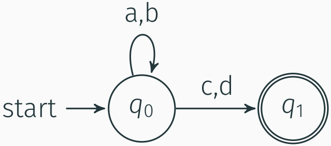
Worteingabe:

aababb $\in L \rightsquigarrow$ akzeptiert

Beispiel: Automat

Gegeben ist eine Sprache L. Gesucht ist ein Automat M, der **genau** die Wörter aus L akzeptiert.

$L_1 = \{uv \mid u \in \{a,b\}^*,\ v \in \{c,d\}\}$



Denkpause

knifflige Aufgabe

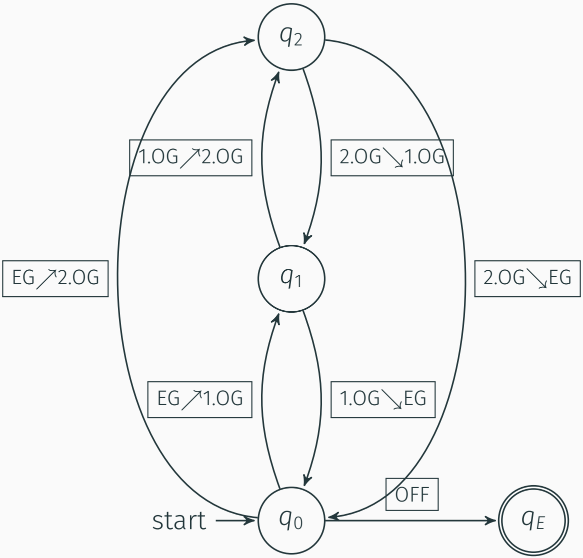
Wir entwerfen einen Automat zur Aufzugskontrolle. Der Aufzug hat folgende Möglichkeiten:

$$\Sigma = \{ \boxed{\text{EG}\nearrow\text{1.OG}}, \boxed{\text{EG}\nearrow\text{2.OG}}, \boxed{\text{1.OG}\searrow\text{EG}}, \boxed{\text{1.OG}\nearrow\text{2.OG}}, \boxed{\text{2.OG}\searrow\text{EG}}, \boxed{\text{2.OG}\searrow\text{1.OG}}, \boxed{\text{OFF}} \}$$

- Der Aufzug startet vom Erdgeschoss und darf sich nur aus Stockwerken bewegen, in denen er sich befindet.
- Der Aufzug kann nur im Erdgeschoss ausgeschaltet werden. Er kann dann keine Bewegung durchführen. Er muss ausgeschaltet werden.

Zeichne einen Automaten an, dessen akzeptierte Sprache genau die Menge der korrekten Abläufe ist.

Lösung



Aufgaben

Finde Automaten, die **genau** folgende Sprachen erkennen.

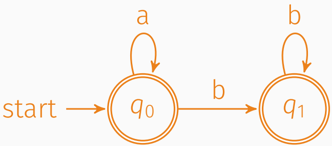
Normal

- $L_1 = \{a^n b^m \mid n, m \in \mathbb{N}\}$
- $L_2 = \{w \mid |w| = 3, w \in \{a, b, c\}^*\}$
- $L_3 = \{uv \mid u \in \{◀, ▲, ▶, ▼\}^*,\ v \in \{\text{STOP}\}\}$

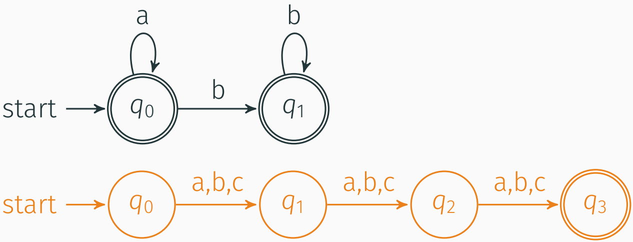
Etwas Schwerer

- $L_4 = \{a^n \mid n \equiv 1 \bmod 3\}$
- $L_5 = \{w \in \{a, b, c\}^* \mid |w|_a = 3, |w|_b = 1\}$
- $L_6 = \{w \in \{a, b\}^* \mid |w|_a \equiv |w|_b \bmod 3\}$

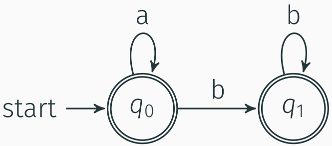
Lösung

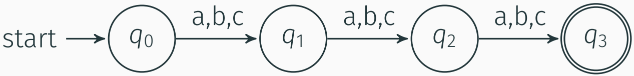


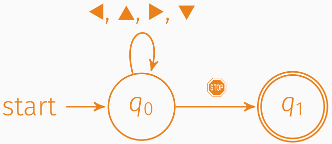
Lösung

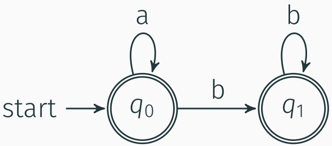


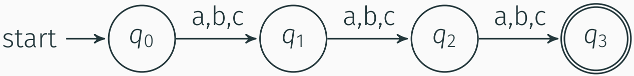
Lösung

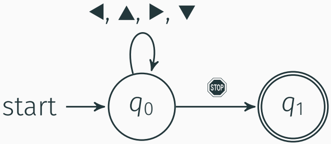


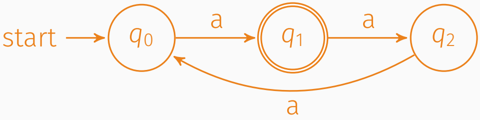




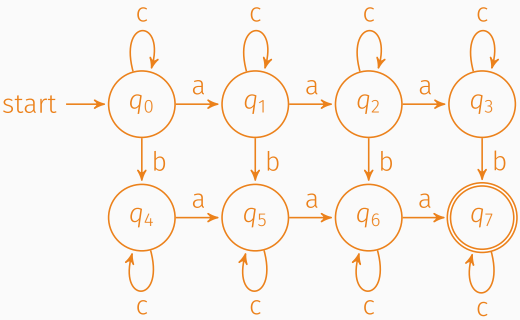

a
b
b
start
q_0
q_1


start
q_0
a,b,c
q_1
a,b,c
q_2
a,b,c
q_3

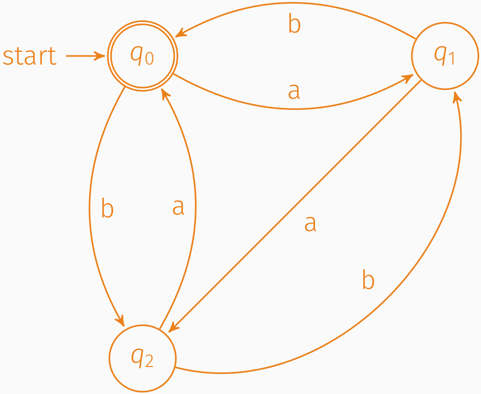

◀, ▲, ▶, ▼
STOP
start
q_0
q_1


start
q_0
a
q_1
a
q_2
a

Lösung



Lösung



Murmelpause

Verschiedene endliche Automaten

NEA

Beim lesen eines Wortes ist es manchmal unklar, welchen Übergang der Automat nehmen soll.
$\rightsquigarrow$ Der Automat ist *nichtdeterministisch*.

DEA

Wir können unsere Möglichkeiten so einschränken, dass bei jedem Zeichen eindeutig ist, welcher Übergang genutzt wird.
$\rightsquigarrow$ Der Automat ist *deterministisch*.

Wir beschränken unseren Automaten folgendermaßen:

Von jedem Zustand muss genau ein Übergang für jedes $a \in \Sigma$ ausgehen.

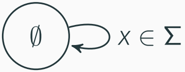
Wir beschränken unseren Automaten folgendermaßen:

Von jedem Zustand muss genau ein Übergang für jedes $a \in \Sigma$ ausgehen.

Um dies zu ermöglichen führen wir eine neue Komponente ein:

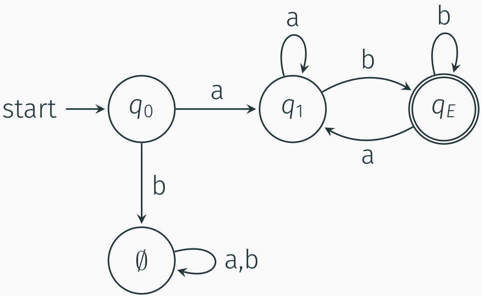
Fangzustand

Dieser Zustand kann nicht verlassen werden.
Falls der Automat in diesem Zustand landet, kommt er nicht mehr raus. Das Wort kann nicht akzeptiert werden.



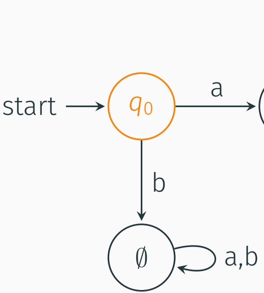
Beispiel

$L = \{axb \mid x \in \{a,b\}^*\}$



Beispiel

$L = \{axb \mid x \in \{a, b\}^*\}$

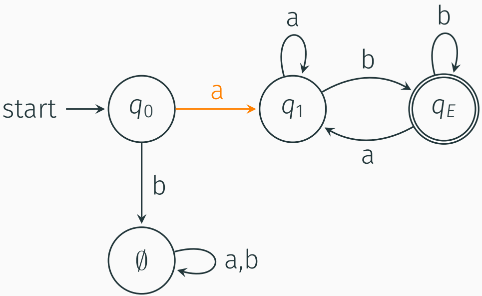


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a, b\}^*\}$

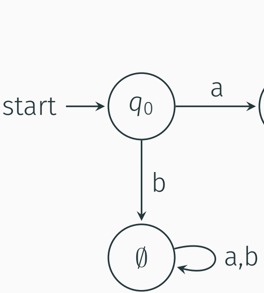


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a,b\}^*\}$

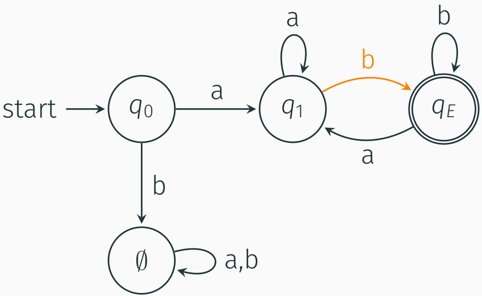


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a, b\}^*\}$

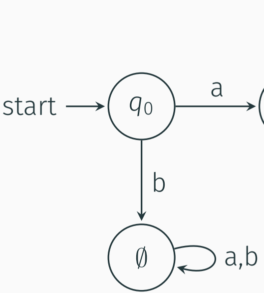


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a,b\}^*\}$

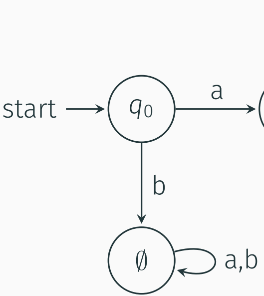


Worteingabe:

aababb $\in L$?

Beispiel

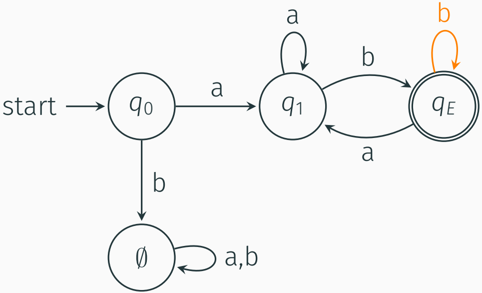
$L = \{axb \mid x \in \{a, b\}^*\}$



Worteingabe:

aababb $\in L$?

$L = \{axb \mid x \in \{a,b\}^*\}$

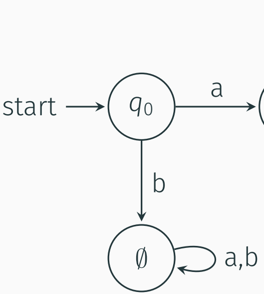


Worteingabe:

aababb $\in L$?

Beispiel

$L = \{axb \mid x \in \{a,b\}^*\}$



Worteingabe:

aababb $\in L \rightsquigarrow$ akzeptiert

Aufgaben

Finde deterministische endliche Automaten (DEAs) für die folgenden Sprachen.

Normal

- $L_1 = \{a^{2n} \mid n \in \mathbb{N}\}$ über $\Sigma = \{a\}$
- $L_2 = \{w \in \{a,b\}^* \mid |w|_a \geq 2\}$ über $\Sigma = \{a,b\}$

Prüfungsaufgabe: Etwas Schwerer

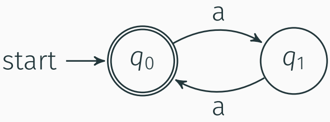
- $L_3 = \{w \in \{a,b,c\}^* \mid \ |w|_b \geq 1$ und $aaca$ ist Suffix von w$\}$ über $\Sigma = \{a,b,c\}$

Prüfungsaufgabe: Schwer

- $L_4 = \{bin(n) \mid \exists k \in \mathbb{N} : n = 4k, bin(n)$ ist Binärdarstellung von $n\}$ über $\Sigma = \{1,0\}$
  Achtung: Keine führenden Nullen.

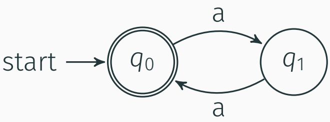
Lösung: Normal

$L_1 = \{a^{2n} \mid n \in \mathbb{N}\}$

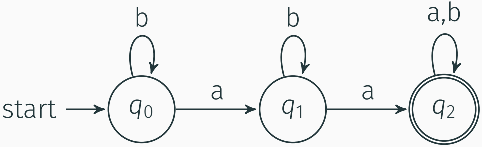


Lösung: Normal

$L_1 = \{a^{2n} \mid n \in \mathbb{N}\}$

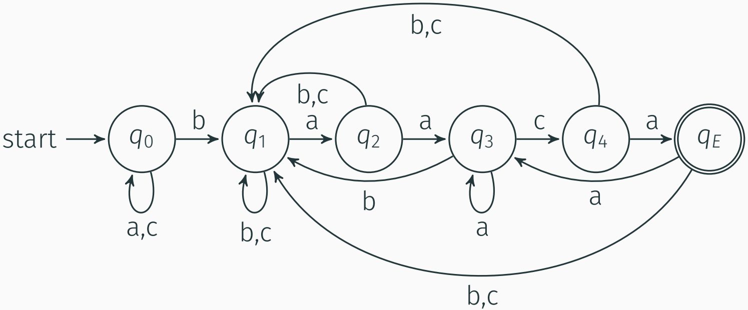


$L_2 = \{w \in \{a,b\}^* \mid |w|_a \geq 2\}$



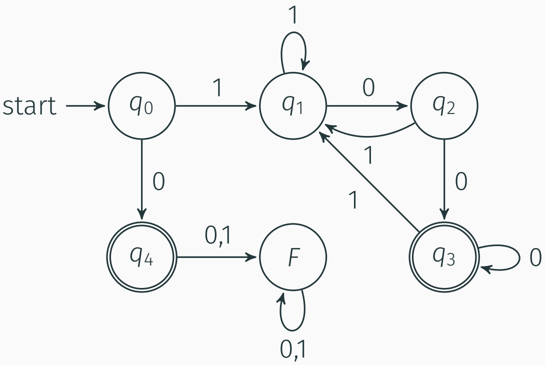
Lösung: Prüfungsaufgabe: Etwas Schwerer

$L_3 = \{w \in \{a, b, c\}^* \mid |w|_b \geq 1$ **und** $aaca$ **ist Suffix von w**$\}$



Lösung: Prüfungsaufgabe: Schwer

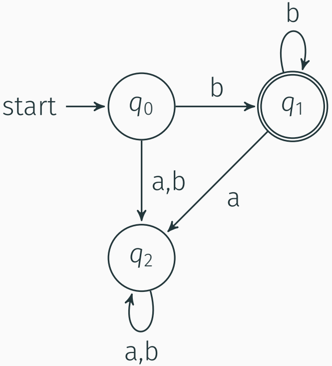
$L_4 = \{bin(n) \mid \exists k \in \mathbb{N} : n = 4k, bin(n)$ ist Binärdarstellung von $n\}$



Scheinklausuraufgabe WS17/18

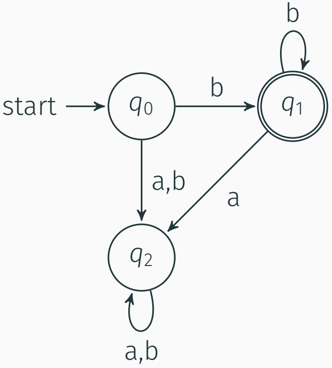
Gegeben sei folgender NEA M:

Nenne Wörter die erkannt werden.



Gegeben sei folgender NEA M:

Nenne Wörter die erkannt werden.



$\rightsquigarrow$ b, bb, bbb, bbbb, bbbbb,...

Murmelpause

Grammatik und Automaten

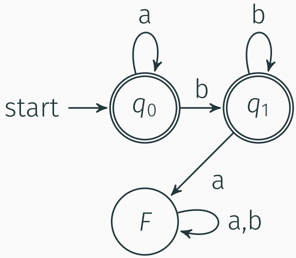
Automaten: Formal

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 Überführungsfunktion
- z_0: Der Startzustand
- E: Die Menge der Endzustände

Automaten: Formal

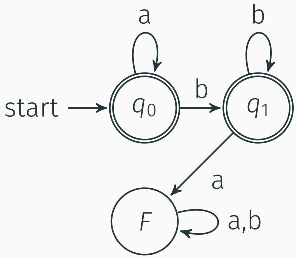
$L_1 = \{a^n b^m \mid n, m \in \mathbb{N}\}$



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

Automaten: Formal

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

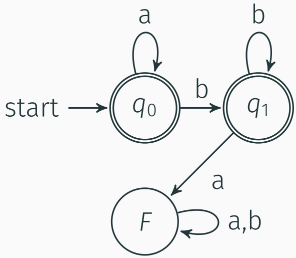


$M = (Z, \Sigma, \delta, q_0, E)$ mit:

- $Z = \{q_0, q_1, F\}$

Automaten: Formal

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

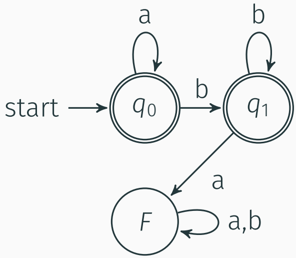


$M = (Z, \Sigma, \delta, q_0, E)$ mit:

- $Z = \{q_0, q_1, F\}$
- $\Sigma = \{a, b\}$

Automaten: Formal

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

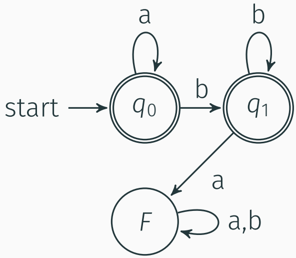


$M = (Z, \Sigma, \delta, q_0, E)$ mit:

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

Automaten: Formal

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



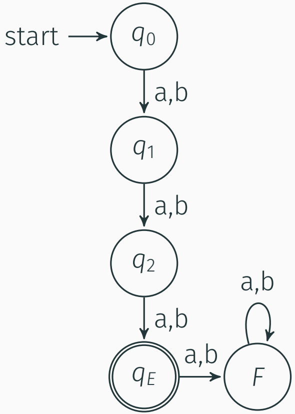
$M = (Z, \Sigma, \delta, q_0, E)$ mit:

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

Kurz selbst denken...

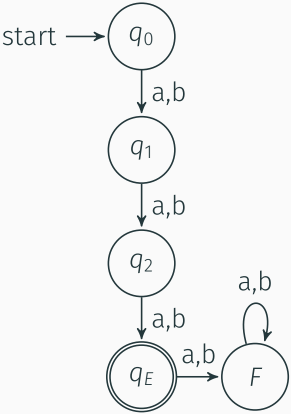
$L_2 = \{w \in \{a,b\}^* \mid |w| = 3\}$

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



Kurz selbst denken...

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

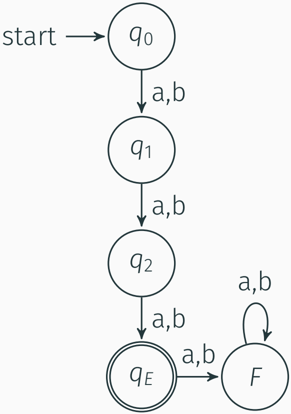


$M = (Z, \Sigma, \delta, q_0, E)$ mit:

- $Z = \{q_0, q_1, q_2, q_E, F\}$

Kurz selbst denken...

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

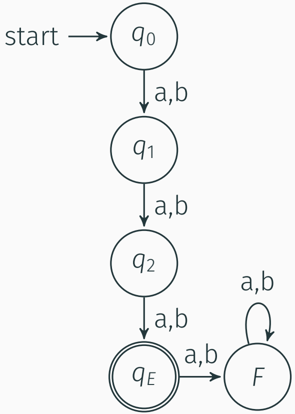


$M = (Z, \Sigma, \delta, q_0, E)$ mit:

- $Z = \{q_0, q_1, q_2, q_E, F\}$
- $\Sigma = \{a, b\}$

Kurz selbst denken...

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

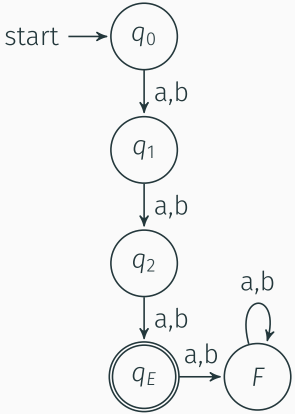


$M = (Z, \Sigma, \delta, q_0, E)$ mit:

- $Z = \{q_0, q_1, q_2, q_E, F\}$
- $\Sigma = \{a, b\}$
- δ:
  - $\delta(q_0, a) = q_1$
  - $\delta(q_0, b) = q_1$
  - $\delta(q_1, a) = q_2$
  - $\delta(q_1, b) = q_2$
  - $\delta(q_2, a) = q_3$
  - $\delta(q_2, b) = q_3$
  - $\delta(q_E, a) = F$
  - $\delta(q_E, b) = F$
  - $\delta(F, a) = F$
  - $\delta(F, b) = F$

Kurz selbst denken...

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



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

- $Z = \{q_0, q_1, q_2, q_E, F\}$
- $\Sigma = \{a, b\}$
- δ:
  - $\delta(q_0, a) = q_1$
  - $\delta(q_0, b) = q_1$
  - $\delta(q_1, a) = q_2$
  - $\delta(q_1, b) = q_2$
  - $\delta(q_2, a) = q_3$
  - $\delta(q_2, b) = q_3$
  - $\delta(q_E, a) = F$
  - $\delta(q_E, b) = F$
  - $\delta(F, a) = F$
  - $\delta(F, b) = F$
- $E = \{q_E\}$

Automaten und Grammatiken

Satz

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

Sei $A \subseteq \Sigma^*$ eine Sprache und M ein Automat mit **T(M) = A**, *(d.h. M erkennt die Sprache A)*.

Wir definieren eine Typ 3-Grammatik G mit **L(G)=A**, *(d.h. die Grammatik G erzeugt die Sprache A)*.

Es ist $G = (V, \Sigma, P, S)$ mit:
V = Menge der Zustände des Automaten (Z)
S = Startzustand des Automaten (z_0)

Falls $\varepsilon \in A$, dann enthält P die Regel „$z_0 \to \varepsilon$“

Unsere Menge der Produktionsregeln P besteht aus folgenden Regeln:

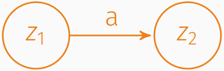
Jeder "δ-Anweisung" $\delta(z_1, a) = z_2$ ordnen wir folgende Regeln zu.

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

Unsere Menge der Produktionsregeln P besteht aus folgenden Regeln:

Jeder "δ-Anweisung"$\delta(z_1, a) = z_2$ ordnen wir folgende Regeln zu.

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

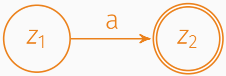


„$\delta(z_1, a) = z_2$“

Unsere Menge der Produktionsregeln P besteht aus folgenden Regeln:

Jeder "δ-Anweisung"$\delta(z_1, a) = z_2$ ordnen wir folgende Regeln zu.

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



„$\delta(z_1, a) = z_2$“

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

Dabei gilt: $x = a_1 a_2 ... a_n$

Die folgenden Aussagen sind äquivalent:

- x wird von Automat M erkannt ($x \in T(M)$)
- Es gibt eine Folge von Zuständen $z_0, z_1, \ldots, z_n$ mit: z_0 ist Startzustand, z_n ist Endzustand **und**:
  $\forall i \in \{1, ..., n\} : \delta(z_{i-1}, a_i) = z_i$
- Es gibt Folge an Variablen $z_0, z_1, \ldots, z_n$ mit: z_0 ist Startvariable und x lässt sich von z_0 ausgehend ableiten.
- x wird von der Grammatik G produziert ($x \in L(G)$)

$\square$

Murmelpause

Reguläre Ausdrücke

mehr Möglichkeiten reguläre Sprachen zu beschreiben

Graphen und „Bilder“ 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.

Reguläre Ausdrücke verwenden

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.

Beispiel

$\gamma = ((a|b)^* \mid \varepsilon) \implies aba \in L(\gamma)$

Wie Sprachen und reguläre Ausdrücke zusammenhängen

- Wenn $\gamma = \emptyset$, beschreibt es die leere Sprache: $L(\gamma) = \{\}$

Reguläre Ausdrücke verwenden

Wie Sprachen und reguläre Ausdrücke zusammenhängen

- Wenn $\gamma = \emptyset$, beschreibt es die leere Sprache: $L(\gamma) = \{\}$
- Wenn γ ein einzelnes Wort ist, ist genau dieses Wort in der Sprache enthalten.
  $\gamma = \varepsilon$: $L(\gamma) = \{\varepsilon\}$, $\quad \gamma = a$: $L(\gamma) = \{a\}$

Wie Sprachen und reguläre Ausdrücke zusammenhängen

- Wenn $\gamma = \emptyset$, beschreibt es die leere Sprache: $L(\gamma) = \{\}$
- Wenn γ ein einzelnes Wort ist, ist genau dieses Wort in der Sprache enthalten.
  $\gamma = \varepsilon$: $L(\gamma) = \{\varepsilon\}$, $\quad \gamma = a$: $L(\gamma) = \{a\}$
- Wenn γ aus zwei hintereinandergeschrieben Ausdrücken besteht, repräsentiert Konkatenation.
  $\gamma = (a)^*(b)^* : L(\gamma) = \{a^n b^m | n, m \in \mathbb{N}\}$

Wie Sprachen und reguläre Ausdrücke zusammenhängen

- Wenn $\gamma = \emptyset$, beschreibt es die leere Sprache: $L(\gamma) = \{\}$
- Wenn γ ein einzelnes Wort ist, ist genau dieses Wort in der Sprache enthalten.
  $\gamma = \varepsilon$: $L(\gamma) = \{\varepsilon\}$, $\quad \gamma = a$: $L(\gamma) = \{a\}$
- Wenn γ aus zwei hintereinandergeschrieben Ausdrücken besteht, repräsentiert Konkatenation.
  $\gamma = (a)^*(b)^* : L(\gamma) = \{a^n b^m | n, m \in \mathbb{N}\}$
- Wenn γ aus zwei mit „oder“ verknüpften Ausdrücken besteht, sind beide Seiten in der Sprache enthalten.
  $\gamma = (a \mid bc)$: $L(\gamma) = \{a, bc\}$

Wie Sprachen und reguläre Ausdrücke zusammenhängen

- Wenn $\gamma = \emptyset$, beschreibt es die leere Sprache: $L(\gamma) = \{\}$
- Wenn γ ein einzelnes Wort ist, ist genau dieses Wort in der Sprache enthalten.
  $\gamma = \varepsilon$: $L(\gamma) = \{\varepsilon\}$, $\quad \gamma = a$: $L(\gamma) = \{a\}$
- Wenn γ aus zwei hintereinandergeschrieben Ausdrücken besteht, repräsentiert Konkatenation.
  $\gamma = (a)^*(b)^* : L(\gamma) = \{a^n b^m | n, m \in \mathbb{N}\}$
- Wenn γ aus zwei mit „oder“ verknüpften Ausdrücken besteht, sind beide Seiten in der Sprache enthalten.
  $\gamma = (a \mid bc)$: $L(\gamma) = \{a, bc\}$
- Wenn γ ein Ausdruck mit einem Stern ist, kann dieser innere Ausdruck beliebig oft wiederholt werden (auch null mal).
  $\gamma = (a)^*$: $L(\gamma) = \{\varepsilon, a, aa, aaa, ...\} = \{a\}^*$

Wie Sprachen und reguläre Ausdrücke zusammenhängen

- Wenn $\gamma = \emptyset$, beschreibt es die leere Sprache: $L(\gamma) = \{\}$
- Wenn γ ein einzelnes Wort ist, ist genau dieses Wort in der Sprache enthalten.
  $\gamma = \varepsilon$: $L(\gamma) = \{\varepsilon\}$, $\quad \gamma = a$: $L(\gamma) = \{a\}$
- Wenn γ aus zwei hintereinandergeschrieben Ausdrücken besteht, repräsentiert Konkatenation.
  $\gamma = (a)^*(b)^* : L(\gamma) = \{a^n b^m | n, m \in \mathbb{N}\}$
- Wenn γ aus zwei mit „oder" verknüpften Ausdrücken besteht, sind beide Seiten in der Sprache enthalten.
  $\gamma = (a \mid bc)$: $L(\gamma) = \{a, bc\}$
- Wenn γ ein Ausdruck mit einem Stern ist, kann dieser innere Ausdruck beliebig oft wiederholt werden (auch null mal).
  $\gamma = (a)^*$: $L(\gamma) = \{\varepsilon, a, aa, aaa, ...\} = \{a\}^*$
- Alles zusammen — $\gamma = ((a)^* \mid (bc)^*)$:
  $L(\gamma) = \{\varepsilon, a, bc, aa, bcbc, aaa, ...\} = \{a\}^* \cup \{bc\}^*$

Reguläre Ausdrücke

Aufgaben

Finde einen regulären Ausdruck 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

- $L(\gamma_5) = \{a^n \mid n \equiv 1 \mod 3\}$
- $L(\gamma_6) = \{uv \mid u \in \{◀, ▲, ▶, ▼\}^*,\ v \in \{\text{🛑}\}\}$
- $L(\gamma_7) = \{w \mid |w|_a = 3, |w|_b = 1, w \in \{a, b, c\}^*\}$

Lösung

Alle Lösungen sind Beispiellösungen, es sind auch andere möglich. Klammern die nicht zur Bedeutung beitragen, dürfen wir für die Kurzschreibweise weglassen.

- $\gamma_1 = (aa)^*$

Lösung

Alle Lösungen sind Beispiellösungen, es sind auch andere möglich. Klammern die nicht zur Bedeutung beitragen, dürfen wir für die Kurzschreibweise weglassen.

- $\gamma_1 = (aa)^*$
- $\gamma_2 = (a)^*(b)^*$

Lösung

Alle Lösungen sind Beispiellösungen, es sind auch andere möglich. Klammern die nicht zur Bedeutung beitragen, dürfen wir für die Kurzschreibweise weglassen.

- $\gamma_1 = (aa)^*$
- $\gamma_2 = (a)^*(b)^*$
- $\gamma_3 = (a|b)^*\ (c|d)$

Lösung

Alle Lösungen sind Beispiellösungen, es sind auch andere möglich. Klammern die nicht zur Bedeutung beitragen, dürfen wir für die Kurzschreibweise weglassen.

- $\gamma_1 = (aa)^*$
- $\gamma_2 = (a)^*(b)^*$
- $\gamma_3 = (a|b)^*\ (c|d)$
- formal: $\gamma_4 = ((a|b)|c)((a|b)|c)((a|b)|c)$, kurz:
  $\gamma_4 = (a|b|c)(a|b|c)(a|b|c)$

Lösung

Alle Lösungen sind Beispiellösungen, es sind auch andere möglich. Klammern die nicht zur Bedeutung beitragen, dürfen wir für die Kurzschreibweise weglassen.

- $\gamma_1 = (aa)^*$
- $\gamma_2 = (a)^*(b)^*$
- $\gamma_3 = (a|b)^*\ (c|d)$
- formal: $\gamma_4 = ((a|b)|c)((a|b)|c)((a|b)|c)$, kurz: $\gamma_4 = (a|b|c)(a|b|c)(a|b|c)$
- $\gamma_5 = a(aaa)^*$

Lösung

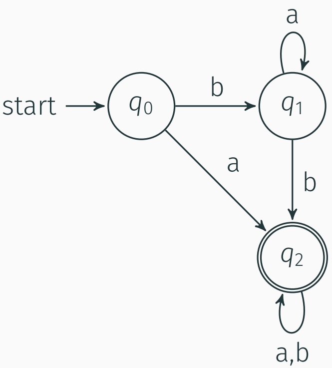
Alle Lösungen sind Beispiellösungen, es sind auch andere möglich. Klammern die nicht zur Bedeutung beitragen, dürfen wir für die Kurzschreibweise weglassen.

- $\gamma_1 = (aa)^*$
- $\gamma_2 = (a)^*(b)^*$
- $\gamma_3 = (a|b)^*\ (c|d)$
- formal: $\gamma_4 = ((a|b)|c)((a|b)|c)((a|b)|c)$, kurz: $\gamma_4 = (a|b|c)(a|b|c)(a|b|c)$
- $\gamma_5 = a(aaa)^*$
- formal: $\gamma_6 = (((\blacktriangleleft|\ \blacktriangle)\ |\ \blacktriangleright)\ |\ \blacktriangledown)^*$🛑, kurz: $\gamma_6 = (\blacktriangleleft|\ \blacktriangle|\ \blacktriangleright|\ \blacktriangledown)^*$🛑

Alle Lösungen sind Beispiellösungen, es sind auch andere möglich. Klammern die nicht zur Bedeutung beitragen, dürfen wir für die Kurzschreibweise weglassen.

- $\gamma_1 = (aa)^*$
- $\gamma_2 = (a)^*(b)^*$
- $\gamma_3 = (a|b)^*\ (c|d)$
- formal: $\gamma_4 = ((a|b)|c)((a|b)|c)((a|b)|c)$, kurz:
  $\gamma_4 = (a|b|c)(a|b|c)(a|b|c)$
- $\gamma_5 = a(aaa)^*$
- formal: $\gamma_6 = (((\blacktriangleleft|\ \blacktriangle)\ |\ \blacktriangleright)\ |\ \blacktriangledown)^*$🛑, kurz: $\gamma_6 = (\blacktriangleleft|\ \blacktriangle|\ \blacktriangleright|\ \blacktriangledown)^*$🛑
- formal: $\gamma_7 = (c)^*(a(c)^*a(c)^*a(c)^*b \mid a(c)^*a(c)^*b(c)^*a \mid a(c)^*b(c)^*a(c)^*a \mid b(c)^*a(c)^*a(c)^*a)(c)^*$,
  kurz:
  $\gamma_7 = c^*(ac^*ac^*ac^*b \mid ac^*ac^*bc^*a \mid ac^*bc^*ac^*a \mid bc^*ac^*ac^*a)c^*$

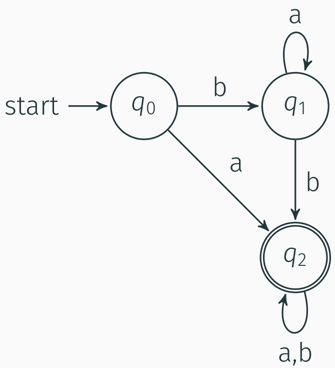
Gegeben sei folgender DEA M:



Welcher reguläre Ausdruck beschreibt $T(M)$?

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

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. $(a|b)^*$

Murmelpause

Das können wir jetzt beantworten

Mengen

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

Das können wir jetzt beantworten

Formale Sprachen

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

Das können wir jetzt beantworten

Beweise

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

Grammatiken

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

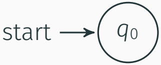
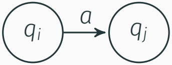
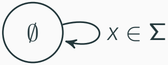
Das können wir jetzt beantworten

Automaten

- Was sind Automaten?
- Wie wandelt man Automaten zu einer äquivalenten Grammatik um?
- Was macht einen deterministischen Automaten aus?

Noch Fragen?

Glossar

Abk.	Bedeutung	Was?!
	Startzustand	Hier wird ein Wort eingegeben
	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 benotigt um Determinismus zu gewährleisten. In Graphiken oft nicht eingezeichnet, ist aber da. Malt den hin.

Glossar

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(γ)	Sprache von reg. Ausdruck γ	Die Sprache die von einem regulärem Ausdruck γ erkannt wird