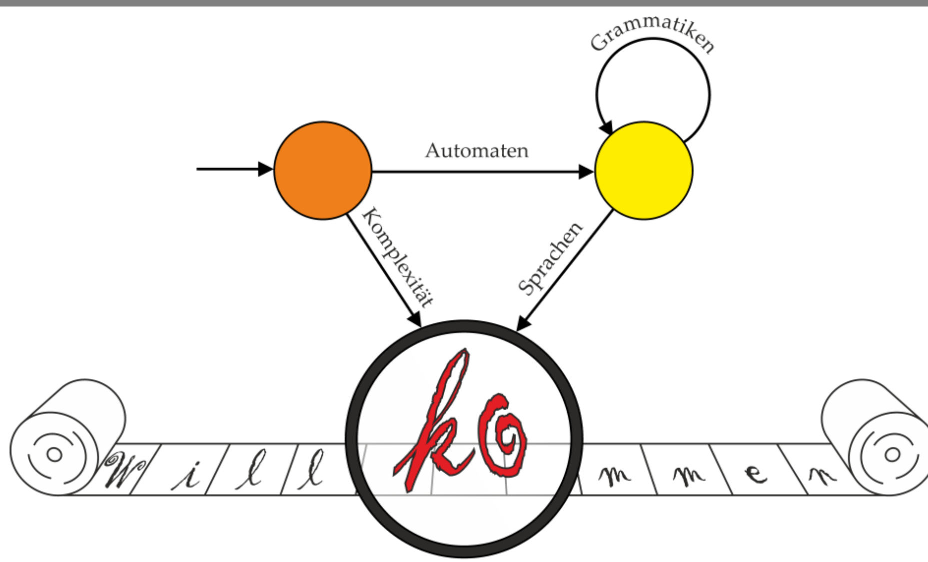
Karlsruhe Institute of Technology

Grundlagen der Informatik II

Tutorium 2

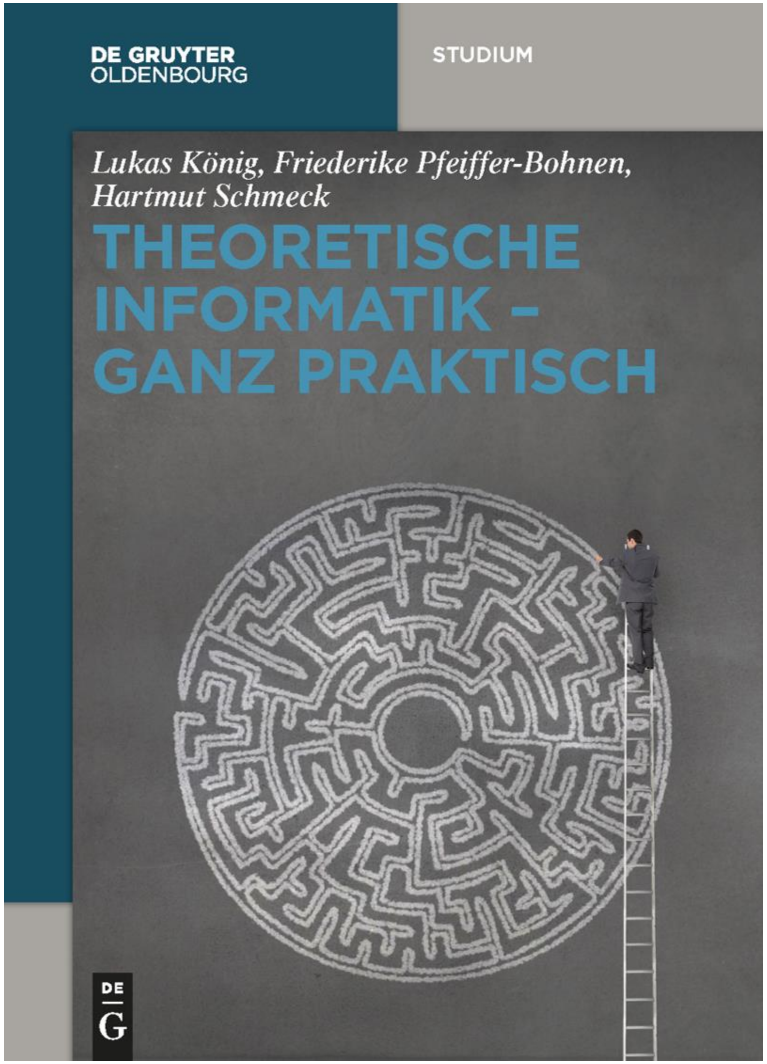
Institut für Angewandte Informatik und Formale Beschreibungsverfahren – Prof. Tatiana von Landesberger



Lehrbuch

- Erschienen im September 2016
- Einfache **praxisorientierte** Einführung in die schwierigen (bzw. oft „als schwierig empfundenen") theoretischen Inhalte
- Verknüpfung mit den Aufgaben des Aufgabenpools
- Verknüpfung mit dem XWizard
- **Viele Hinweise zu Verständnis und Klausurvorbereitung**

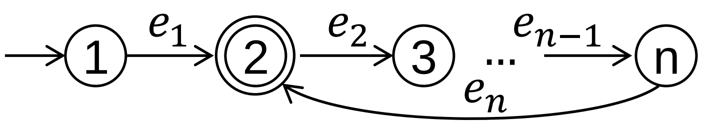
www.dasinfobuch.de



Einführungsaufgabe: Pumping Lemma (PPL)

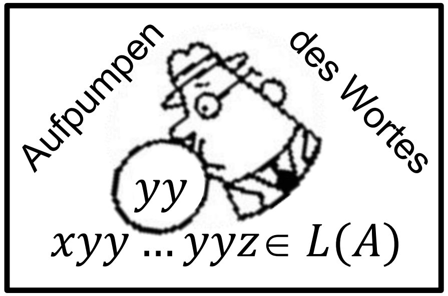
- Welche Idee liegt dem Pumping-Lemma (PPL) für EA-Sprachen zugrunde?
  - Ein endlicher Automat A mit n Zuständen muss, um Wörter mit mindestens n Zeichen zu erkennen, mindestens eine Schleife haben.

Beispiel:



- Was folgt daraus?
  - Dass es eine Zeichenfolge y in einem solchen Wort geben muss, die man „pumpen“ kann, also ein 0-fach, 1-fach, 2-fach,… schreiben, sodass der EA diese resultierenden Wörter auch akzeptiert:

    $xyz \in L(A) \Rightarrow xy^iz \in L(A)$ für alle $i \in \{0, 1, 2, 3, ...\}$



und…?

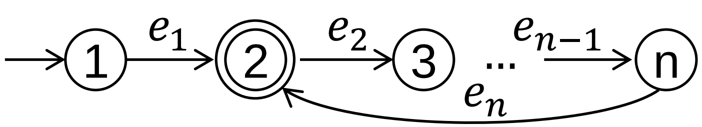
  - Da jede von einem endlichen Automaten erkannte Sprache diese Eigenschaft hat, kann man **das Fehlen dieser Eigenschaft** als Beweis ansehen, dass die Sprache nicht von einem endlichen Automaten erkannt werden kann. !

Ab S. 207

Einführungsaufgabe: PPL

- Welche Idee liegt dem Pumping-Lemma (PPL) für EA-Sprachen zugrunde?
  - Ein endlicher Automat A mit n Zuständen muss, um Wörter mit mindestens n Zeichen zu erkennen, mindestens eine Schleife haben.

Beispiel:

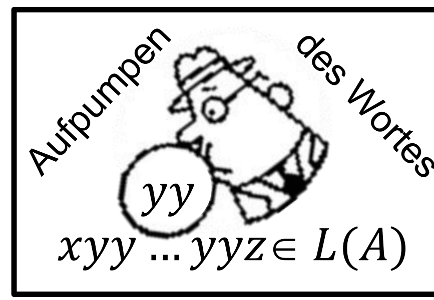


- Was folgt daraus?

Was ist y in dem Beispiel?

  - … eine Zeichenfolge v in einem solchen Wort geben muss, die man …“ kann, also 0-fach, 1-fach, 2-fach,… schreiben, sodass der EA …ultierenden Wörter auch akzeptiert:

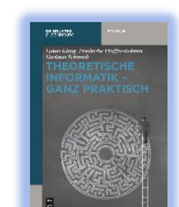
$xyz \in L(A) \Rightarrow xy^iz \in L(A)$ für alle $i \in \{0, 1, 2, 3, \dots\}$



und…?

  - Da jede von einem endlichen Automaten erkannte Sprache diese Eigenschaft hat, kann man **das Fehlen dieser Eigenschaft** als Beweis ansehen, dass die Sprache nicht von einem endlichen Automaten erkannt werden kann.

!



Ab S. 207

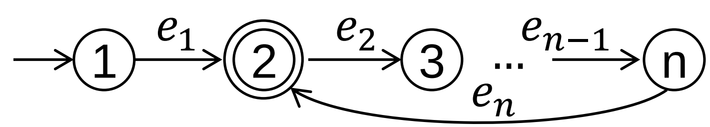
Einführungsaufgabe: PPL

- Welche Idee liegt dem Pumping-Lemma (PPL) für EA-Sprachen zugrunde?

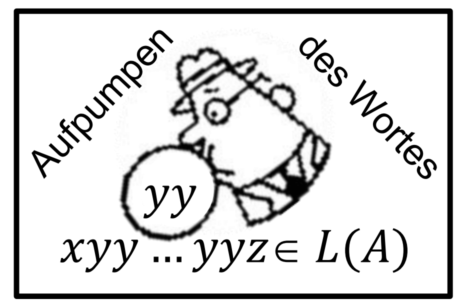
Die Zeichenfolge, welche durch einmaligen Durchgang der Schleife beginnend aus Zustand 2 entsteht.

…omat A mit n Zuständen muss, um Wörter mit mindestens …nnen, mindestens eine Schleife haben.

Beispiel:



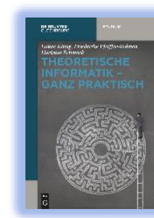
…chenfolge v in einem solchen Wort geben muss, die man …lso 0-fach, 1-fach, 2-fach,… schreiben, sodass der EA …nden Wörter auch akzeptiert:



$xyz \in L(A) \Rightarrow xy^iz \in L(A)$ für alle $i \in \{0, 1, 2, 3, \dots\}$

…d…?

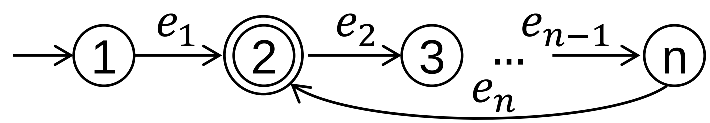
Da jede von einem endlichen Automaten erkannte Sprache diese Eigenschaft hat, kann man **das Fehlen dieser Eigenschaft** als Beweis ansehen, dass die Sprache nicht von einem endlichen Automaten erkannt werden kann. **!**



Ab S. 207

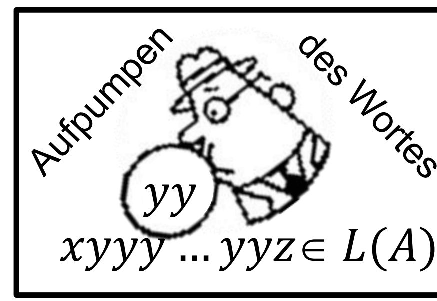
Einführungsaufgabe: PPL

- Welche Idee liegt dem Pumping-Lemma (PPL) für EA-Sprachen zugrunde?
  - Ein endlicher Automat A mit n Zuständen muss, um Wörter mit mindestens n Zeichen zu erkennen, mindestens eine Schleife haben.

Beispiel: 

- Was folgt daraus?
  - Dass es eine Zeichenfolge v in einem solchen Wort geben muss, die man … 0-fach, 1-fach, 2-fach,… schreiben, sodass der EA … Wörter auch akzeptiert:
    … $L(A)$ für alle $i \in \{0, 1, 2, 3, \dots\}$
  - …d…?

Die Schleife kann natürlich auch früher als bei Zustand *n* auftreten



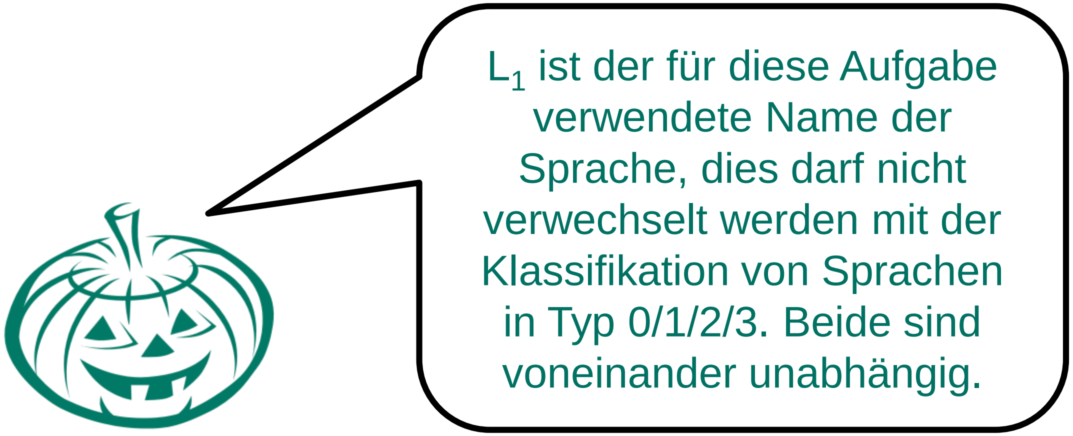
  - Da jede von einem endlichen Automaten erkannte Sprache diese Eigenschaft hat, kann man **das Fehlen dieser Eigenschaft** als Beweis ansehen, dass die Sprache nicht von einem endlichen Automaten erkannt werden kann. **!**

Ab S. 207

Aufgabe 2-1: Pumping-Lemma

Zeigen Sie mithilfe des Pumping-Lemmas (PPL), dass kein Endlicher Automat existiert, der L_1 bzw. L_2 erkennt.

(a) $L_1 = \{0^k 1^{2k} | \mathrm{k} \geq 1\}$



Aufgabe 2-1: Pumping-Lemma

Zeigen Sie mithilfe des Pumping-Lemmas (PPL), dass kein Endlicher Automat existiert, der L_1 bzw. L_2 erkennt.

(a) $L_1 = \{0^k 1^{2k} | \mathrm{k} \geq 1\}$

Lösung:

- Wähle ein beliebiges n. Dazu das Wort $w = 0^n 1^{2n} \in L_1$. Es ist $|w| \geq n$.
- Betrachte eine beliebige Partition von w bzw. x, y, z mit $w = xyz = 0^n 1^{2n}$ mit
  - *(1)* $|xy| \leq n$,
  - *(2)* $|y| \geq 1$,
  - (3) $\forall$ i $\in$ IN_0 : $xy^iz \in L_1$(A).
- Daraus folgt
  - $xy = 0^j$ mit $1 \leq j \leq n$, da $|xy| \leq n$ und $|y| \geq 1$,
  - $y = 0^k$ mit $1 \leq k \leq j$ und $x = 0^{j-k}$,
  - $z = 0^{n-j} 1^{2n}$.
- Wähle als Pumpvariable $i = 2$, dann ist $xy^2z = 0^{j-k} 0^{2k} 0^{n-j} 1^{2n} = 0^{n+k} 1^{2n} \notin L_1$.
- Demnach kann L_1 von keinem EA akzeptiert werden (es existiert also kein EA, der L_1 erkennt).

Aufgabe 2-1: Pumping-Lemma

(b) $L_2 = \{uu | u \in \{a, b\}^*\}$

Lösung:

- Wähle ein beliebiges n. Dazu das Wort $w = a^n b^n a^n b^n \in L_2$. Es ist $|w| \geq n$.
- Betrachte eine beliebige Partition von w bzw. x, y, z mit $w = xyz = a^n b^n a^n b^n$ mit

  (1) $|xy| \leq n$,

  (2) $|y| \geq 1$,

  (3) $\forall i \in \mathrm{IN}_0 : xy^i z \in L_2$ (A).
- Daraus folgt
  - $xy = a^j$ mit $1 \leq j \leq n$, da $|xy| \leq n$ und $|y| \geq 1$,
  - $y = a^k$ mit $1 \leq k \leq j$ und $x = a^{j-k}$,
  - $z = a^{n-j} b^n a^n b^n$.
- Wähle als Pumpvariable $i = 0$, dann ist $xy^0 z = xz = a^{j-k} a^{n-j} b^n a^n b^n = a^{n-k} b^n a^n b^n \notin L_2$.
- Demnach kann L_2 von keinem EA akzeptiert werden (es existiert also kein EA, der L_2 erkennt).

2-2: Single-Choice-Aufgabe

Mit dem Pumping-Lemma für EA-Sprachen kann für eine Sprache L bewiesen werden, dass ein endlicher Automat A existiert, mit $L = L(A)$.

☐ WAHR

☒ FALSCH

Erklärung:
Das PPL ist keine „Genau-Dann-Wenn"-Eigenschaft, sondern gilt **nur in eine Richtung**:

Wird eine Sprache von einem endlichen Automaten erkannt ⇒ gilt auch das PPL

Zu zeigen, dass das PPL für eine Sprache gilt, **heißt also nicht** zu beweisen, dass es einen EA für die Sprache gibt. Nur wenn es für eine Sprache **nicht** gilt, gibt es ganz sicher **keinen** EA für diese Sprache.

Einführungsaufgabe: Grammatiken

Was ist eine formale Grammatik?

- Formales System zur Beschreibung von Sprachen (erzeugend)
  - 4-Tupel: $G = (N, T, P, S)$
    - Welche Bedeutung haben die **Nonterminalsymbole in der Menge $\boldsymbol{N}$** und **Terminalsymbole in der Menge $\boldsymbol{T}$**?
      - Ergeben hintereinander geschrieben Symbolketten (Wörter).
      - Symbolketten (oder Teile davon) können durch Regeln/Produktionen aus P in andere Symbolketten überführt werden.
      - Eine abgeleitete Symbolkette w, die ausschließlich aus Terminal-symbolen besteht, heißt **Wort der Sprache** der Grammatik : $w \in L(G)$
    - Was für eine Besonderheit hat das **Nonterminalsymbol $\boldsymbol{S}$**?
      - Startsymbol: Mit diesem Symbol beginnt die Ableitung.
    - Was ist die **Regelmenge $\boldsymbol{P}$**?
      - Enthält Regeln der Form $\varphi \rightarrow \psi$, wobei φ und ψ Symbolketten sind.
- Beispiel:

$$N = \{S, A\}$$
$$T = \{a, b\}$$
$$P = \{\ S \rightarrow \lambda,\quad S \rightarrow SA,\quad A \rightarrow ab\}$$

Ableitung: $\boldsymbol{S \Rightarrow SA \Rightarrow SAA \Rightarrow SAAA \Rightarrow^* ababab \in L(G)}$

… …

Nonterminale

Terminale

Eine Grammatik **generiert** Wörter einer Sprache

Einführungsaufgabe: Grammatiken

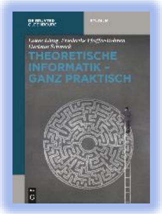
Was ist eine formale Grammatik?

- Formales System zur Beschreibung von Sprachen (erzeugend)
  - 4-Tupel: $G = (N, T, P, S)$
    - Welche Bedeutung haben die **Nonterminalsymbole in der Menge N** und **Terminalsymbole in der Menge T**?
      - Ergeben hintereinander geschrieben Symbolketten (Wörter).
      - Symbolketten (oder Teile davon) können durch Regeln/Produktionen aus P in andere Symbolketten überführt werden.
      - Eine abgeleitete Symbolkette w, die ausschließlich aus Terminal-symbolen besteht, heißt **Wort der Sprache** der Grammatik : $w \in L(G)$
    - Was für eine Besonderheit hat das **Nonterminalsymbol S**?
      - S
    - Was ist
      - E … en sind.
- Beispiel: $N =$ … $\boldsymbol{babab} \in \boldsymbol{L(G)}$

$T =$ … $\cdots$

$=$ … minale

Unter diesem Symbol finden Sie einen Verweis zu diesem Thema im neuen Lehrbuch.
„Theoretische Informatik – ganz praktisch“
(Hier: *Ab Seite 143*)



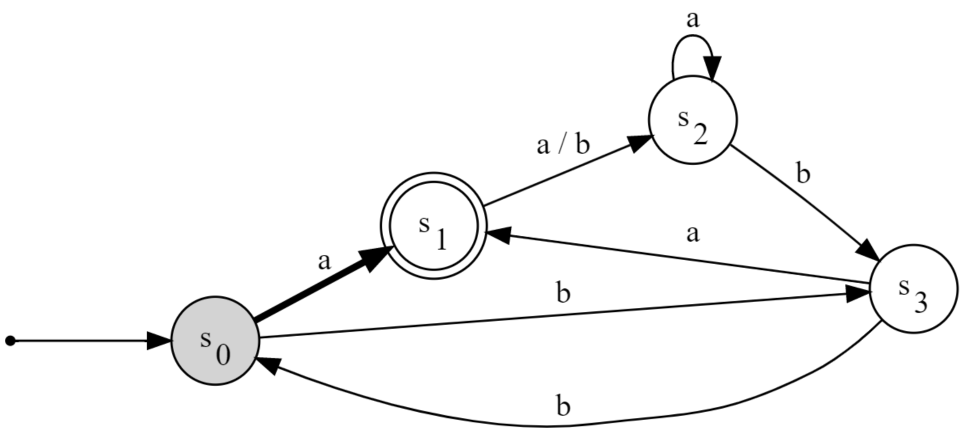
Ab S. 143

Eine Grammatik **generiert** Wörter einer Sprache

Aufgabe 2-3: Endliche Automaten und Grammatiken

Gegeben sei ein Endlicher Automat (EA): $A_i = (E_i, S_i, \delta_i, s_{0i}, F_i)$

Geben Sie die dazugehörige Grammatik vollständig an.



Lösung:

$$G = (\{s0, s1, s2, s3\}, \{a, b\}, P, s0)$$

$$\begin{aligned}P = \{&s0 \rightarrow as1 \mid bs3,\\ &s1 \rightarrow \lambda \mid as2 \mid bs2,\\ &s2 \rightarrow as2 \mid bs3,\\ &s3 \rightarrow as1 \mid bs0\}\end{aligned}$$

Aufgabe 2-4: Grammatiken

Geben Sie Grammatiken G_1 und G_2 (vollständig) an, sodass gilt: $L(G_1) = L_1$ und $L(G_2) = L_2$ und geben Sie die Produktion des jeweiligen Testworts an.

(a) $L_1 = \{w \in \{0,1\}^* \mid \forall u, v \in \{0,1\}^* : w \neq u00v\}$

Testwort: 0110

Lösung:

$$
\begin{aligned}
G &= (N, T, P, S) \\
N &= \{S\} \\
T &= \{0,1\} \\
P &= \{S \rightarrow \lambda, \\
&\quad\; S \rightarrow 0, \\
&\quad\; S \rightarrow 1S, \\
&\quad\; S \rightarrow 01S\}
\end{aligned}
$$

Testwort: $\underline{S} \Rightarrow 01\underline{S} \Rightarrow 011\underline{S} \Rightarrow 0110$

Skript ID-2734

Aufgabe 2-4: Grammatiken

(b) $L_2 = \{w \in \{0,1\}^+ \mid |w|_1 \; mod \; 3 = 0\}$ (Für ein Alphabet $E, w \in E^*, a \in E$ bezeichne $|w|_a$ die Anzahl der a's in w.)

Testwort: 01011

Aufgabe 2-4: Grammatiken

(b) $L_2 = \{w \in \{0,1\}^+ \mid |w|_1 \; mod \; 3 = 0\}$ (Für ein Alphabet $E, w \in E^*, a \in E$ bezeichne $|w|_a$ die Anzahl der a's in w.)

Testwort: 01011

Lösung (rechtslinear):

$G = (N, T, P, S)$
$N = \{S, A, B\}$
$T = \{0,1\}$
$P = \{$
$S \rightarrow 0S|1A|0,$
$A \rightarrow 0A \mid 1B,$
$B \rightarrow 0B|1S|1\}$
$\}$

Testwort: $S \Rightarrow 0S \Rightarrow 01A \Rightarrow 010A \Rightarrow 0101B \Rightarrow 01011$

Single-Choice-Aufgabe

Alle Grammatiken, die eine Sprache L vom Typ i erzeugen, sind auch selbst vom Typ i ($i \in \{0, \dots 3\}$).

☐ WAHR

☒ FALSCH

Erklärung:
Eine Grammatik vom Typ i kann eine Sprache vom Typ j mit $j \geq i$ erzeugen, aber nicht vom Typ k mit $k < i$. Bspw. kann eine Typ-2-Sprache von einer Grammatik vom Typ 1 erzeugt werden, aber nicht von einer Grammatik vom Typ 3, da bei dieser zu starke Einschränkungen der Regelbildung vorliegen.

Einführungsaufgabe: $\frac{\text{Reguläre Ausdrücke}}{\text{Endliche Automaten}}$

Auf welchen Basisoperationen basiert ein **regulärer Ausdruck** und welche Sprache wird durch folgenden regulären Ausdruck α beschrieben?

$$\alpha = (a^* + b^* \cdot a)b$$

Iteration Summe Produkt

- Basisoperationen:
- $L(\alpha) = \{w \in \{a,b\}^* \mid w = a^n b$ oder $w = b^n ab$ für $n \in \mathbb{N}_0\}$

! Klammer vor Iteration vor Produkt vor Summe !

Aufgabe 2-5: Reguläre Ausdrücke

Für ein Alphabet $E, w \in E^*, a \in E$ bezeichne $|w|_a$ die Anzahl der a's in w.

Erzeugen Sie zu den Sprachen $L_i, i \in \{1,2\}$ reguläre Ausdrücke α_i, sodass gilt:

$$L(\alpha_i) = L_i$$

(a) $L_1 = \{w \in \{0,1\}^* \mid |w|_1 \geq 1\}$

Es gibt mindestens eine Eins. Davor und danach können beliebig viele Einsen oder Nullen stehen.

Lösung:

beliebig

$E = \{0,1\}$

$RA: \alpha_1 = (0+1)^* 1 (0+1)^*$

Aufgabe 2-5: Reguläre Ausdrücke

(b) $L_2 = \{w \in \{0,1\}^* \mid |w|_1 \leq 3\}$

Lösung:

Es muss keine Eins kommen, aber es kann eine kommen. Die Nullen davor und danach sind beliebig

$$E = \{0,1\}$$

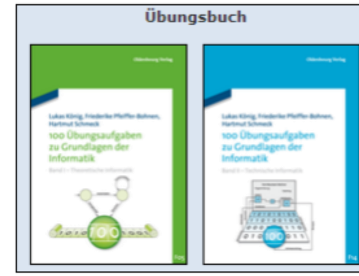
$$RA: \alpha_2 = 0^*\underbrace{(0^* + 10^*)}_{\text{1te Eins}}\underbrace{(0^* + 10^*)}_{\text{2te Eins}}\underbrace{(0^* + 10^*)}_{\text{3te Eins}}$$

(Die Anzahl der Einsen ist maximal drei, die Anzahl der Nullen ist beliebig.)

Auch Klausur-relevant…

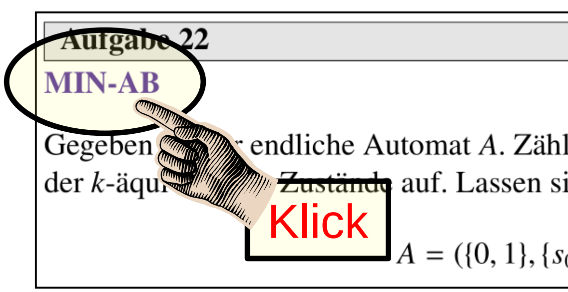
Weitere Aufgaben zu den Themen dieses Tutoriums

- Aus dem **Aufgabenpool** bzw. **Übungsbuch**:
  - Kapitel 4: Typ-3 Grammatiken: RL. Gramm. und reg. Ausdrücke
  - Kapitel 7: Pumping-Lemma für reguläre Sprachen
    (4 Aufgaben: PUM-AA, PUM-AB, PUM-AC, PUM-AF).



- Heimaufgaben zu diesem Tutorium

- Bei Fragen oder Kommentaren zu **allen Aufgaben**
  - nutzen Sie die **Diskussionsplattformen** oder
  - fragen Sie Ihren Tutor.



Fragen zu den letzten Heimaufgaben

Während die Tutoriumsaufgaben einen Einstieg in ein Thema ermöglichen, dienen die Heimaufgaben dem tieferen Verständnis und insbesondere auch der Klausurvorbereitung

?

Heimaufgabe 2-6: Reguläre Ausdrücke

Für ein Alphabet $E, w \in E^*, a \in E$ bezeichne $|w|_a$ die Anzahl der a's in w.

Erzeugen Sie zu der Sprache $L_i, i \in \{1\}$ einen regulären Ausdruck α_i, sodass gilt:

$$L(\alpha_i) = L_i$$

$L_1 = \{w \in \{0,1\}^{2n} | n \in IN_0\}$

Lösung:

$E \rightleftarrows \{0,1\}$

$RA : \alpha_1 = (\underbrace{(0+1)}_{\text{1tes Zeichen}}\underbrace{(0+1)}_{\text{2tes Zeichen}})^*$

Beliebige Wiederholung zweier Zeichen.

(Die Anzahl der Zeichen ist gerade.)

Heimaufgabe 2-7: Moore Automat

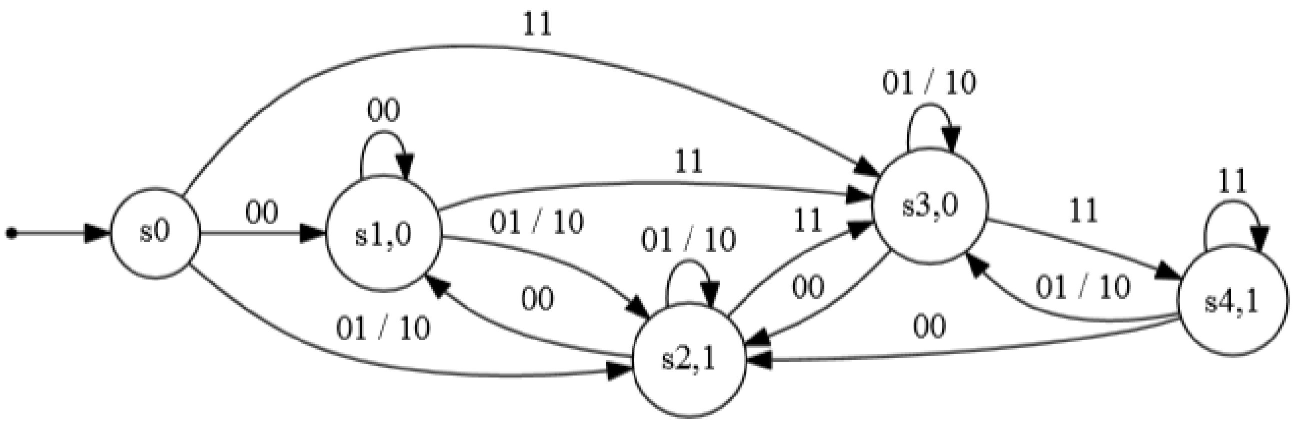
Entwerfen Sie einen Moore-Automaten M = (E,S,A,δ,γ,s0) zur bitseriellen Addition zweier gleichlanger Zahlen.

Heimaufgabe 2-7: Moore Automat

Lösung:

M = ({00,01,10,11},{s0,s1,s2,s3,s4},{0,1},δ,γ,a)

δ, γ:



Heimaufgabe 2-7: Moore Automat

Lösung:

Mit der Bedeutung:

- Im Startzustand s0 wird keine Ausgabe generiert.
- In Zustand s1 ist der Übertrag 0 und die Ausgabe 0.
- In Zustand s2 ist der Übertrag 0 und die Ausgabe 1.
- In Zustand s3 ist der Übertrag 1 und die Ausgabe 0.
- In Zustand s4 ist der Übertrag 1 und die Ausgabe 1.