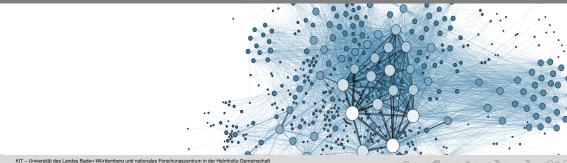




Grundbegriffe der Informatik **Tutorium 33**

Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de Lukas Bach, lukas.bach@student.kit.edu | 02.02.2017



Mealy-Automat



Maximilian Staab, maximilian.staab@fsmi.um Lukas Bach,

Mealy-Automat

Ein Mealy-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit...

Automaten

Mealy-Automat

lukas.bach@student.kit.

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Mealy-Automat



Maximilian Staab, maximilian.staab@fsmi.um Lukas Bach,

Mealy-Automat

Lukas Bach, lukas bach@student.kit., Ein Mealy-Automat ist ein Tupel $A=(Z,z_0,X,f,Y,h)$ mit...

endliche Zustandsmenge Z

Automaten

Mealy-Automat

Moore-Automat

Endlich

Akzeptoren

Reguläre Ausdrücke

Mealy-Automat



Maximilian Staab. maximilian.staab@fsmi.um

Lukas Bach,

Mealy-Automat

Ein Mealy-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit... lukas.bach@student.kit.

- endliche Zustandsmenge Z
 - Anfangszustand $z_0 \in Z$

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Mealy-Automat



Maximilian Staab, maximilian.staab@fsmi.un Lukas Bach,

Mealy-Automat

Lukas Bach, lukas.bach@student.kit., Ein Mealy-Automat ist ein Tupel $A=(Z,z_0,X,f,Y,h)$ mit...

- endliche Zustandsmenge Z
- Anfangszustand $z_0 \in Z$
- Eingabealphabet X

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Mealy-Automat



Maximilian Staab. maximilian.staab@fsmi.u

Lukas Bach.

Mealy-Automat

Ein Mealy-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit... lukas bach@student kit

- endliche Zustandsmenge Z
- Anfangszustand $z_0 \in Z$
- Eingabealphabet X
- Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Mealy-Automat



Maximilian Staab. maximilian.staab@fsmi.u

Lukas Bach.

Mealy-Automat

Ein Mealy-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit... lukas bach@student kit

- endliche Zustandsmenge Z
- Anfangszustand $z_0 \in Z$
- Eingabealphabet X
- Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$
- Ausgabealphabet Y

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Mealy-Automat



Maximilian Staab. maximilian.staab@fsmi.u Lukas Bach.

Mealy-Automat

Ein Mealy-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit... lukas bach@student kit

- endliche Zustandsmenge Z Automaten
 - Anfangszustand $z_0 \in Z$
 - Eingabealphabet X
 - Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$
 - Ausgabealphabet Y
 - Ausgabefunktion $h: Z \times X \rightarrow Y^*$

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Mealy-Automat



Maximilian Staab, maximilian.staab@fsmi.us Lukas Bach.

Mealy-Automat

Lukas Bach, lukas bach@student.kit., Ein Mealy-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit...

- Automaten
- Mealy-Automat
- Moore-Automat
- Endliche
- Akzeptoren
- Reguläre Ausdrücke
- Rechtslineare Grammatiker

- endliche Zustandsmenge Z
- Anfangszustand $z_0 \in Z$
- Eingabealphabet X
- Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$
- Ausgabealphabet Y
- Ausgabefunktion $h: Z \times X \rightarrow Y^*$

Darstellung als Graph

Mealy-Automat



Maximilian Staab, maximilian.staab@fsmi.us Lukas Bach.

Mealy-Automat

Lukas Bach, lukas.bach@student.kit., Ein Mealy-Automat ist ein Tupel $A=(Z,z_0,X,f,Y,h)$ mit...

- Automaten
- Mealy-Automat
- Moore-Automat
- Endliche
- Akzeptoren
- Reguläre Ausdrücke
- Rechtslineare Grammatiken

- endliche Zustandsmenge Z
 Anfangszustand z₀ ∈ Z
- Eingabealphabet X
- Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$
- Ausgabealphabet Y
- Ausgabefunktion $h: Z \times X \rightarrow Y^*$

Darstellung als Graph

■ Zustände → Knoten

Mealy-Automat



Maximilian Staab, maximilian.staab@fsmi.us Lukas Bach.

Mealy-Automat

Lukas Bach, lukas bach@student.kit., Ein Mealy-Automat ist ein Tupel $A=(Z,z_0,X,f,Y,h)$ mit...

- Automaten
- Mealy-Automat
- Moore-Automat
- Endlich
- Akzeptoren

Reguläre Ausdrücke

Rechtslineare

• Anfangszustand $z_0 \in Z$

■ Eingabealphabet X

■ Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$

endliche Zustandsmenge Z

Ausgabealphabet Y

• Ausgabefunktion $h: Z \times X \rightarrow Y^*$

Darstellung als Graph

- Zustände → Knoten
- Startzustand → Pfeil an diesen Knoten (ohne Anfang)

Mealy-Automat



Maximilian Staab, maximilian.staab@fsmi.um Lukas Bach.

Mealy-Automat

Lukas Bach, lukas bach@student.kit., Ein Mealy-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit...

- Automaten
- Mealy-Automat
- Moore-Automat
- Endliche
- Akzeptoren
- Reguläre Ausdrücke
- Rechtslineare Grammatiken

- endliche Zustandsmenge Z
 Anfangszustand z₀ ∈ Z
- Eingabealphabet X
- Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$
- Ausgabealphabet Y
- Ausgabefunktion $h: Z \times X \rightarrow Y^*$

Darstellung als Graph

- Zustände → Knoten
- Startzustand → Pfeil an diesen Knoten (ohne Anfang)
- Zustandsüberführungsfunktion → Kanten mit Beschriftung

Mealy-Automat



Maximilian Staab maximilian.staab@fsmi.u Lukas Bach.

Mealy-Automat

Ein Mealy-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit... lukas bach@student kit

Automaten

endliche Zustandsmenge Z

Mealy-Automat

• Anfangszustand $z_0 \in Z$

Moore-Automat

Eingabealphabet X

Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$

Akzeptoren

Ausgabealphabet Y

Reguläre

Ausgabefunktion $h: Z \times X \rightarrow Y^*$

Ausdrücke

Darstellung als Graph

Zustände → Knoten

Startzustand → Pfeil an diesen Knoten (ohne Anfang)

Zustandsüberführungsfunktion → Kanten mit Beschriftung

Ausgabefunktion → zusätzliche Kantenbeschriftung

Beispiel Mealy-Automat





maximilian.staab@fsmi.uni ' 'Lukas Bach,

lukas.bach@student.kit.ed

Automaten

Mealy-Automat

Moore-Automat

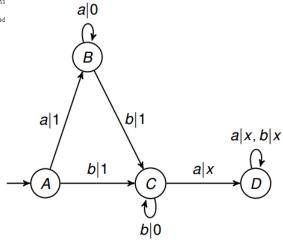
Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare



Moore-Automat



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Moore-Automat



Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach, lukas bach@student kit.

Moore-Automat

Ein Moore-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit...

- endliche Zustandsmenge Z
- Anfangszustand $z_0 \in Z$
- Eingabealphabet X
- lacktriangle Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$
- Ausgabealphabet Y

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Moore-Automat



Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach, lukas bach@student kit.

Moore-Automat

Ein Moore-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit...

- endliche Zustandsmenge Z
- Anfangszustand $z_0 \in Z$
- Eingabealphabet X
- Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$
- Ausgabealphabet Y
- → Bis hierhin alles wie bei Mealy!

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Moore-Automat



Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach, lukas bach@student kit.

Moore-Automat

Ein Moore-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit...

Automaten

endliche Zustandsmenge Z
Anfangszustand z₀ ∈ Z

Mealy-Automat

Moore-Automat

Eingabealphabet X

Endliche

Lingaboaiphabot X

Akzeptoren

■ Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$

Reguläre

Ausgabealphabet Y

Ausdrücke

→ Bis hierhin alles wie bei Mealy!

Rechtslinear

• Ausgabefunktion $h: Z \to Y^*$

Moore-Automat



Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach, lukas bach@student kit.

Moore-Automat

Ein Moore-Automat ist ein Tupel $A = (Z, z_0, X, f, Y, h)$ mit...

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare Grammatiken endliche Zustandsmenge Z

• Anfangszustand $z_0 \in Z$

Eingabealphabet X

■ Zustandsübergangsfunktion $f: Z \times X \rightarrow Z$

Ausgabealphabet Y

→ Bis hierhin alles wie bei Mealy!

• Ausgabefunktion $h: Z \to Y^*$

Bemerkung

Für jeden Mealy-Automaten kann man einen Moore-Automaten konstruieren, der genau die gleiche Aufgabe erfüllt, und umgekehrt.

Umwandlung Mealy- in Moore-Automat



Maximilian Staab, Links ein Mealy-, rechts ein Moore-Automat

maximilian.staab@fsmi.uni Lukas Bach,

lukas.bach@student.kit.ed

Automaten

Mealy-Automat

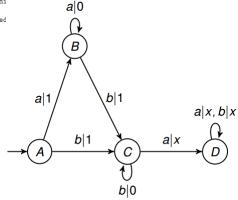
Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke



Umwandlung Mealy- in Moore-Automat



Maximilian Staab, Links ein Mealy-, rechts ein Moore-Automat

maximilian.staab@fsmi.uni Lukas Bach,

lukas.bach@student.kit.ed

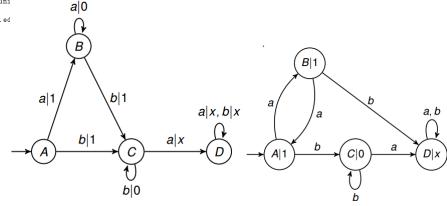
Automaten

Mealy-Automat

Moore-Automat

Endliche Akzeptoren

Reguläre Ausdrücke



Umwandlung Mealy- in Moore-Automat



Maximilian Staab, Links ein Mealy-, rechts ein Moore-Automat

maximilian.staab@fsmi.uni Lukas Bach,

lukas.bach@student.kit.ed

Automaten

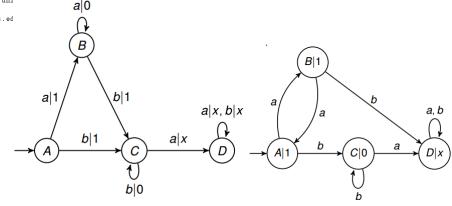
Mealy-Automat

Moore-Automat

Endliche Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken



Aufgabe

Wie sieht der Mealy-Automat als äquivalenter Moore-Automat aus, wie sieht der Moore-Automat als äquivalenter Mealy-Automat aus?

Endliche Akzeptoren



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Endliche Akzeptoren



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas bach@student kit edu

lukas.bach@student.kit.ed

Sonderfall von Moore-Automaten

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Endliche Akzeptoren



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

- Sonderfall von Moore-Automaten
- Bei einem Akzeptor will man nur wissen, ob die Eingabe akzeptiert wurde oder nicht (also reicht ein Bit als Ausgabealphabet)

Endliche Akzeptoren



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche Akzeptoren

Reguläre Ausdrücke

- Sonderfall von Moore-Automaten
- Bei einem Akzeptor will man nur wissen, ob die Eingabe akzeptiert wurde oder nicht (also reicht ein Bit als Ausgabealphabet)
- Statt der Ausgabefunktion h schreibt man einfach die Menge der akzeptierenden Zustände $F \subseteq Z$ auf

Endliche Akzeptoren



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche Akzeptoren

Reguläre Ausdrücke

- Sonderfall von Moore-Automaten
- Bei einem Akzeptor will man nur wissen, ob die Eingabe akzeptiert wurde oder nicht (also reicht ein Bit als Ausgabealphabet)
- Statt der Ausgabefunktion h schreibt man einfach die Menge der akzeptierenden Zustände $F \subseteq Z$ auf
- Zustände, die nicht akzeptieren, heißen ablehnend

Endliche Akzeptoren



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche Akzeptoren

Reguläre Ausdrücke

- Sonderfall von Moore-Automaten
- Bei einem Akzeptor will man nur wissen, ob die Eingabe akzeptiert wurde oder nicht (also reicht ein Bit als Ausgabealphabet)
- Statt der Ausgabefunktion h schreibt man einfach die Menge der akzeptierenden Zustände $F \subseteq Z$ auf
- Zustände, die nicht akzeptieren, heißen ablehnend
- Im Graphen werden akzeptierende Zustände einfach mit einem doppelten Kringel gekennzeichnet

Akzeptierte Wörter und Sprachen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Akzeptierte Wörter und Sprachen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.

Akzeptierte Wörter

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Akzeptierte Wörter und Sprachen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Akzeptierte Wörter

Automaten

Ein Wort $w \in X^*$ wird vom endlichen Akzeptor akzeptiert

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Akzeptierte Wörter und Sprachen



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Akzeptierte Wörter

Automaten

Mealy-Automat

Moore-Automat

Endliche Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Ein Wort $w \in X^*$ wird vom endlichen Akzeptor akzeptiert, wenn man ausgehend vom Anfangszustand bei Eingabe von w in einem akzeptierenden Zustand endet.

Akzeptierte Wörter und Sprachen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Akzeptierte Wörter

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken Ein Wort $w \in X^*$ wird vom endlichen Akzeptor akzeptiert, wenn man ausgehend vom Anfangszustand bei Eingabe von w in einem akzeptierenden Zustand endet.

Bemerkung

Wird ein Wort nicht akzeptiert, dann wurde es abgelehnt

Akzeptierte Wörter und Sprachen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Akzeptierte Wörter

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken Ein Wort $w \in X^*$ wird vom endlichen Akzeptor akzeptiert, wenn man ausgehend vom Anfangszustand bei Eingabe von w in einem akzeptierenden Zustand endet.

Bemerkung

Wird ein Wort nicht akzeptiert, dann wurde es abgelehnt

Akzeptierte formale Sprache

Akzeptierte Wörter und Sprachen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Akzeptierte Wörter

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Bemerkung

Wird ein Wort nicht akzeptiert, dann wurde es abgelehnt

ausgehend vom Anfangszustand bei Eingabe von w in einem

Ein Wort $w \in X^*$ wird vom endlichen Akzeptor akzeptiert, wenn man

Reguläre Ausdrücke

Rechtslineare

Akzeptierte formale Sprache

akzeptierenden Zustand endet.

Die von einem Akzeptor A akzeptierte formale Sprache L(A) ist die Menge aller von ihm akzeptierten Wörter.

Endliche Akzeptoren



Maximilian Staab,

Aufgabe zu endlichen Akzeptoren

Lukas Bach,

, Konstruiere einen endlichen Akzeptor, der die Sprache

$$L_1(A) = \{ w \in \{a, b\}^* : (N_a(w) \ge 3 \land N_b(w) \ge (2) \}$$
 erkennt.

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Endliche Akzeptoren



Maximilian Staab, maximilian.staab@fsmi.us Lukas Bach.

Aufgabe zu endlichen Akzeptoren

Lukas Bach, lukas, bach@student, kit. Konstruiere einen endlichen Akzeptor, der die Sprache

$$L_1(A) = \{ w \in \{a, b\}^* : (N_a(w) \ge 3 \land N_b(w) \ge (2) \}$$
 erkennt.

Automaten

Lösung

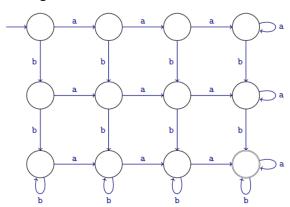
Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke



Endliche Akzeptoren



Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach.

Aufgabe zu endlichen Akzeptoren

lukas.bach@student.kit. Konstruiere einen endlichen Akzeptor, der die Sprache $L_2(A) = \{w_1 ababbw_2 | w_1, w_2 \in \{a, b\}^*\}$ erkennt.

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Endliche Akzeptoren



Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach,

Aufgabe zu endlichen Akzeptoren

Lukas.bach@student.kit. Konstruiere einen endlichen Akzeptor, der die Sprache $L_2(A) = \{w_1 \, ababbw_2 | w_1, \, w_2 \in \{a, b\}^*\}$ erkennt.

Automaten

Mealy-Automat

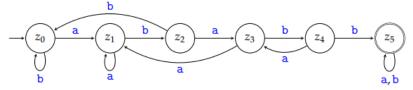
Moore-Automat

Endliche Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken

Lösung



Endliche Akzeptoren



Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach.

Aufgabe zu endlichen Akzeptoren

Lukas.bach@student.kit. Konstruiere einen endlichen Akzeptor, der die Sprache $L_2(A) = \{w_1 \, ababbw_2 | w_1, \, w_2 \in \{a, b\}^*\}$ erkennt.

Automaten

Mealy-Automat

Moore-Automat

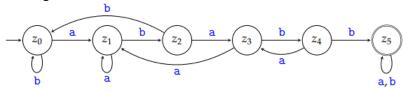
Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken

Lösung



Aufgabe

Konstuiere einen endlichen Akzeptor der die Sprache $L_3 = \{w \in \{a, b\}^* | w \notin L_2\}$ akzeptiert.

Endliche Akzeptoren



Maximilian Staab, maximilian.staab@fsmi.ur Lukas Bach,

Aufgabe zu endlichen Akzeptoren

lukas.bach@student.kit. Konstruiere einen endlichen Akzeptor, der die Sprache $L_2(A) = \{w_1 \, ababbw_2 | w_1, \, w_2 \in \{a, b\}^*\}$ erkennt.

Automaten

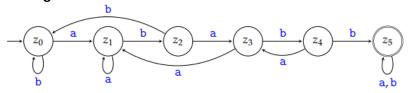
Mealy-Automat

Moore-Automat

Endliche Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken Lösung



Aufgabe

Konstuiere einen endlichen Akzeptor der die Sprache $L_3 = \{w \in \{a, b\}^* | w \notin L_2\}$ akzeptiert.

Lösung

Ablehnende Zustände wereden zu akzeptierenden und andersrum.

Endliche Akzeptoren



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken

Aufgaben zu endlichen Akzeptoren

- Gebe für den unten stehenden Automaten an, welche Sprache dieser akzeptiert.
- Gebe für die folgende Sprache über dem Alphabet $\{a,b\}$ einen endlichen Akzeptor an: $L = \{w \in \Sigma^* | N_a(w) \mod 3 > N_b(w) \mod 2\}$

$$\begin{array}{c|c} \bullet & s_0 \\ \hline \end{array} \begin{array}{c} a,b \\ \hline \end{array} \begin{array}{c} s_1 \\ \hline \end{array} \begin{array}{c} a,b \\ \hline \end{array} \begin{array}{c} s_2 \\ \hline \end{array} \begin{array}{c} b \\ \hline \end{array} \begin{array}{c} s_3 \\ \hline \end{array}$$

Lösungen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach, Lösung 1

 $L = \{w \in \Sigma^* | |w| \mod 2 = 1\}$ (Worte ungerader Länger)

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Grammatiker

Lösungen



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de,

lukas.bach@student.kit.edu Lösung 1

 $L = \{w \in \Sigma^* | |w| \text{ mod } 2 = 1\}$ (Worte ungerader Länger)

Automaten

Lösung 2

Mealy-Automat

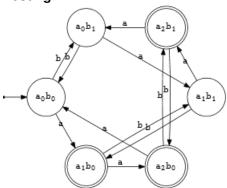
Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare



Endliche Akzeptoren



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas bach@student kit edu

Automaten

Mealy-Automat

Moore-Automat

Wann wird das leere Wort ε von einem endlichen Akzeptor akzeptiert?

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Grammatiker

Endliche Akzeptoren



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas bach@student kit edu

Automaten

Mealy-Automat

Moore-Automat

Wann wird das leere Wort ε von einem endlichen Akzeptor akzeptiert? $\varepsilon \in L(A)$ gilt genau dann, wenn der Startzustand akzeptiert wird.

Endliche

Akzeptoren

Reguläre Ausdrücke

Regulärer Ausdruck



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Regulärer Ausdruck

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Grammatiken

Regulärer Ausdruck



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Regulärer Ausdruck

Automaten Alphabet $Z = \{|, (,), *, \emptyset\}$ von "Hilfssymbolen"

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Grammatiken

Regulärer Ausdruck



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Regulärer Ausdruck

- Alphabet $Z = \{|, (,), *, \emptyset\}$ von "Hilfssymbolen"
- Alphabet A enthalten keine Zeichen aus Z

Mealy-Automat

Moore-Automat

Automaten

Endliche

Akzeptoren

Reguläre Ausdrücke

Regulärer Ausdruck



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas bach@student.kit.edu

Regulärer Ausdruck

Automaten Alphabet 7

• Alphabet $Z = \{|, (,), *, \emptyset\}$ von "Hilfssymbolen"

Alphabet A enthalten keine Zeichen aus Z

■ Ein regulärer Ausdruck (RA) über A ist eine Zeichenfolge über dem Alphabet A ∪ Z, die gewissen Vorschriften genügt.

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Regulärer Ausdruck



Maximilian Staab,

Automaten

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Regulärer Ausdruck

- Alphabet $Z = \{|, (,), *, \emptyset\}$ von "Hilfssymbolen"
 - Alphabet A enthalten keine Zeichen aus Z
 - Ein regulärer Ausdruck (RA) über A ist eine Zeichenfolge über dem Alphabet A ∪ Z, die gewissen Vorschriften genügt.
 - Vorschriften

Moore-Automat

Mealy-Automat

Akzeptoren

Reguläre Ausdrücke

Regulärer Ausdruck



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Regulärer Ausdruck

- Alphabet $Z = \{|, (,), *, \emptyset\}$ von "Hilfssymbolen"
 - Alphabet A enthalten keine Zeichen aus Z
 - Alphabet A enthalten keine Zeichen aus Z
 - Ein regulärer Ausdruck (RA) über A ist eine Zeichenfolge über dem Alphabet A ∪ Z, die gewissen Vorschriften genügt.
 - Vorschriften
 - Ø ist ein RA

Automaten

Mealy-Automat

Moore-Automat

Endlich

Akzeptoren

Reguläre Ausdrücke

Regulärer Ausdruck



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Regulärer Ausdruck

- Alphabet $Z = \{|, (,), *, \emptyset\}$ von "Hilfssymbolen"
 - Alphabet A enthalten keine Zeichen aus Z
- Ein regulärer Ausdruck (RA) über A ist eine Zeichenfolge über dem Alphabet A ∪ Z, die gewissen Vorschriften genügt.
- Vorschriften
 - Ø ist ein RA
 - Für jedes $x \in A$ ist x ein RA

Automaten

Mealy-Automat

Moore-Automat

Endlich

Akzeptoren

Reguläre Ausdrücke

Regulärer Ausdruck



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Regulärer Ausdruck

- Alphabet $Z = \{|, (,), *, \emptyset\}$ von "Hilfssymbolen"
 - Alphabet A enthalten keine Zeichen aus Z
- Ein regulärer Ausdruck (RA) über A ist eine Zeichenfolge über dem Alphabet A ∪ Z, die gewissen Vorschriften genügt.
- Vorschriften
 - Ø ist ein RA
 - Für jedes $x \in A$ ist x ein RA
 - Wenn R_1 und R_2 RA sind, dann auch $(R_1|R_2)$ und (R_1R_2)

Automaten Mealv-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Regulärer Ausdruck



Maximilian Staab,

Automaten

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,

lukas.bach@student.kit.edu

Regulärer Ausdruck

- Alphabet $Z = \{|, (,), *, \emptyset\}$ von "Hilfssymbolen"
- Alphabet A enthalten keine Zeichen aus Z
- Ein regulärer Ausdruck (RA) über A ist eine Zeichenfolge über dem Alphabet A ∪ Z, die gewissen Vorschriften genügt.
- Vorschriften
 - Ø ist ein RA
 - Für jedes $x \in A$ ist x ein RA
 - Wenn R_1 und R_2 RA sind, dann auch $(R_1|R_2)$ und (R_1R_2)
 - Wenn R ein RA ist, dann auch (R*)

Moore-Automat Endliche

Mealv-Automat

Akzeptoren

Reguläre Ausdrücke

Klammerregeln



Maximilian Staab, maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach, lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Klammerregeln



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

"Stern- vor Punktrechnung"

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Grammatiken

Klammerregeln



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

"Stern- vor Punktrechnung"

Automaten

"Punkt- vor Strichrechnung"

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Grammatiker

Klammerregeln



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

"Stern- vor Punktrechnung"

Automaten

"Punkt- vor Strichrechnung"

Mealy-Automat

 $\rightarrow R_1|R_2R_3*$ Kurzform für $(R_1|(R_2(R_3*)))$

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Klammerregeln



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

"Stern- vor Punktrechnung"

Automaten

"Punkt- vor Strichrechnung"

Mealy-Automat

 $\rightarrow R_1|R_2R_3*$ Kurzform für $(R_1|(R_2(R_3*)))$

Moore-Automat

■ Bei mehreren gleichen Operatoren ohne Klammern links geklammert

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Klammerregeln



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

"Stern- vor Punktrechnung"

Automaten

"Punkt- vor Strichrechnung"

Mealy-Automat

 $\rightarrow R_1|R_2R_3*$ Kurzform für $(R_1|(R_2(R_3*)))$

Moore-Automat

Bei mehreren gleichen Operatoren ohne Klammern links geklammert

Endliche

 $\rightarrow R_1|R_2|R_3$ Kurzform für $((R_1|R_2)|R_3)$

Akzeptoren

Reguläre Ausdrücke

Klammerregeln



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

"Stern- vor Punktrechnung"

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken "Punkt- vor Strichrechnung"

 $\rightarrow R_1|R_2R_3*$ Kurzform für $(R_1|(R_2(R_3*)))$

■ Bei mehreren gleichen Operatoren ohne Klammern links geklammert

 $\rightarrow R_1|R_2|R_3$ Kurzform für $((R_1|R_2)|R_3)$

Aufgabe

Entferne so viele Klammern wie möglich, ohne die Bedeutung des RA zu verändern.

• $(((((ab)b)*)*)|(\emptyset*))$

Klammerregeln



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

"Stern- vor Punktrechnung"

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken "Punkt- vor Strichrechnung"

 $\rightarrow R_1|R_2R_3*$ Kurzform für $(R_1|(R_2(R_3*)))$

■ Bei mehreren gleichen Operatoren ohne Klammern links geklammert

 $\rightarrow R_1|R_2|R_3$ Kurzform für $((R_1|R_2)|R_3)$

Aufgabe

Entferne so viele Klammern wie möglich, ohne die Bedeutung des RA zu verändern.

 $(((((ab)b)*)*)|(\emptyset*)) \rightarrow (abb)**|\emptyset*$

Klammerregeln



Maximilian Staab

maximilian.staab@fsmi.uni-karlsruhe.de. Lukas Bach.

lukas.bach@student.kit.edu

"Stern- vor Punktrechnung"

Automaten

Mealv-Automat

"Punkt- vor Strichrechnung"

Bei mehreren gleichen Operatoren ohne Klammern links geklammert

Moore-Automat

Endliche

Akzeptoren

Aufgabe

Reguläre Ausdrücke

Entferne so viele Klammern wie möglich, ohne die Bedeutung des RA zu verändern.

 $(((((ab)b)*)*)|(\emptyset*)) \rightarrow (abb)**|\emptyset*$

 $\rightarrow R_1|R_2R_3*$ Kurzform für $(R_1|(R_2(R_3*)))$

 $\rightarrow R_1|R_2|R_3$ Kurzform für $((R_1|R_2)|R_3)$

((a(a|b))|b)

Klammerregeln



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

"Stern- vor Punktrechnung"

Automaten

"Punkt- vor Strichrechnung"

Mealy-Automat

 $\rightarrow R_1|R_2R_3*$ Kurzform für $(R_1|(R_2(R_3*)))$

Moore-Automat

Bei mehreren gleichen Operatoren ohne Klammern links geklammert

Endliche

 $\rightarrow R_1|R_2|R_3$ Kurzform für $((R_1|R_2)|R_3)$

Akzeptoren

Aufgabe

Reguläre Ausdrücke Entferne so viele Klammern wie möglich, ohne die Bedeutung des RA zu verändern.

Rechtslinear

 $\qquad (((((ab)b)*)*)|(\emptyset*)) \rightarrow (abb)**|\emptyset*$

 $((a(a|b))|b) \rightarrow a(a|b)|b$

Alternative Definition



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

Lukas . bach@student . kit . ed Wir können die Syntax von regulären Ausdrücken auch über eine kontextfreie Grammatik definieren.

Automaten

Mealy-Automat Autgab

Vervollständigt die folgende Grammatik.

Moore-Automat

$$G = (\{R\}, \{|, (,), *, \emptyset\} \cup A, R, P)$$

Endliche Akzeptoren

$$\mathsf{mit}\; P = \{R \to \emptyset, R \to \emptyset\}$$

Reguläre Ausdrücke

Rechtslineare

Grammatiken

Alternative Definition



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de.

Lukas Bach, lukas.bach@student.kit.edWir können die Syntax von regulären Ausdrücken auch über eine kontextfreie Grammatik definieren.

Automaten

Mealv-Automat

Aufgabe

Moore-Automat

Vervollständigt die folgende Grammatik.

Endliche

$$G = (\{R\}, \{|, (,), *, \emptyset\} \cup A, R, P)$$

mit $P = \{R \rightarrow \emptyset, R \rightarrow x \text{ (mit } x \in A),$

Akzeptoren

$$R \rightarrow (R|R), R \rightarrow (RR),$$

Reguläre Ausdrücke

$$R \to (R*)$$

 $R \to \varepsilon$ }

Alternative Definition



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de,

Lukas Bach,
lukas.bach@student.kit.edWir können die Syntax von regulären Ausdrücken auch über eine kontextfreie Grammatik definieren.

Automaten

Mealy-Automat Aufgabe

Vervollständigt die folgende Grammatik.

Endliche $G = (\{R\}, \{|, (,), *, \emptyset\} \cup A, R, P)$

 $mit P = \{R \to \emptyset, R \to x \text{ (mit } x \in A),$

 $R \rightarrow (R|R), R \rightarrow (RR),$

Reguläre $R \to (R*)$

 $R \rightarrow \epsilon$ }

Rechtslineare

Akzeptoren

Wieso brauchen wir ε ?

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Notation

Automaten

Spitze Klammern (,)

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Grammatiker

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Notation

Automaten

• Spitze Klammern \langle, \rangle

Mealy-Automat

Regeln

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Notation

Automaten

• Spitze Klammern \langle,\rangle

Mealy-Automat

Regeln

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Grammatiken

Durch R beschriebene Sprache



Maximilian Staab.

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Notation

Automaten

Spitze Klammern (,)

Mealy-Automat

Regeln

Moore-Automat

Endliche

 $\langle x \rangle =$

Akzeptoren

Reguläre Ausdrücke

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach,

lukas.bach@student.kit.edu

Notation

Automaten

Spitze Klammern (,)

Mealy-Automat

Regeln

Moore-Automat

Endliche

Akzeptoren

 $\langle x \rangle = \{x\} \text{ für jedes } x \in A$

Reguläre

Ausdrücke

Rechtslineare

Grammatiken

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Notation

Automaten

• Spitze Klammern \langle , \rangle

Mealy-Automat

Regeln

Moore-Automat

Endliche

 $\langle x \rangle = \{x\} \text{ für jedes } x \in A$

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Notation

Automaten

• Spitze Klammern \langle,\rangle

Mealy-Automat

Regeln

Moore-Automat

Endliche

 $\langle x \rangle = \{x\} \text{ für jedes } x \in A$

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Notation

Automaten

• Spitze Klammern \langle, \rangle

Mealy-Automat

Regeln

Moore-Automat

Endliche

 $\langle x \rangle = \{x\} \text{ für jedes } x \in A$

Akzeptoren

Reguläre Ausdrücke

 $\langle R_1 R_1 \rangle =$

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Notation

Automaten

• Spitze Klammern \langle,\rangle

Mealy-Automat

Regeln

Moore-Automat

Endliche

 $\langle x \rangle = \{x\} \text{ für jedes } x \in A$

Akzeptoren

Reguläre Ausdrücke

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Notation

Automaten

• Spitze Klammern \langle,\rangle

Mealy-Automat

Regeln

Moore-Automat

Endliche

 $\langle x \rangle = \{x\} \text{ für jedes } x \in A$

Akzeptoren

Reguläre Ausdrücke

lacktriangledown $\langle R*
angle =$

Durch R beschriebene Sprache



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Notation

Automaten

• Spitze Klammern \langle,\rangle

Mealy-Automat

Regeln

Moore-Automat

Endliche

 $\langle x \rangle = \{x\} \text{ für jedes } x \in A$

Akzeptoren

Reguläre Ausdrücke

Charakterisierung regulärer Sprachen



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas bach@student kit edu

Automaten

Mealy-Automat

Moore-Automat

Endliche Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken

Satz

Für jede formale Sprache *L* sind äquivalent:

- 1. L kann von einem endlichen Akzeptor erkannt werden.
- 2. L kann durch einen regulären Ausdruck beschrieben werden
- 3. L kann von einer rechtslinearen Grammatik erzeugt werden.

Solche Sprachen heißten regulär.

Anwendung von regulären Ausdrücken



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.edu

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare

Zum selbst probieren: http://regexr.com/

Achtung: Reguläre Ausdrücke in praktischer Programmierung funktionieren zwar ähnlich, haben aber eine andere Syntax und können teils mehr!

Rechtslineare Grammatiken



Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas bach@student kit edu

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken

Definition

Eine rechtslineare Grammatik ist eine reguläre Grammatik G = (N, T, S, P) mit der Einschränkung, dass alle Produktionen die folgende Form haben:

- $X \to W$ mit $W \in T^*$ oder
- $x \rightarrow wY$ mit $w \in T^*$, $Y \in N$

Maximilian Staab,

maximilian.staab@fsmi.uni-karlsruhe.de, Lukas Bach.

lukas.bach@student.kit.

Aufgabe zu rechtslinearen Grammatiker

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrück

Ausdrücke

Rechtslineare Grammatiken Gebe zu $L=\{w\in\{0,1\}^*|\exists k\in\mathbb{N}_0: \textit{Num}_2(w)=2^k+1\}$ jeweils einen regulären Ausdruck R und eine rechtslineare Grammatik G an, sodass $L=\langle R\rangle=L(G)$ gilt.

Maximilian Staah

maximilian.staab@fsmi.uni-karlsruhe.de. Lukas Bach.

lukas.bach@student.kit.

Gebe zu $L = \{w \in \{0, 1\}^* | \exists k \in \mathbb{N}_0 : Num_2(w) = 2^k + 1\}$ jeweils einen

regulären Ausdruck R und eine rechtslineare Grammatik G an, sodass

Automaten

Mealy-Automat

Moore-Automat

Akzeptoren

Lösung Endliche

 $L = \langle R \rangle = L(G)$ gilt.

R = (0 * 10) | (0 * 1(0) * 1) =

Reguläre

Ausdrücke

Maximilian Staah

maximilian.staab@fsmi.uni-karlsruhe.de. Lukas Bach.

lukas.bach@student.kit.

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

Rechtslineare Grammatiken

Gebe zu $L = \{w \in \{0, 1\}^* | \exists k \in \mathbb{N}_0 : Num_2(w) = 2^k + 1\}$ jeweils einen regulären Ausdruck R und eine rechtslineare Grammatik G an, sodass $L = \langle R \rangle = L(G)$ gilt.

Lösung

R = (0*10)|(0*1(0)*1) = 0*10|0*10*1

Maximilian Staah

maximilian.staab@fsmi.uni-karlsruhe.de. Lukas Bach.

lukas.bach@student.kit.

Aufgabe zu rechtslinearen Grammatiken

Automaten

Mealv-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre Ausdrücke

Rechtslineare Grammatiken

Gebe zu $L = \{w \in \{0, 1\}^* | \exists k \in \mathbb{N}_0 : Num_2(w) = 2^k + 1\}$ jeweils einen regulären Ausdruck R und eine rechtslineare Grammatik G an, sodass $L = \langle R \rangle = L(G)$ gilt.

Lösung

$$R = (0*10)|(0*1(0)*1) = 0*10|0*10*1$$

•
$$G = (\{S, A\}, \{0, 1\}, S, \{S \rightarrow 0S | 10 | 1A, A \rightarrow 0A | 1\})$$

Maximilian Staab,

maximilian.staab@fsmi.uni Lukas Bach,

lukas.bach@student.kit.ed

Automaten

Mealy-Automat

Moore-Automat

Endliche

Akzeptoren

Reguläre

Ausdrücke

