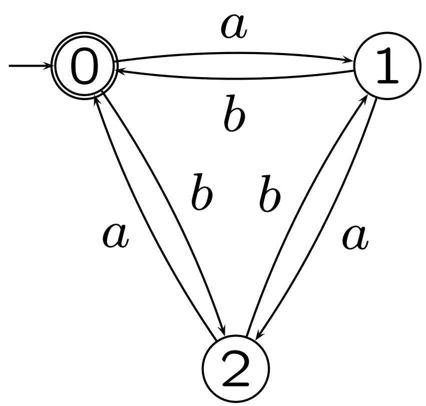
Übung "Grundbegriffe der Informatik"

Karlsruher Institut für Technologie

Matthias Schulz, Gebäude 50.34, Raum 034

email: schulz@ira.uka.de

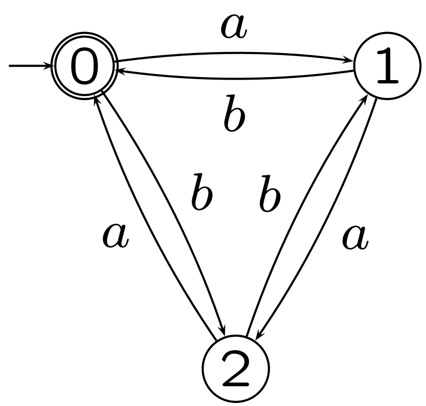
Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke



Regulärer Ausdruck für $L(A)$?

.

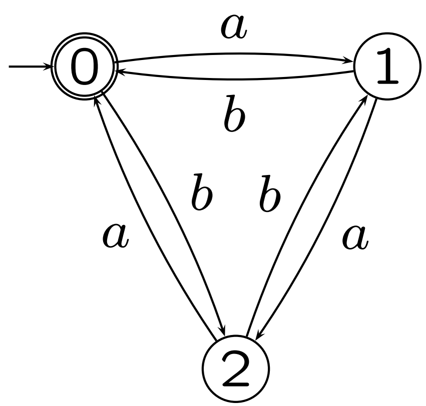
Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke



1. $F = \{z_0\} \Rightarrow R = (R')*$

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

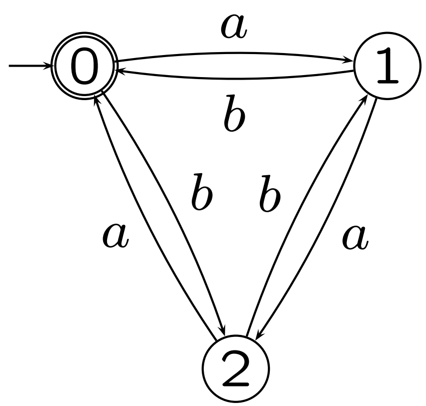


1. $F = \{z_0\} \Rightarrow R = (R')*$

R' beschreibt alle Wege von 0 nach 0, die nur über 1 und 2 gehen.

.

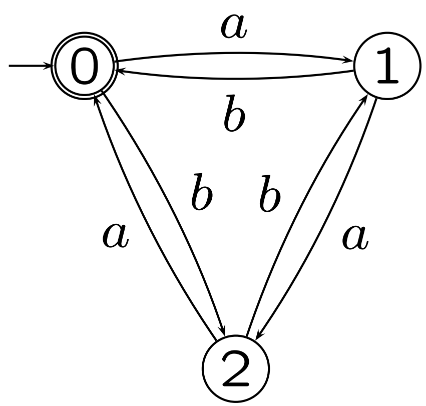
Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke



2. Erstes Zeichen a $\rightarrow$ 1. Zustand 1

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

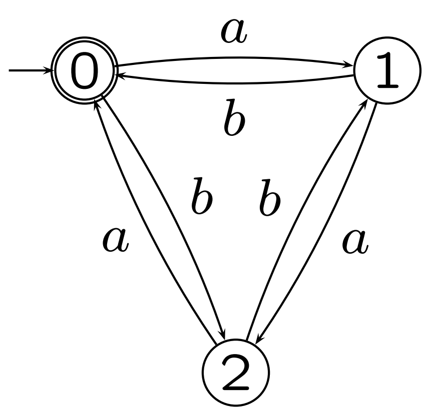


2. Erstes Zeichen $a \rightarrow$ 1. Zustand 1

Danach beliebig oft zwischen 1 und 2 hin und her $\rightarrow (ab)*$

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke



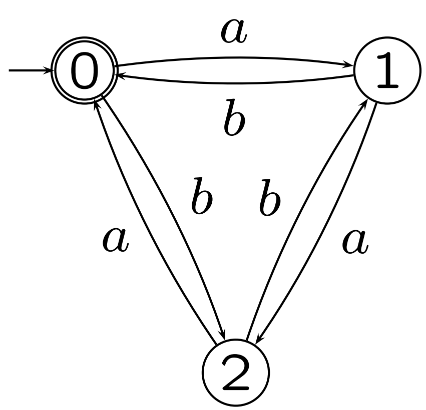
2. Erstes Zeichen $a \rightarrow$ 1. Zustand 1

Danach beliebig oft zwischen 1 und 2 hin und her $\rightarrow (ab)*$

Dann mit b oder aa zurück nach 0.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke



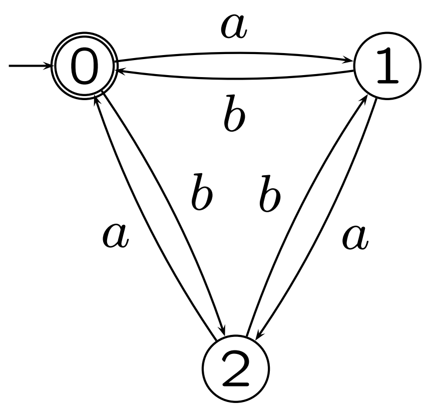
3. Erstes Zeichen $b \rightarrow$ 1. Zustand 2

Danach beliebig oft zwischen 2 und 1 hin und her $\rightarrow (ba)*$

Dann mit a oder bb zurück nach 0.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke



4. Zusammensetzen: $R = (a(ab) * (b \mid aa) \mid b(ba) * (a \mid bb))*$

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Rückwärts: $R = (a(ab) * (b \mid aa) \mid b(ba) * (a \mid bb))*$

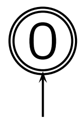
Akzeptor konstruieren.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Rückwärts: $R = (a(ab)*(b \mid aa) \mid b(ba)*(a \mid bb))*$

Akzeptor konstruieren.



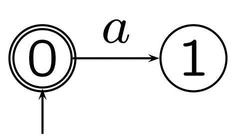
1. $R = (R')*$, also ist Anfangszustand akzeptierend.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Rückwärts: $R = (a(ab) * (b \mid aa) \mid b(ba) * (a \mid bb))*$

Akzeptor konstruieren.



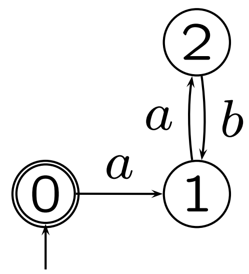
2. Mit a lande ich in anderem Zustand.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Rückwärts: $R = (\mathbf{a}(ab)*(b \mid aa) \mid b(ba)*(a \mid bb))*$

Akzeptor konstruieren.

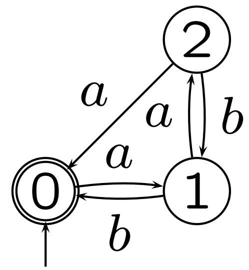


3. Mit ab komme ich in Zustand 1 zurück, also Zwischenzustand 2 einfügen.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Rückwärts: $R = (a(\mathbf{ab})*(b \mid aa) \mid b(ba)*(a \mid bb))*$

Akzeptor konstruieren.

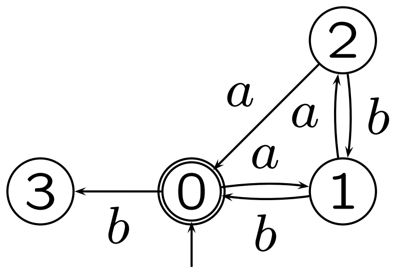


4. Nach 0 komme ich danach mit b oder aa (über gleichen Zwischenzustand).

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Rückwärts: $R = (a(ab) * (b \mid aa)|b(ba) * (a \mid bb))*$

Akzeptor konstruieren.



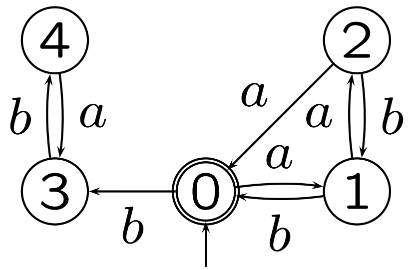
5. Mit b als erstem Zeichen komme ich in neuen Zustand.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Rückwärts: $R = (a(ab) * (b \mid aa) \mid \mathbf{b}(ba) * (a \mid bb))*$

Akzeptor konstruieren.



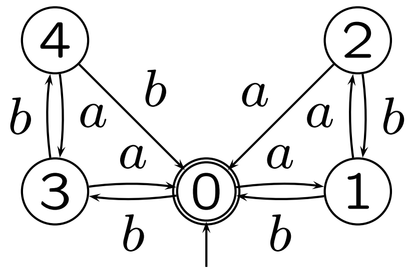
6. Mit ba komme ich nach 3 zurück über Zustand 4.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Rückwärts: $R = (a(ab) * (b \mid aa) \mid b(\mathbf{ba})*(a \mid bb))*$

Akzeptor konstruieren.



7. Mit a oder bb komme ich nach 0 zurück.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Akzeptor konstruieren: Jeder Zustand entspricht

"Menge an Stellen im Regulären Ausdruck,
an denen man bei Zusammensetzung von w sein kann."

$R = (aab \mid ab)*$

$z_0 =$ Anfang
$z_1 = f(z_0, a) =$ Erstes oder Drittes a
$z_2 = f(z_1, a) =$ Zweites a
$\ldots$

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Idee für reguläre Ausdrücke:

Zustände des Akzeptors durchnummerieren.

$\langle R_{ij}^k \rangle$ sei Menge aller Wörter w, so dass man von i bei Eingabe von w nach j kommt und dabei nur Zustände aus $\mathbb{G}_k$ durchläuft.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Idee für reguläre Ausdrücke:

Zustände des Akzeptors durchnummerieren.

$\langle R_{ij}^k \rangle$ sei Menge aller Wörter w, so dass man von i bei Eingabe von w nach j kommt und dabei nur Zustände aus $\mathbb{G}_k$ durchläuft.

R_{ij}^0 sind alle einfach.

.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Idee für reguläre Ausdrücke:

Zustände des Akzeptors durchnummerieren.

$\langle R_{ij}^k \rangle$ sei Menge aller Wörter w, so dass man von i bei Eingabe von w nach j kommt und dabei nur Zustände aus $\mathbb{G}_k$ durchläuft.

R_{ij}^{k+1}: Gehe von i nach k über Zustände aus $\mathbb{G}_k$.
Gehe beliebig oft von k nach k über Zustände aus $\mathbb{G}_k$.
Gehe von k nach j über Zustände aus $\mathbb{G}_k$.

Oder gehe direkt von i nach j über Zustände aus $\mathbb{G}_k$.

Akzeptoren $\leftrightarrow$ Reguläre Ausdrücke

Idee für reguläre Ausdrücke:

Zustände des Akzeptors von 0 bis $n-1$ durchnummerieren.

$\langle R_{ij}^{k} \rangle$ sei Menge aller Wörter w, so dass man von i bei Eingabe von w nach j kommt und dabei nur Zustände aus $\mathbb{G}_k$ durchläuft.

$$R_{ij}^{k+1} = R_{ik}^{k}(R_{kk}^{k}) * R_{kj}^{k} \mid R_{ij}^{k}$$

Sei 0 Anfangszustand und $j_0, \ldots, j_m$ akzeptierende Zustände.

Dann ist $R = R_{0j_0}^{n} \mid \ldots \mid R_{0j_m}^{n}$.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

Nach Vorlesung:

Akzeptor $\stackrel{Warshall}{\rightarrow}$ Regulärer Ausdruck $\stackrel{induktiv}{\rightarrow}$ RLG

Geht das auch einfacher?

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Idee 1: $G = (Z, X, z_0, P)$ so dass gilt:

$z_0 \Rightarrow^* wz \iff f^*(z_0, w) = z$.

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Idee 1: $G = (Z, X, z_0, P)$ so dass gilt:

$z_0 \Rightarrow^* wz \iff f^*(z_0, w) = z$.

Also: $z_0 \Rightarrow^* wz \Rightarrow wxf(z, x)$ muss Ableitung sein.

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Idee 1: $G = (Z, X, z_0, P)$ so dass gilt:

$z_0 \Rightarrow^* wz \iff f^*(z_0, w) = z$.

Also: $z_0 \Rightarrow^* wz \Rightarrow wxf(z, x)$ muss Ableitung sein,
also $\forall z \in Z \forall x \in X$: $z \to xf(z, x)$ muss Produktion sein.

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Idee 2: Ableitung $z_0 \Rightarrow^* wz$ soll mit w enden **können**, falls $z \in F$ gilt.

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Idee 2: Ableitung $z_0 \Rightarrow^* wz$ soll mit w enden **können**, falls $z \in F$ gilt.

Also $z_0 \Rightarrow^* wz \Rightarrow w$ soll möglich sein, wenn $z \in F$ gilt.

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Idee 2: Ableitung $z_0 \Rightarrow^* wz$ soll mit w enden **können**, falls $z \in F$ gilt.

Also $z_0 \Rightarrow^* wz \Rightarrow w$ soll möglich sein, wenn $z \in F$ gilt.

Also $z \to \epsilon$ soll Produktion sein, falls $z \in F$ gilt.

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Also: $G = (Z, X, z_0, P)$ mit
$P = \{z \to xf(z,x) \mid z \in Z, x \in X\} \cup \{z \to \epsilon \mid z \in F\}$

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Also: $G = (Z, X, z_0, P)$ mit
$P = \{z \to xf(z,x) \mid z \in Z, x \in X\} \cup \{z \to \epsilon \mid z \in F\}$

Dann gilt:
$w \in L(G) \iff z_0 \Rightarrow^* w \iff \exists z \in F : z_0 \Rightarrow^* wz \iff f^*(z_0, w) \in F \iff w \in L(A)$

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Also: $G = (Z, X, z_0, P)$ mit
$P = \{z \to xf(z, x) \mid z \in Z, x \in X\} \cup \{z \to \epsilon \mid z \in F\}$

Noch einfacher?

.

Akzeptoren $\leftrightarrow$ Rechtslineare Grammatiken (RLG)

$A = (Z, z_0, X, f, F)$.

Also: $G = (Z, X, z_0, P)$ mit
$P = \{z \to xf(z, x) \mid z \in Z, x \in X\} \cup \{z \to \epsilon \mid z \in F\}$

Müllzustände J führen dazu, dass aus wJ kein Wort $w' \in X^*$ abgeleitet werden kann
$\to$ Produktionen mit Müllzuständen auf der rechten Seite können gelöscht werden.

.

Strukturelle Induktion - Wörter

Induktionsanfang: Zeige: X gilt für $w = \epsilon$.

Induktionsvoraussetzung: Schreibe: X gilt für beliebiges, aber festes $w \in A^*$.

Induktionsschritt: Schreibe: Sei $x \in A$ beliebig.

Zeige: Dann gilt X auch für wx.

Strukturelle Induktion - Strukturen

Es gibt atomare Elemente und Operationen, die aus maximal k Elementen ein größeres Element "zusammensetzen".

Induktionsanfang: Zeige: X gilt für **alle** atomaren Elemente.

Induktionsvoraussetzung: Schreibe: X gilt für beliebige, aber feste Elemente $e_1, \ldots, e_k$.

Induktionsschritt: Zeige für **jede** Operation $\circ$ mit $j \leq k$ Argumenten:
Dann gilt X auch für $\circ(e_1, e_2, \ldots e_j)$.

Regulärer Ausdruck $\to$ RLG

G_1 sei RLG für R_1, G_2 sei RLG für R_2.

Konstruiere RLG H_1 für $(R_1 \mid R_2)$ (siehe Vorlesung)

Konstruiere RLG H_2 für $(R_1 R_2)$

Konstruiere RLG H_3 für $(R_1 *)$

.

Regulärer Ausdruck $\to$ RLG

$G_1 = (N_1, T_1, S_1, P_1), G_2 = (N_2, T_2, S_2, P_2)$ mit $N_1 \cap N_2 = \emptyset$.

Konstruiere RLG H_2 für $(R_1 R_2)$

Idee: Wenn Wort aus $L(G_1)$ zu Ende, hänge S_2 an.

.

Regulärer Ausdruck $\to$ RLG

$G_1 = (N_1, T_1, S_1, P_1), G_2 = (N_2, T_2, S_2, P_2)$ mit $N_1 \cap N_2 = \emptyset$.

Konstruiere RLG H_2 für $(R_1 R_2)$

Es gelte $P_1 = Q_1 \cup Q_2$ mit $\forall X \in N_1 \forall w \in T_1^*$:

$X \to w \in P_1 \iff X \to w \in Q_2$.

$H_2 = (N_1 \cup N_2, T_1 \cup T_2, S_1,$
$Q_1 \cup \{X \to wS_2 \mid X \to w \in Q_2\} \cup P_2)$.

.

Regulärer Ausdruck $\rightarrow$ RLG

$G_1 = (N_1, T_1, S_1, P_1)$.

Konstruiere RLG H_3 für (R_1*)

Idee: Wenn Wort zu Ende, hänge wieder Startsymbol an; Startsymbol kann zu ϵ werden.

.

Regulärer Ausdruck $\rightarrow$ RLG

$G_1 = (N_1, T_1, S_1, P_1)$.

Konstruiere RLG H_3 für (R_1*)

$P_1 = Q_1 \cup Q_2$ mit $\forall X \in N_1 \forall w \in T_1^*$:

$X \rightarrow w \in P_1 \iff X \rightarrow w \in Q_2$.

$H_3 = (N_1, T_1, S_1,$
$\{S_1 \rightarrow \epsilon\} \cup Q_1 \cup \{X \rightarrow wS_1 \mid X \rightarrow w \in Q_2\})$

.

Regulärer Ausdruck $\rightarrow$ RLG

Problem: $R = ((ab) * aa)$

$G = (\{S\}, \{a, b\}, S, \{S \rightarrow abS \mid aa\})$

$H_3 = (\{S\}, \{a, b\}, S, \{S \rightarrow abS \mid aaS \mid \epsilon\})$

.

Regulärer Ausdruck $\rightarrow$ RLG

Problem: $R = ((ab) * aa)$

$G = (\{S\}, \{a, b\}, S, \{S \rightarrow abS \mid aa\})$

$H_3 = (\{S\}, \{a, b\}, S, \{S \rightarrow abS \mid aaS \mid \epsilon\})$

$ab \in L(H_3), ab \notin \langle R* \rangle$

.

Regulärer Ausdruck $\rightarrow$ RLG

$G_1 = (N_1, T_1, S_1, P_1)$.

Konstruiere RLG H_3 für (R_1*)

Idee: Neues Startsymbol S', das nicht in N_1 liegt.

Wenn Wort zu Ende, hänge wieder S' an; S' kann zu ϵ oder S_1 werden.

.

Regulärer Ausdruck $\to$ RLG

$G_1 = (N_1, T_1, S_1, P_1)$.

Konstruiere RLG H_3 für (R_1*)

Es gelte $S' \notin N_1$ und $P_1 = Q_1 \cup Q_2$ mit $\forall X \in N_1 \forall w \in T_1^*$:

$X \to w \in P_1 \iff X \to w \in Q_2$.

$H_3 = (N_1 \cup \{S'\}, T_1, S',$
$\{S' \to \epsilon \mid S_1\} \cup Q_1 \cup \{X \to wS' \mid X \to w \in Q_2\})$

.