

11.1.2013

Willkommen zur elften Übung zur Vorlesung

Grundbegriffe der Informatik



Matthias Janke

email: matthias.janke@kit.edu

- ▶ Anmeldung für den Übungsschein nicht vergessen!
- ▶ Gestern waren 605 Personen angemeldet
- ▶ Anmeldung für die Klausur nicht vergessen!
- ▶ Gestern waren 444 Personen angemeldet
- ▶ Anmeldung über Studierendenportal: studium.kit.edu
- ▶ Online Klausur-Anmeldung möglich für: INFO, INWI, MATH, PHYS

Endliche Akzeptoren

Zusammenhänge: Akzeptoren, reguläre Ausdrücke, RLG

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

Der Endzustand E , Müllzustand J

- ▶ $\forall x \in X : \forall z \in E : f(z, x) \in F$
- ▶ $J \cap F = \emptyset \wedge \forall x \in X : \forall z \in J : f(z, x) \in J$

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

Der Endzustand E , Müllzustand J

- ▶ $\forall x \in X : \forall z \in E : f(z, x) \in F$
→ Zustand aus E irgendwann erreicht \Rightarrow Wort wird akzeptiert.
- ▶ $J \cap F = \emptyset \wedge \forall x \in X : \forall z \in J : f(z, x) \in J$
→ Zustand aus J irgendwann erreicht \Rightarrow Wort wird abgelehnt. (Müllzustände)

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

Der Endzustand E , Müllzustand J

- ▶ $\forall x \in X : \forall z \in E : f(z, x) \in F$
→ Zustand aus E irgendwann erreicht \Rightarrow Wort wird akzeptiert.
- ▶ $J \cap F = \emptyset \wedge \forall x \in X : \forall z \in J : f(z, x) \in J$
→ Zustand aus J irgendwann erreicht \Rightarrow Wort wird abgelehnt. (Müllzustände)

In beiden Fällen möglich: $|E| = 1$ bzw. $|J| = 1$.

$w \in X^*$, enthält Eingabe Teilwort w ?

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

$w \in X^*$, enthält Eingabe Teilwort w ?

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

- ▶ $Z = \{ \text{Präfixe von } w \}$

$w \in X^*$, enthält Eingabe Teilwort w ?

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

- ▶ $Z = \{ \text{Präfixe von } w \}$
- ▶ $z_0 = \epsilon$

$w \in X^*$, enthält Eingabe Teilwort w ?

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

- ▶ $Z = \{ \text{Präfixe von } w \}$
- ▶ $z_0 = \epsilon$
- ▶ $F = w$

$w \in X^*$, enthält Eingabe Teilwort w ?

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

► $Z = \{ \text{Präfixe von } w \}$

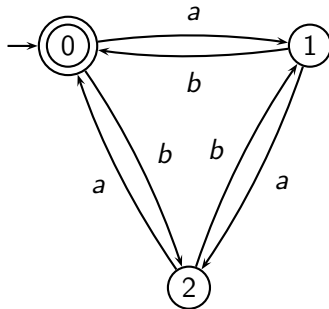
► $z_0 = \epsilon$

► $F = w$

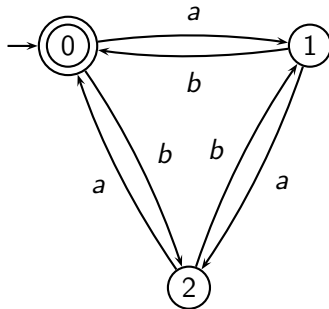
► $f(u, x) = \begin{cases} u & \text{falls } u = w \\ \text{längstes Präfix von } w, & \\ \text{das Suffix von } ux \text{ ist} & \text{sonst} \end{cases}$

Endliche Akzeptoren

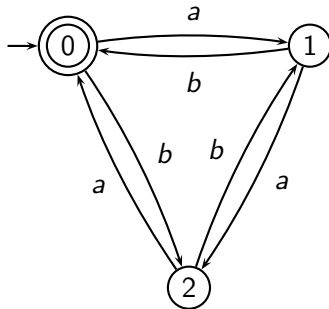
Zusammenhänge: Akzeptoren, reguläre Ausdrücke, RLG



Regulärer Ausdruck für $L(A)$?

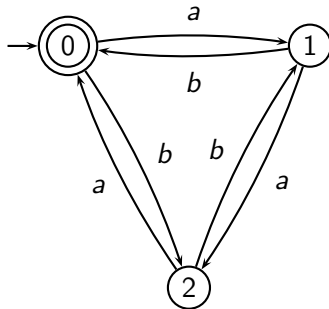


1. $F = \{z_0\} \Rightarrow R = (R')^*$

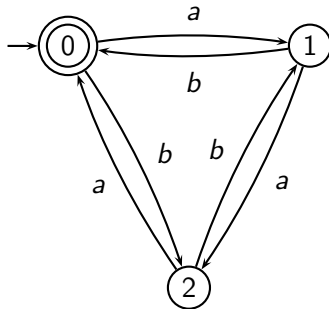


1. $F = \{z_0\} \Rightarrow R = (R')^*$

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

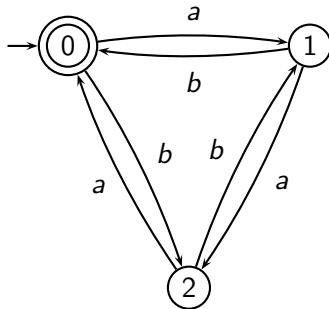


2. Erstes Zeichen $a \rightarrow$ 1. Zustand 1



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

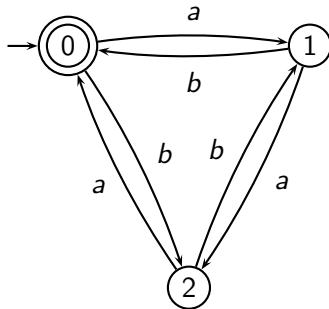
Danach beliebig oft zwischen 1 und 2 hin und her $\rightarrow (ab)^*$



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

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

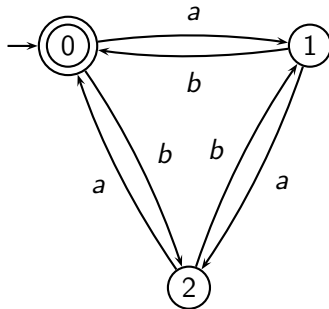
Dann mit b oder aa zurück nach 0.



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.



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

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

Akzeptor konstruieren.

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

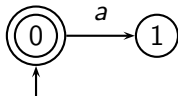
Akzeptor konstruieren.



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

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

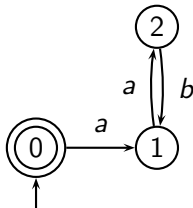
Akzeptor konstruieren.



2. Mit a lande ich in anderem Zustand.

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

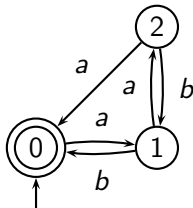
Akzeptor konstruieren.



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

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

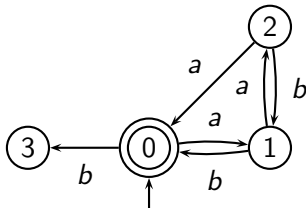
Akzeptor konstruieren.



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

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

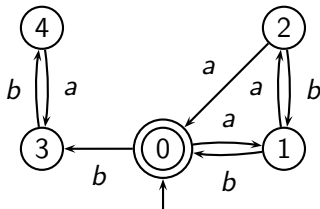
Akzeptor konstruieren.



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

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

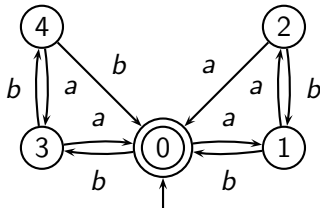
Akzeptor konstruieren.



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

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.

Akzeptor konstruieren: Jeder Zustand entspricht
"Menge an Stellen im Regulären Ausdruck,
an denen man bei Zusammensetzung von w sein kann."

$$R = a * b *$$

$$z_0 = \text{Anfang}$$

$$z_1 = f(z_0, a) = \text{Erstes } a$$

$$z_2 = f(z_1, b) = \text{Erstes } b$$

...

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.

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 .

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, \dots, j_m akzeptierende Zustände.

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

Akzeptoren \leftrightarrow Rechtslineare Grammatiken (RLG)

Nach Vorlesung:

Akzeptor $\xrightarrow{\text{Warshall}}$ Regulärer Ausdruck $\xrightarrow{\text{induktiv}}$ RLG

Geht das auch einfacher?

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

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

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

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

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.

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

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 $\forall z \in Z \forall x \in X : z \rightarrow xf(z, x)$ muss Produktion sein.

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

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

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

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.

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

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

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

Also: $G = (Z, X, z_0, P)$ mit

$$P = \{z \rightarrow xf(z, x) \mid z \in Z, x \in X\} \cup \{z \rightarrow \epsilon \mid z \in F\}$$

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

Also: $G = (Z, X, z_0, P)$ mit

$$P = \{z \rightarrow xf(z, x) \mid z \in Z, x \in X\} \cup \{z \rightarrow \epsilon \mid z \in F\}$$

Dann gilt:

$$\begin{aligned} w \in L(G) &\iff z_0 \Rightarrow^* w \iff \exists z \in F : z_0 \Rightarrow^* wz \iff \\ &f^*(z_0, w) \in F \iff w \in L(A) \end{aligned}$$

Akzeptoren \leftrightarrow Rechtslineare Grammatiken (RLG)

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

Also: $G = (Z, X, z_0, P)$ mit

$$P = \{z \rightarrow xf(z, x) \mid z \in Z, x \in X\} \cup \{z \rightarrow \epsilon \mid z \in F\}$$

Noch einfacher?

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

Also: $G = (Z, X, z_0, P)$ mit

$$P = \{z \rightarrow xf(z, x) \mid z \in Z, x \in X\} \cup \{z \rightarrow \epsilon \mid z \in F\}$$

Müllzustände J führen dazu, dass aus wJ kein Wort $w' \in X^*$ abgeleitet werden kann

→ Produktionen mit Müllzuständen auf der rechten Seite können gelöscht werden.

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^*)

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

$G_1 = (N_1, T_1, S_1, P_1)$, $G_2 = (N_2, T_2, S_2, P_2)$ 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 $\forall X \in N_1 \forall w \in T_1^*$:

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

$H_2 = (N_1 \cup N_2, T_1 \cup T_2, S_1,$

$Q_1 \cup \{X \rightarrow w S_2 \mid X \rightarrow w \in Q_2\} \cup P_2)$.

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

Konstruiere RLG H_3 für (R_1^*)

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

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

Konstruiere RLG H_3 für (R_1^*)

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

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

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

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

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

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

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

Nach Konstruktion für R^* :

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

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

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

Nach Konstruktion für R^* :

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

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

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

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.

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

Konstruiere RLG H_3 für (R_1^*)

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

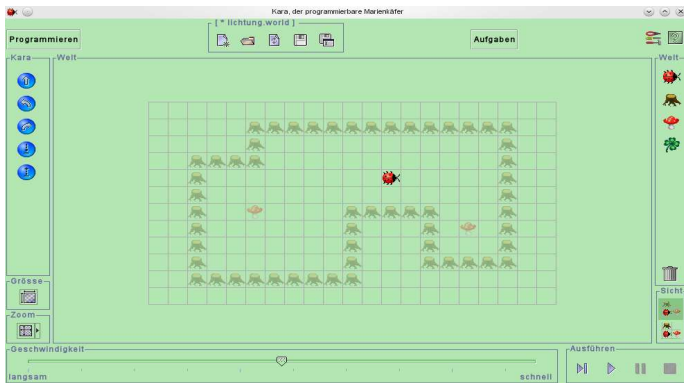
$$X \rightarrow w \in P_1 \iff X \rightarrow w \in Q_2.$$

$$H_3 = (N_1 \cup \{S'\}, T_1, S',$$

$$\{S' \rightarrow \epsilon \mid S_1\} \cup Q_1 \cup \{X \rightarrow wS' \mid X \rightarrow w \in Q_2\})$$

<http://www.swisseduc.ch/informatik/karatojava/kara/>

Mittels grafischer “Entwicklungsumgebung” kann über endliche Automaten das Verhalten eines Marienkäfers programmiert werden.



Themen für das elfte Übungsblatt:

- ▶ Akzeptoren
- ▶ reguläre Ausdrücke
- ▶ rechtslineare Grammatiken
- ▶ Beweis durch Induktion