

Grundlagen der Theoretischen Informatik

3. Endliche Automaten (V)

21.05.2015

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Bis jetzt

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational = Reguläre Ausdrücke

Pumping-Lemma: Formal

Theorem (Pumping-Lemma für L_3 -Sprachen)

Sei $L \in \mathbf{RAT}$. Dann existiert ein $n \in \mathbb{N}$, so dass:

Für alle

$$x \in L \quad \text{mit} \quad |x| \geq n$$

existiert eine Zerlegung

$$x = uvw \quad u, v, w \in \Sigma^*$$

mit

- $|v| \geq 1$
- $|v| < n$
- $uv^m w \in L$ für alle $m \in \mathbb{N}$

Pumping-Lemma: Stärkere Variante

Theorem (Pumping-Lemma für L_3 -Sprachen, stärkere Variante)

Sei $L \in \mathbf{RAT}$. Dann existiert ein $n \in \mathbb{N}$, so dass:

Für alle

$$x \in L \quad \text{mit} \quad |x| \geq n$$

existiert eine Zerlegung

$$x = uvw \quad u, v, w \in \Sigma^*$$

mit

- $|v| \geq 1$
- $|uv| < n$ (statt $|v| < n$)
- $uv^m w \in L$ für alle $m \in \mathbb{N}$

Pumping-Lemma: Anwendung der stärkeren Variante

Beispiel (Palindrome)

Die Sprache der Palindrome

$$L = \{ww^R \mid w \in \{a, b\}^*\}$$

ist nicht regulär

- Beweis gelingt **nicht** mit der schwächeren Variante des PL (die schwächere Version gilt für die Sprache)
- Beweis **gelingt** mit der stärkeren Varianten des PL

Pumping-Lemma: Anwendung der stärkeren Variante

Beispiel (Palindrome)

Die Sprache der Palindrome

$$L = \{ww^R \mid w \in \{a, b\}^*\}$$

ist nicht regulär

Annahme: L ist regulär. Dann gilt für L_1 das Pumping-Lemma. Sei n die Zahl aus dem Pumping-Lemma.

Dann muss sich das Wort $a^n bba^n \in L$ aufpumpen lassen (da $|a^n bba^n| \geq n$).

Sei $a^n bba^n = uvw$ eine passende Zerlegung laut Lemma.

Da $|uv| < n$, ist $u = a^i$, $v = a^j$, $w = a^k bba^n$, wobei $j > 0$ und $i + j + k = n$

Aber dann $uv^0w = a^{i+k} bba^n \notin L$, da $i + k < n$. Widerspruch.

Übersicht

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- **Abschlusseigenschaften und Wortprobleme**
- Rational = Reguläre Ausdrücke

Abschlusseigenschaften

Theorem. Wenn L, L' reguläre Sprachen sind, dann sind auch

- \bar{L}
- $L \cup L'$
- $L \circ L'$
- L^*
- $L \cap L'$

reguläre Sprachen.

Abschlusseigenschaften

Idee:

1) $L = L(\mathcal{A})$ mit $\mathcal{A} = (K, \Sigma, \delta, s_0, F)$

- $\mathcal{A}_{\neg} = (K, \Sigma, \delta, s_0, K \setminus F)$

2) $L = L(\mathcal{A}), L' = L(\mathcal{A}')$ mit $\mathcal{A} = (K, \Sigma, \Delta, I, F), \mathcal{A}' = (K', \Sigma', \Delta', I', F')$

- $\mathcal{A}_{\cup} = (K \cup K', \Sigma, \Delta \cup \Delta', I \cup I', F \cup F')$
- $\mathcal{A}_{\circ} = (K \cup K', \Sigma, \Delta \cup \Delta' \cup (F \times \varepsilon) \times I', I, F')$
- $\mathcal{A}_{*} = (K \cup \{\varepsilon_{neu}\}, \Sigma, \Delta \cup (F \times \varepsilon) \times I, I \cup \{\varepsilon_{neu}\}, F \cup \{\varepsilon_{neu}\})$
- $L \cap L' = \overline{(\bar{L} \cup \bar{L}')}$

Anwendung

Beispiele:

1. $L_{eq} = \{w \mid \#_a(w) = \#_b(w)\}$ nicht regulär.
2. $L_{neq} = \{w \mid \#_a(w) \neq \#_b(w)\}$ nicht regulär.
3. Die Dycksprache D_k ist nicht regulär

Anwendung

Beispiele:

1. $L_{eq} = \{w \mid \#_a(w) = \#_b(w)\}$ nicht regulär.
2. $L_{neq} = \{w \mid \#_a(w) \neq \#_b(w)\}$ nicht regulär.
3. Die Dycksprache D_k ist nicht regulär
4. $L = \{ww^R \mid w \in \{a, b\}^*\}$

Anwendung

Beispiele:

1. $L_{eq} = \{w \mid \#_a(w) = \#_b(w)\}$ nicht regulär.
2. $L_{neq} = \{w \mid \#_a(w) \neq \#_b(w)\}$ nicht regulär.
3. Die Dycksprache D_k ist nicht regulär
4. $L = \{ww^R \mid w \in \{a, b\}^*\}$

Beweis. Annahme: L regulär.

Es ist leicht zu sehen, dass $\{a^m b b a^n \mid m, n \in \mathbb{N}\}$ regulär. Dann ist auch:

$$L \cap \{a^m b b a^n \mid m, n \in \mathbb{N}\} = \{a^n b b a^n \mid n \in \mathbb{N}\} \text{ regulär}$$

Widerspruch: $\{a^n b b a^n \mid n \in \mathbb{N}\}$ ist nicht regulär (Pumping Lemma, stärkere Version).

Wortprobleme

Lemma. Sei \mathcal{A} ein endlicher Automat.

Es ist entscheidbar, ob die Sprache $L(\mathcal{A})$

1. leer ist.
2. unendlich ist.

Wortprobleme

Lemma. Sei \mathcal{A} ein endlicher Automat.

Es ist entscheidbar, ob die Sprache $L(\mathcal{A})$

1. leer ist.
2. unendlich ist.

Korollar

Sei G eine rechtslineare Grammatik.

Es ist entscheidbar, ob die Sprache $L(G)$

1. leer ist.
2. unendlich ist.

Wortprobleme

Beweis.

Zu 1.: $L(\mathcal{A})$ ist nicht leer

gdw

Es gibt einen Weg von einem initialen zu einem finalen Zustand.

Wortprobleme

Beweis.

Zu 1.: $L(\mathcal{A})$ ist nicht leer

gdw

Es gibt einen Weg von einem initialen zu einem finalen Zustand.

Zu 2.: $L(\mathcal{A})$ ist unendlich

gdw

Es gibt einen Weg von einem initialen zu einem finalen Zustand, der einen Zyklus enthält.

Beides ist leicht zu überprüfen.

Wortprobleme

Lemma.

Seien $\mathcal{A}_1, \mathcal{A}_2$ endliche Automaten.

Es ist entscheidbar, ob

(1) $L(\mathcal{A}_1) \cap L(\mathcal{A}_2) = \emptyset$

(2) $L(\mathcal{A}_1) = L(\mathcal{A}_2)$

Wortprobleme

Beweis.

Seien $\mathcal{A}_1, \mathcal{A}_2$ endliche Automaten.

(1) Man kann zu \mathcal{A}_1 und \mathcal{A}_2 einen endlichen Automaten \mathcal{A}_\cap konstruieren mit $L(\mathcal{A}_1) \cap L(\mathcal{A}_2) = L(\mathcal{A}_\cap)$.

Die Frage, ob $L(\mathcal{A}_\cap) = \emptyset$ ist entscheidbar.

(2) Man kann zu \mathcal{A}_1 und \mathcal{A}_2 einen endlichen Automaten $\mathcal{A}_=$ konstruieren mit

$$L(\mathcal{A}_=) = (L(\mathcal{A}_1) \cap \overline{L(\mathcal{A}_2)}) \cup (\overline{L(\mathcal{A}_1)} \cap L(\mathcal{A}_2))$$

Es gilt

$$L(\mathcal{A}_=) = \emptyset \quad \underline{\text{gdw}} \quad L(\mathcal{A}_1) = L(\mathcal{A}_2)$$

Dies ist entscheidbar.

Übersicht

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational = Reguläre Ausdrücke

Rational = Reguläre Ausdrücke

Hauptsatz von Kleene

Theorem (Hauptsatz von Kleene)

Die durch endliche Automaten akzeptierten Sprachen sind genau die, die man durch reguläre Ausdrücke beschreiben kann.

Reminder: Reguläre Ausdrücke

Definition (Reguläre Ausdrücke)

Menge \mathcal{Reg}_Σ der **regulären Ausdrücke** (über Σ) ist definiert durch:

1. \emptyset ist ein regulärer Ausdruck
2. Für jedes $a \in \Sigma$ ist a ein regulärer Ausdruck
3. Sind r und s reguläre Ausdrücke, so auch
 - $(r + s)$ (Vereinigung),
 - (rs) (Konkatenation),
 - (r^*) (Kleene Stern)

Klammern können weggelassen werden, dann

- $*$ hat Vorrang vor Konkatenation
- Konkatenation hat Vorrang vor $+$

Reguläre Ausdrücke

Definition (Semantik regulärer Ausdrücke)

Ein regulärer Ausdruck r stellt eine Sprache $\mathcal{I}(r)$ über Σ wie folgt dar:

$$\begin{aligned}\mathcal{I}(0) &:= \emptyset \\ \mathcal{I}(a) &:= \{a\} \quad \text{für } a \in \Sigma \\ \mathcal{I}(r + s) &:= \mathcal{I}(r) \cup \mathcal{I}(s) \\ \mathcal{I}(rs) &:= \mathcal{I}(r)\mathcal{I}(s) \\ \mathcal{I}(r^*) &:= \mathcal{I}(r)^*\end{aligned}$$

Wir benutzen auch das Makro ...

$$1 := 0^*$$

Es gilt: $\mathcal{I}(1) = \{\varepsilon\}$

Hauptsatz von Kleene

Theorem (Hauptsatz von Kleene)

Die durch endliche Automaten akzeptierten Sprachen sind genau die, die man durch reguläre Ausdrücke beschreiben kann.

Beweis.

„ \Rightarrow “ (schwierigere Richtung)

Gegeben ein DEA \mathcal{A} .

Zustände von \mathcal{A} seien q_1, \dots, q_n .

O.B.D.A. sei q_1 der initiale Zustand von \mathcal{A}

Induktion über die **Kompliziertheit** des Akzeptierens

Hauptsatz von Kleene

Beweis.

„ \Rightarrow “ (schwierigere Richtung)

Gegeben ein DEA \mathcal{A} .

Zustände von \mathcal{A} seien q_1, \dots, q_n .

O.B.D.A. sei q_1 der initiale Zustand von \mathcal{A}

Induktion über die **Kompliziertheit** des Akzeptierens

Dafür sei genauer:

$$R_{i,j}^k := \{w \in \Sigma^* : \delta^*(q_i, w) = q_j \text{ und für alle Präfixe } u \text{ von } w \\ \text{mit } \varepsilon \neq u \neq w \text{ gilt } \delta^*(q_i, u) \in \{q_1, q_2, \dots, q_k\}\}$$

Hauptsatz von Kleene: Beweis

Beweis (Forts.)

Offensichtlich ist

$$L(\mathcal{A}) = \bigcup_{q_f \in F} R_{1,f}^n$$

Es genügt zu zeigen:

Alle Mengen $R_{1,f}^n$ sind durch reguläre Ausdrücke beschreibbar.

Dazu: Durch Induktion über k (an der Tafel):

Für alle $1 \leq i, j \leq n$: $R_{1,j}^k$ ist durch einen regulären Ausdruck beschreibbar.

Hauptsatz von Kleene: Beweis

Beweis (Forts.) Dazu: Durch Induktion über k (an der Tafel):

Für alle $1 \leq i, j \leq n$: $R_{1,j}^k$ ist durch einen regulären Ausdruck beschreibbar.

Induktionsbasis $k = 0$

$$R_{ij}^0 = \begin{cases} \{a \in \Sigma \mid \delta(q_i, a) = q_j\} & \text{falls } i \neq j \\ \{a \in \Sigma \mid \delta(q_i, a) = q_j\} \cup \{\varepsilon\} & \text{falls } i = j \end{cases}$$

Induktionsvoraussetzung $R_{ij}^k \in \mathfrak{Reg}_\Sigma$

Induktionsschritt Zu zeigen: $R_{ij}^{k+1} \in \mathfrak{Reg}_\Sigma$

$$R_{ij}^{k+1} = R_{ij}^k \cup R_{i,k+1}^k (R_{k+1,k+1}^k)^* R_{k+1,j}^k$$

Hauptsatz von Kleene: Beweis

Beweis (Forts.)

„ \Leftarrow “ (einfachere Richtung)

Durch **Induktion** über den **Aufbau** regulärer Ausdrücke:

Zu jedem regulären Ausdruck gibt es einen äquivalenten ε -NDEA

(an der Tafel)

Übersicht

- Determinierte endliche Automaten (DEAs)
- Indeterminierte endliche Automaten (NDEAs)
- Automaten mit epsilon-Kanten
- Endliche Automaten akzeptieren genau die Typ-3-Sprachen
- Pumping Lemma
- Abschlusseigenschaften und Wortprobleme
- Rational = Reguläre Ausdrücke

Bis jetzt

1. Motivation
2. Terminologie
3. Endliche Automaten und reguläre Sprachen
4. Kellerautomaten und kontextfreie Sprachen
5. Turingmaschinen und rekursiv aufzählbare Sprachen
6. Berechenbarkeit, (Un-)Entscheidbarkeit
7. Komplexitätsklassen P und NP

1. Teilklausur

1. Motivation
2. Terminologie
3. Endliche Automaten und reguläre Sprachen
4. Kellerautomaten und kontextfreie Sprachen
5. Turingmaschinen und rekursiv aufzählbare Sprachen
6. Berechenbarkeit, (Un-)Entscheidbarkeit
7. Komplexitätsklassen P und NP