Grundlagen der Theoretischen Informatik

3. Endliche Automaten (IV)

6.05.2021

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Letzte Vorlesung

- 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

Heute

- 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 und Wortprobleme

Abschlusseigenschaften

Lemma. Seien zwei reguläre Sprachen L, L' gegeben.

Dann kann man folgende endlichen Automaten konstruieren:

- $\mathcal{A}_\neg$ akzeptiert $\overline{L} = \Sigma^* \setminus L$
- $\mathcal{A}_\cup$ akzeptiert $L \cup L'$
- $\mathcal{A}_\circ$ akzeptiert $L \circ L'$
- $\mathcal{A}_*$ akzeptiert L^*
- $\mathcal{A}_\cap$ akzeptiert $L \cap L'$

Abschlusseigenschaften

Idee:

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

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

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

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

Abschlusseigenschaften

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

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

reguläre Sprachen.

Beweis.

Gemäß Lemma existieren Automaten, die diese Sprachen akzeptieren.

Also sind sie regulär.

Anwendung

Beispiele:

1. $L_{eq} = \{w \mid \#_a(w) = \#_b(w)\}$ nicht regulär.

Anwendung

Beispiele:

1. $L_{eq} = \{w \mid \#_a(w) = \#_b(w)\}$ nicht regulär.

Beweis. Annahme: L_{eq} regulär

Da $\{a^m cb^n \mid m, n \geq 0\}$ regulär, wäre dann auch

$L_{eq} \cap \{a^m cb^n \mid m, n \geq 0\}$ regulär.

Aber $L_{eq} \cap \{a^m cb^n \mid m, n \geq 0\} = \{a^i cb^i \mid i \geq 0\}$ nicht regulär.

Anwendung

Beispiele:

2. $L_{neq} = \{w \mid \#_a(w) \neq \#_b(w)\}$ nicht regulär.

Anwendung

Beispiele:

2. $L_{neq} = \{w \mid \#_a(w) \neq \#_b(w)\}$ nicht regulär.

Beweis. $L_{eq} = \overline{L}_{neq}$.

Falls L_{neq} regulär wäre, dann wäre auch L_{eq} regulär. Widerspruch.

Anwendung

Beispiele:

3. Die Dycksprache D_k ist nicht regulär

Anwendung

Beispiele:

3. Die Dycksprache D_k ist nicht regulär

Definition

Gegeben:

– $k \in \mathbb{N}$

– $\Sigma_k := \{x_1, \overline{x_1}, x_2 \ldots, x_k, \overline{x_k}\}$ ein Alphabet mit $2k$ Symbolen

Die Dycksprache D_k ist die kleinste Menge, die folgende Bedingungen erfüllt:

1. $\epsilon \in D_k$,
2. Falls $w \in D_k$, so auch $x_i w \overline{x_i}$.
3. Falls u, $v \in D_k$, so auch uv.

Interpretiert man die x_i als öffnende und die $\overline{x_i}$ als zugehörige schließende Klammern, so kann man die Dycksprache als die **Menge aller korrekten Klammerausdrücke** sehen.

Anwendung

Beispiele:

3. Die Dycksprache D_k ist nicht regulär

Beweis:

$D_k \cap \{x_i\}^*\{\overline{x_i}\}^* = \{x_i^n \overline{x_i}^n \mid n \in \mathbb{N}\}$.

- $\{x_i\}^*\{\overline{x_i}\}^*$ ist regulär
- $\{x_i^n \overline{x_i}^n \mid n \in \mathbb{N}\}$ ist nicht regulär (Pumping Lemma)

Anwendung

Beispiele:

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

Beweis. Annahme: L regulär.

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

Dann ist auch $\quad L \cap \{a^m bba^n \mid m, n \in \mathbb{N}\} = \{a^n bba^n \mid n \in \mathbb{N}\} \quad$ regulär

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

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

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

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.

Beweis.

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

gdw

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

Leicht zu überprüfen

Wortprobleme

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

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

1. leer ist.
2. unendlich ist.

Beweis.

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

gdw

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

Leicht zu überprüfen.

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

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

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

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.

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

Beweis.

(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 $\mathfrak{Reg}_\Sigma$ der **regulären Ausdrücke** (über Σ) ist definiert durch:

1. 0 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 $\mathfrak{I}(r)$ über Σ wie folgt dar:

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

Wir benutzen auch das Makro ...

$$1 := 0^*$$

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

Hauptsatz von Kleene

Theorem (Hauptsatz von Kleene)
Die durch endliche Automaten akzeptierten Sprachen sind genau die, die man durch reguläre Ausdrücke beschreiben kann.

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 $q_1, \ldots, q_n$.

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 $q_1, \ldots, q_n$.

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

Induktion über die **Kompliziertheit** des Akzeptierens

Dafür sei genauer:

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

Hauptsatz von Kleene: Beweis

Beweis (Forts.)

Offensichtlich ist

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

Es genügt zu zeigen:

Alle Mengen $R^n_{1,f}$ sind durch reguläre Ausdrücke beschreibbar.

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

Für alle $1 \leq i, j \leq n$: $R^k_{1,j}$ ist durch einen regulären Ausdruck beschreibb.

Hauptsatz von Kleene: Beweis

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

Für alle $1 \leq i, j \leq n$: $\quad R^k_{1,j}$ ist durch einen regulären Ausdruck beschreibb.

Induktionsbasis k = 0

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

$\{a \in \Sigma \mid \delta(q_i, a) = q_j\}$ endlich

$\{a \in \Sigma \mid \delta(q_i, a) = q_j\} = \{a_1, \ldots, a_p\}$

Fall 1: $i \neq j$

$R^0_{ij} = \mathfrak{I}(a_1 + \cdots + a_p)$

Fall 1: $i = j$

$R^0_{ij} = \mathfrak{I}(a_1 + \cdots + a_p + 1)$

Hauptsatz von Kleene: Beweis

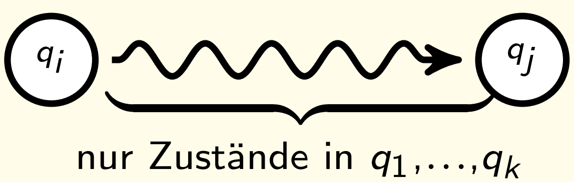
Beweis (Forts.) Dazu: Durch Induktion über k (an der Tafel):

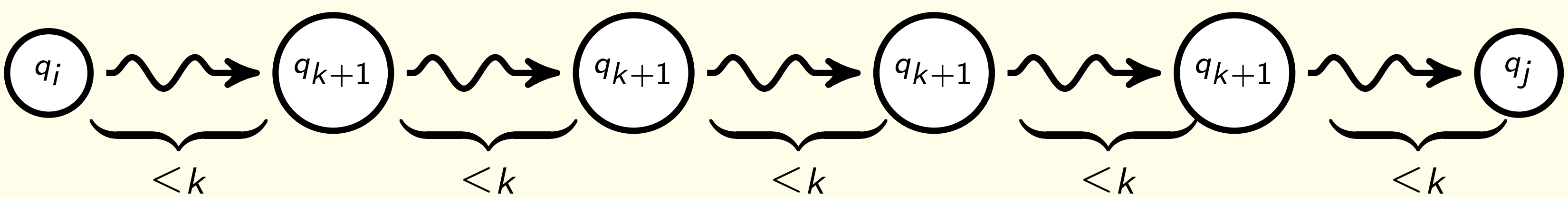
Für alle $1 \leq i, j \leq n$: $\quad R^k_{1,j}$ ist durch einen regulären Ausdruck beschreibb.

Induktionsvoraussetzung $R^k_{ij} \in \mathfrak{Reg}_\Sigma$

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

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





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)

Datei: Reg.-Ausdruck-to-NDEA-Beweis.pdf auf der Webseite der Vorlesung.

Beispiele

Berechnung eines regulären Ausdrucks für $L(\mathcal{A})$

`https://userpages.uni-koblenz.de/~sofronie/gti-ss-2015/slides/beispiel1.pdf`

Berechnung eines Automaten mit $L(\mathcal{A}) = \mathcal{I}(r)$

`https://userpages.uni-koblenz.de/~sofronie/gti-ss-2015/slides/beispiel2.pdf`

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