Tutorium
Automaten und Formale Sprachen
Teil I.

Christopher Blöcker, B. Sc.
inf9900@fh-wedel.de

SS 2012

JFLAP

Programm zur Simulation von Automaten etc., u.a.

- Endliche Automaten (+ Reguläre Ausdrücke)
- Kellerautomaten
- Turingmaschinen
- Grammatiken
- ...

Download

Google $\rightarrow$ "JFLAP download".

Start

- Linux : java -jar /pfad/JFLAP.jar &
- Windows : durch Anklicken

Frage

Warum interessieren wir uns für Formale Sprachen?

Antwort

Die Theorie der Formalen Sprachen bildet die Grundlage für Programmiersprachen.

Wenn die syntaktische Beschreibung einer Programmiersprache gegeben ist, so können wir anhand dieser Information entscheiden, ob ein gegebenes Programm gültig ist und zu dieser Sprache „gehört“.

Automaten können für ebendieses Problem verwendet werden. Sie entscheiden, ob eine gegebene Eingabe zu einer bestimmten Sprache gehört.

Außerdem bietet die Sprachentheorie die Möglichkeit, bestimmte Eigenschaften von Sprachen zu beweisen oder zu widerlegen.

Frage

Was ist eine Formale Sprache?

Definition: Formale Sprache

Eine Formale Sprache ist eine Menge von Wörtern über einem Alphabet Σ.

Anmerkung

Wir werden Sprachen allgemein mit L bezeichnen.
Für jede Sprache gilt

$$L \subseteq \Sigma^*,$$

wobei Σ^* die Menge aller Wörter bezeichnet, die aus den gegebenen Buchstaben gebildet werden können.

Definition: Alphabet

Ein Alphabet Σ ist eine Menge von Zeichen.
Die Zeichen des Alphabets werden auch als Buchstaben bezeichnet.

Definition: Wort

Ein Wort über einem Alphabet Σ ist eine endliche Folge von Zeichen aus Σ.

Anmerkung

Das Wort, das aus 0 Zeichen besteht, wird als „Leeres Wort“ ϵ bezeichnet.

Fragen

Mächtigkeit von Σ^*?
Beweis?

Beispiele

Sei $\Sigma = \{a, b, c\}$.

- Σ^1 bezeichnet die Menge aller Wörter der Länge 1
  $\Sigma^1 = \{a, b, c\}$
- Σ^2 bezeichnet die Menge aller Wörter der Länge 2
  $\Sigma^2 = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$
- Σ^k bezeichnet die Menge aller Wörter der Länge $k, k \in \mathbb{N}$
- Σ^0 bezeichnet die Menge aller Wörter der Länge 0
  $\Sigma^0 = \{\epsilon\}$
- Σ^+ bezeichnet die Menge aller nicht-leeren Wörter mit

$$\Sigma^+ = \bigcup_{k \in \mathbb{N}^+} \Sigma^k$$

- Σ^* bezeichnet die Menge aller Wörter mit

$$\Sigma^* = \Sigma^0 \cup \Sigma^+$$

Konkatenation

Zweistellige Operation in Σ^*.

$$\circ : \Sigma^* \times \Sigma^* \to \Sigma^*$$

$(\Sigma^*, \circ, \epsilon)$ ist ein Monoid.

Beispiel

Sei

$$x = \text{abc}, y = \text{def}.$$

Dann gilt:

$$x \circ y = \text{abcdef},$$

$$y \circ x = \text{defabc}.$$

Definition

Ein Monoid ist ein Tripel $(\mathcal{M}, *, e)$ mit

1. Die Verknüpfung $*$ ist assoziativ

$$a * (b * c) = (a * b) * c = a * b * c$$

mit $a, b, c \in \mathcal{M}$

2. e ist neutrales Element bezüglich $*$

$$e * a = a = a * e$$

mit $a \in \mathcal{M}$

Ein Monoid ist eine Halbgruppe mit ausgezeichnetem neutralen Element.

Frage

Wie lassen sich Sprachen beschreiben?

Antwort

- Explizite Aufzählung

$$\mathcal{L} = \{a, ab, abc\}$$

- Mengendefinition

$$\mathcal{L} = \{w \in \Sigma^* \mid w \text{ endet mit a}\}$$

- Angabe eines akzeptierenden Automaten
- Angabe eines Regulären Ausdrucks

Definition

Ein deterministischer Endlicher Automat ist ein 5-Tupel.

$$\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$$

mit

$\mathcal{Q}$	endliche, nichtleere Menge von Zuständen
Σ	endliche, nichtleere Menge, das Eingabealphabet mit $\Sigma \cap \mathcal{Q} = \emptyset$
δ	überführungsfunktion mit $\delta : \mathcal{Q} \times \Sigma \to \mathcal{Q}$
q_0	Startzustand
$\mathcal{F}$	Menge von Endzuständen mit $\mathcal{F} \subseteq \mathcal{Q}$

Zustandsmenge $\mathcal{Q}$

- endlich
  Ein Endlicher Automat darf nur aus endlich vielen Zuständen bestehen. Er hat keinen Speicher, lediglich über den aktuellen Zustand kann er sich merken, was zuletzt (bzw. bisher) geschehen ist. Wäre die Menge der Zustände nicht endlich, so könnte kein Computer den Automaten simulieren.
- nicht leer
  Sonst kann er nichts machen.

Eingabealphabet Σ

- endlich
  Wäre das Eingabealphabet nicht endlich, so wäre auch die überführungsrelation (bezogen auf den zu ihrer Darstellung benötigten Speicherplatz) unendlich groß und kein Computer könnte den Automaten simulieren.
- nicht leer
  Sonst könnte die Sprache nur ϵ enthalten und wäre uninteressant.
- $\Sigma \cap \mathcal{Q} = \emptyset$
  Ein Zustand kann nicht gleichzeitig Eingabezeichen sein.

Überführungsfunktion δ

$$\delta : \mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$$

1. Der Automat befindet sich stets in einem Zustand
2. Er liest ein Zeichen der Eingabe
3. Die überführungsfunktion δ berechnet den daraus resultierenden Nachfolgezustand

Startzustand q_o

Der Startzustand des Automaten wird mit q_0 bezeichnet.
Er kann auch einen beliebigen anderen Namen erhalten und muss gekennzeichnet werden.

Endzustandsmenge $\mathcal{F}$

$$\mathcal{F} \subseteq \mathcal{Q}$$

Die Endzustände haben folgende Bedeutung:
Wenn sich der Automat nach Verarbeiten der gesamten Eingabe (also des gesamten Eingabewortes w) in einem Zustand aus $\mathcal{F}$ befindet, dann gehört w zu der vom Automaten akzeptierten Sprache.

Akzeptierte Sprache

Die von $\mathcal{A}$ akzeptierte Sprache $\mathcal{L}_\mathcal{A}$ ist

$$\mathcal{L}_\mathcal{A} = \{w \in \Sigma^* \mid \mathcal{A} \text{ befindet sich nach Berechnung in einem Endzustand.}\}$$

Frage

Wann ist eine Menge endlich?

$$|\mathcal{M}| \overset{?}{<} \infty$$

Antwort

Eine Menge ist genau dann endlich, wenn sie keine zu sich selbst gleichmächtige, echte Teilmenge enthält.

$$|\mathcal{M}| < \infty \Leftrightarrow \bigwedge_{\mathcal{N} \subsetneq \mathcal{M}} |\mathcal{N}| < |\mathcal{M}|$$

Erweiterte Überführungsfunktion

$$\delta^* : \mathcal{Q} \times \Sigma^* \to \mathcal{Q}$$

mit

$$\delta^*(q_i, w) \leftarrow \begin{cases} q_i & \text{für } w = \epsilon \\ \delta(\delta^*(q_i, u), a) & \text{für } w = u \cdot a \end{cases}$$

Akzeptierte Sprache

Sei $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$ ein Endlicher Automat und δ^* die zu $\mathcal{A}$ gehörende erweiterte Überführungsfunktion.
Die von $\mathcal{A}$ akzeptierte Sprache $\mathcal{L}_\mathcal{A}$ ist

$$\mathcal{L}_\mathcal{A} = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in \mathcal{F}\}.$$

Anmerkung

δ^* ist die rekursive Beschreibung der Arbeitsweise eines Endlichen Automaten. Die lässt sich durch Wiederholte Anwendung der Regel für δ^* verdeutlichen.

Arbeitsweise eines DEA - Pseudocode

Sei $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$ ein DEA und $w \in \Sigma^*$ ein Eingabewort. Die Abarbeitung von w durch $\mathcal{A}$ verläuft dann wie folgt:

```
q = q0
for (i = 0; i < |w|; ++i)
  q = δ(q, wi)
return q ∈ F
```

Äquivalente Beschreibung mit δ^*

δ^* ist die rekursive Beschreibung der Arbeitsweise eines Endlichen Automaten und führt zur selben Folge von Zustandsübergängen.

Sei wieder $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$ ein DEA und $w = abc$ eine Eingabe für $\mathcal{A}$:

$$\begin{aligned}
\delta^*(q_0, abc) &= \delta(\delta^*(q_0, ab), c) \\
&= \delta(\delta(\delta^*(q_0, a), b), c) \\
&= \delta(\delta(\delta(\delta^*(q_0, \epsilon), a), b), c) \\
&= \delta(\delta(\delta(q_0, a), b), c)
\end{aligned}$$

Definition

Ein nichtdeterministischer Endlicher Automat ist ein 5-Tupel.

$$\mathcal{A} = (\mathcal{Q}, \Sigma_\epsilon, \delta, q_0, \mathcal{F})$$

mit

$\mathcal{Q}$	endliche, nichtleere Menge von Zuständen
Σ_ϵ	endliche, nichtleere Menge, das Eingabealphabet mit $\Sigma_\epsilon \cap \mathcal{Q} = \emptyset$
δ	überführungsrelation mit $\delta \subseteq \mathcal{Q} \times \Sigma_\epsilon \times \mathcal{Q}$
q_0	Startzustand
$\mathcal{F}$	Menge von Endzuständen mit $\mathcal{F} \subseteq \mathcal{Q}$

Σ_ϵ

Das Eingabealphabet enthält nun auch das leere Wort

$$\Sigma_\epsilon \leftarrow \Sigma \cup \{\epsilon\}.$$

Der Automat kann einen Zustandsübergang durchführen, ohne ein Eingabezeichen zu lesen (bzw. indem er ϵ liest).

δ

Ein Zustand kann nun mehrere Folgezustände haben

$$\delta \subseteq \mathcal{Q} \times \Sigma_\epsilon \times \mathcal{Q}$$

bzw.

$$\delta : \mathcal{Q} \times \Sigma_\epsilon \to \mathcal{P}(\mathcal{Q}).$$

Regulärer Ausdruck

Ein Regulärer Ausdruck beschreibt eindeutig eine Reguläre Menge. Jede Reguläre Menge kann als Reguläre Sprache aufgefasst werden.

Reguläre Ausdrücke über einem Alphabet Σ sind

- ϵ
- $\emptyset$
- $\mathbf{a} \Leftrightarrow a \in \Sigma$
- $\mathcal{R} \leftarrow \mathcal{R}_1 \circ \mathcal{R}_2$
- $\mathcal{R} \leftarrow \mathcal{R}_1 + \mathcal{R}_2$
- $\mathcal{R} \leftarrow \mathcal{R}^+$
- $\mathcal{R} \leftarrow \mathcal{R}^* = \mathcal{R}^+ + \epsilon$

Beipsiele für Reguläre Ausdrücke mit $\Sigma = \{0, 1\}$

Regulärer Ausdruck	Erzeugte Sprache
$\mathcal{R}_0 = \mathbf{1}$	$\mathcal{L}(\mathcal{R}_0) = \{1\}$
$\mathcal{R}_1 = \mathbf{0} + \mathbf{1}$	$\mathcal{L}(\mathcal{R}_1) = \{0, 1\}$
$\mathcal{R}_2 = \mathbf{0} \cdot \mathbf{1} (= \mathbf{01})$	$\mathcal{L}(\mathcal{R}_2) = \{0, 1\}$
$\mathcal{R}_3 = \mathbf{1} \cdot (\mathbf{0} + \mathbf{1})$	$\mathcal{L}(\mathcal{R}_3) = \{10, 11\}$
$\mathcal{R}_4 = \mathbf{0}^+ \cdot \mathbf{1}$	$\mathcal{L}(\mathcal{R}_4) = \{01, 001, 0001, 00001, \ldots\}$
$\mathcal{R}_5 = \mathbf{1}^*$	$\mathcal{L}(\mathcal{R}_5) = \{\epsilon, 1, 11, 111, 1111, \ldots\}$
$\mathcal{R}_6 = \mathcal{R}_1 \cdot \mathcal{R}_3$	$\mathcal{L}(\mathcal{R}_6) = \{010, 011, 110, 111\}$
$\mathcal{R}_7 = \mathcal{R}_1 + \epsilon$	$\mathcal{L}(\mathcal{R}_7) = \{\epsilon, 0, 1\}$
$\mathcal{R}_8 = \mathcal{R}_6 \cdot \emptyset$	$\mathcal{L}(\mathcal{R}_8) = \emptyset$
$\mathcal{R}_9 = \mathcal{R}_1^+ + \mathcal{R}_7$	$\mathcal{L}(\mathcal{R}_9) = \{\epsilon, 0, 1, 00, 01, 10, 11, \ldots\} = \Sigma^*$