

Grundlagen der Theoretischen Informatik

Sommersemester 2017

20.04.2017

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Bis jetzt

- Organisatorisches
- Literatur
- Motivation und Inhalt
- Kurzer Überblick: Logik
- Kurzer Überblick: Beweismethoden und mathematische Konzepte

Inhalt der Vorlesung

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

Inhalt der Vorlesung

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

Terminologie

In den folgenden Abschnitten führen wir die Begriffe

- **Sprache**
- **Grammatik**

ein.

Wir untersuchen insbesondere

1. wie man Probleme aus der Mathematik, Graphentheorie und Logik als **Probleme über Sprachen** formulieren kann.
2. wie man Klassen von Grammatiken von steigendem Schwierigkeitsgrad definiert: **Chomsky-Hierarchie**.
3. **wie viele** Grammatiken und Sprachen **es überhaupt gibt**
(so viele wie natürliche Zahlen, reelle Zahlen oder komplexe Zahlen?)

Sprache, Grammatik

Alphabete, Wörter

Definition (Alphabet)

Ein **Alphabet** ist eine Menge von Zeichen/Buchstaben

- Grundlage einer Sprache (die zur Verfügung stehenden Zeichen)
- Meist endlich

Alphabete, Wörter

Definition (Alphabet)

Ein **Alphabet** ist eine Menge von Zeichen/Buchstaben

- Grundlage einer Sprache (die zur Verfügung stehenden Zeichen)
- Meist endlich

Definition (Wort)

Ein **Wort** (über einem Alphabet Σ)

ist eine endliche Folge von Zeichen aus Σ

- $|w|$ bezeichnet Länge eines Wortes w
- ε bezeichnet das **leere Wort**
- $\#_a(w)$ ist die Anzahl der Vorkommen des Buchstabens a im Wort w .

Alphabete, Wörter

Beispiele

- $\Sigma_1 = \{0, 1\}$.

Wörter über Σ_1 : ε , 0, 1, 01, 1001, 100101

- $\Sigma_2 = \{a, b, c, d, e, \dots, x, y, z\}$.

Wörter über Σ_2 : ε , a, b, aba, baab, *informatik*

- $\Sigma_3 = \{(,), +, -, *, a\}$.

Wörter über Σ_3 : ε , a, (a + a), (a + a) * (a - a), ++a)

Alphabete, Wörter

Operationen auf Wörtern

Verknüpfung (Konkatenation):

$$w \circ w'$$

assoziativ, oft geschrieben als ww'

i -te Potenz:

$$w^0 = \varepsilon, \quad w^{i+1} = ww^i$$

Reverse:

w^R = das Wort w rückwärts

Alphabete, Wörter

Beispiele

- $\Sigma_1 = \{0, 1\}$.

Wörter über Σ_1 : ε , 0, 1, 01, 1001, 100101

$$\varepsilon \circ 01001 = 01001, \quad 01 \circ 1001 = 011001$$

$$1001^3 = (1001)^3 = 100110011001$$

$$01101^R = (01101)^R = 10110$$

- $\Sigma_2 = \{a, b, c, d, e, \dots, x, y, z\}$.

Wörter über Σ_2 : ε , a, b, aba, baab, informatik

$$ab \circ \varepsilon = ab, \quad aba \circ baab = ababaab$$

$$aba^2 = (aba)^2 = abaaba$$

$$\text{informatik}^R = \text{kitamrofni}.$$

- $\Sigma_3 = \{(\ , \), +, -, *, a\}$.

Wörter über Σ_3 : ε , a, (a + a), (a + a) * (a - a), ++a)

Sprache

Definition (Sprache)

Eine Sprache L (über einem Alphabet Σ)
ist eine Menge von Wörtern über Σ .

Alphabete, Wörter

Beispiele

- $\Sigma_1 = \{0, 1\}$.
Gerade = Menge aller Wörtern über Σ_1 , die die binäre Representation einer geraden Zahl sind.
- $\Sigma_2 = \{a, b, c, d, e, \dots, x, y, z\}$.
ENGLISH = $\{w \mid w \text{ wort über } \Sigma_2, \text{ das auf English eine Bedeutung hat}\}$
- $\Sigma_3 = \{(,), +, -, *, a\}$.
EXPR = Menge der Wörter über Σ_2 , die korrekt geklammerten Ausdrücke sind.

Operationen auf Sprachen

Konkatenation:

$$L \circ M = \{w \circ w' \mid w \in L, w' \in M\}$$

oft geschrieben als LM

i -te Potenz:

$$L^0 = \{\varepsilon\}, \quad L^{i+1} := LL^i$$

Reverse:

$$L^R = \{w^R \mid w \in L\}$$

Operationen auf Sprachen

Kleene-Hülle

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots$$

Variante:

$$L^+ = LL^* = L^1 \cup L^2 \cup \dots$$

Operationen auf Sprachen

Kleene-Hülle

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots$$

Variante:

$$L^+ = LL^* = L^1 \cup L^2 \cup \dots$$

Σ^* bezeichnet die Menge aller Wörter über Σ

Operationen auf Sprachen

Kleene-Hülle

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots$$

Variante:

$$L^+ = LL^* = L^1 \cup L^2 \cup \dots$$

Σ^* bezeichnet die Menge aller Wörter über Σ

Genau genommen besteht ein Unterschied:

ein Buchstabe \neq Wort, das nur aus dem einen Buchstaben besteht

Darum ist Σ selbst keine Sprache über Σ

(Oft wird über diesen Unterschied hinweggesehen)

Reguläre Ausdrücke

Definition (Reguläre Ausdrücke)

Menge \mathfrak{Reg}_Σ 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 $\mathfrak{J}(r)$ über Σ wie folgt dar:

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

Wir benutzen auch das Makro ...

$$1 := 0^*$$

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

Reguläre Ausdrücke

Übung

Welche Sprachen werden durch die folgenden regulären Ausdrücke dargestellt?

- aa
- $(a + b)^*$
- $aa^* + bb^*$

Reguläre Ausdrücke

Übung

Welche Sprachen werden durch die folgenden regulären Ausdrücke dargestellt?

- aa
- $(a + b)^*$
- $aa^* + bb^*$

$$\mathcal{J}(aa) = \mathcal{J}(a)\mathcal{J}(a) = \{a\} \circ \{a\} = \{aa\}$$

$$\mathcal{J}((a + b)^*) = \mathcal{J}(a + b)^* = (\mathcal{J}(a) \cup \mathcal{J}(b))^* = (\{a\} \cup \{b\})^* = \{a, b\}^*$$

$$\mathcal{J}(aa^* + bb^*) = \mathcal{J}(aa^*) \cup \mathcal{J}(bb^*) = \{a\}\{a\}^* \cup \{b\}\{b\}^* = \{a\}^+ \cup \{b\}^+$$

Reguläre Ausdrücke

Übung

Geben Sie einen regulären Ausdruck über $\Sigma = \{a, b, c\}$ an, der die Sprache darstellt, die genau die Wörter enthält, die mit b beginnen.

Reguläre Ausdrücke

Übung

Geben Sie einen regulären Ausdruck über $\Sigma = \{a, b, c\}$ an, der die Sprache darstellt, die genau die Wörter enthält, die mit b beginnen.

Antwort: $b(a + b + c)^*$

Reguläre Ausdrücke als Suchmuster für grep

Das Kommando grep (bzw. egrep)

- Sucht Wörter (Strings) in Dateien
- Benutzt reguläre Ausdrücke als Suchmuster
- Sehr schnell
- Volle Funktionalität mit egrep (UNIX/LINUX)

Reguläre Ausdrücke als Suchmuster für grep

Syntax bei grep

grep	Regulärer Ausdruck
ww'	ww'
$w w'$	$w + w'$
w^*	w^*
w^+	w^+

Reguläre Ausdrücke als Suchmuster für grep

Syntax bei grep

grep	Regulärer Ausdruck
ww'	ww'
$w w'$	$w + w'$
w^*	w^*
w^+	w^+

Syntactic Sugar

grep	Regulärer Ausdruck
[abc]	$a + b + c$
[a-d]	$a + b + c + d$
.	beliebiges Zeichen aus Σ

Grammatik

Grammatik

- Beschreibt eine Sprache
- Menge von Regeln, mit deren Hilfe man Wörter ableiten kann

Grammatik: Intuition

Beispiel: Grammatik für “Hund-Katze-Sätze”

Satz	→	Subjekt Prädikat Objekt
Subjekt	→	Artikel Attribut Substantiv
Artikel	→	ε
Artikel	→	der
Artikel	→	die
Artikel	→	das
Attribut	→	ε
Attribut	→	Adjektiv
Attribut	→	Adjektiv Attribut
Adjektiv	→	kleine
Adjektiv	→	bissige
Adjektiv	→	große
Substantiv	→	Hund
Substantiv	→	Katze
Prädikat	→	jagt
Objekt	→	Artikel Attribut Substantiv

Grammatik: Intuition

Beispiel: Grammatik für “Hund-Katze-Sätze”

- durch diese Grammatik können z.B. die folgenden Sätze gebildet (abgeleitet) werden:
 - der kleine bissige Hund jagt die große Katze
 - die kleine Katze jagt der bissige Hund
 - das große Katze jagt der kleine große bissige kleine Katze
- folgende Sätze werden nicht durch diese Grammatik gebildet
 - die Katze der Hund
 - Katze und Hund
 - der Hund jagt die Katze die jagt Hund

Grammatik: Intuition

Beispiel 2: Aussagenlogische Formel, $\Pi = \{P_1, \dots, P_n\}$

$$F \rightarrow \perp$$

$$F \rightarrow \top$$

$$F \rightarrow P_1$$

...

$$F \rightarrow P_n$$

$$F \rightarrow (\neg F)$$

$$F \rightarrow (F \wedge F)$$

$$F \rightarrow (F \vee F)$$

$$F \rightarrow (F \longrightarrow F)$$

$$F \rightarrow (F \longleftrightarrow F)$$

Grammatik

- Beschreibt eine Sprache
- Menge von Regeln, mit deren Hilfe man Wörter ableiten kann
- Die zu einer Grammatik gehörende Sprache besteht aus den
 - ableitbaren
 - terminalenWörtern

Grammatik

Definition (Grammatik)

Eine **Grammatik** G über einem Alphabet Σ ist ein Tupel

$$G = (V, T, R, S)$$

Dabei ist

- V eine endliche Menge von **Variablen**
- $T \subseteq \Sigma$ eine endliche Menge von **Terminalen** mit $V \cap T = \emptyset$
- R eine endliche Menge von **Regeln**
- $S \in V$ das **Startsymbol**

Grammatik

Definition (Regel)

Eine Regel ist ein Element

$$(P, Q) \in ((V \cup T)^* V (V \cup T)^*) \times (V \cup T)^*$$

Bezeichnung:

P : Prämisse

Q : Conclusio

Grammatik

Definition (Regel)

Eine Regel ist ein Element

$$(P, Q) \in ((V \cup T)^* V (V \cup T)^*) \times (V \cup T)^*$$

Das heißt:

- P und Q sind Wörter über $(V \cup T)$
- P muss mindestens eine Variable enthalten
- Q ist beliebig

Bezeichnung:

P : Prämisse

Q : Conclusio

Grammatik

Schreibweise für Regeln

- Schreibweise für Regel (P, Q) :

$$P \rightarrow_G Q \quad \text{bzw.} \quad P \rightarrow Q$$

- Abkürzung für mehrere Regeln mit derselben Prämisse:

$$P \rightarrow Q_1 \mid Q_2 \mid Q_3 \quad \text{für} \quad P \rightarrow Q_1, P \rightarrow Q_2, P \rightarrow Q_3$$

Grammatik

Schreibweise für Regeln

- Schreibweise für Regel (P, Q) :

$$P \rightarrow_G Q \quad \text{bzw.} \quad P \rightarrow Q$$

- Abkürzung für mehrere Regeln mit derselben Prämisse:

$$P \rightarrow Q_1 \mid Q_2 \mid Q_3 \quad \text{für} \quad P \rightarrow Q_1, P \rightarrow Q_2, P \rightarrow Q_3$$

Konvention (meistens)

- **Variablen** als **Großbuchstaben**
- **Terminale** als **Kleinbuchstaben**

Grammatik

Beispiel

$$\begin{aligned} S &\rightarrow B \\ B &\rightarrow \textit{do begin } B \textit{ end} \\ B &\rightarrow A \\ A &\rightarrow \textit{nop } A \\ A &\rightarrow \varepsilon \end{aligned}$$

Rechnung einer Grammatik

Algorithmus

Eingabe: Eine Grammatik

1. $aktuellWort := S$ (Startsymbol)
2. Wähle eine Regel $P \rightarrow Q$, so dass P in $aktuellWort$ vorkommt
3. Ersetze (ein) Vorkommen von P in $aktuellWort$ durch Q
4. Falls $aktuellWort$ noch Variablen enthält (nicht terminal), GOTO 2

Rechnung einer Grammatik

Algorithmus

Eingabe: Eine Grammatik

1. $aktuellWort := S$ (Startsymbol)
2. Wähle eine Regel $P \rightarrow Q$, so dass P in $aktuellWort$ vorkommt
3. Ersetze (ein) Vorkommen von P in $aktuellWort$ durch Q
4. Falls $aktuellWort$ noch Variablen enthält (nicht terminal), GOTO 2

Ausgabe: Das terminale Wort $aktuellWort$

Rechnung einer Grammatik

Algorithmus

Eingabe: Eine Grammatik

1. $aktuellWort := S$ (Startsymbol)
2. Wähle eine Regel $P \rightarrow Q$, so dass P in $aktuellWort$ vorkommt
3. Ersetze (ein) Vorkommen von P in $aktuellWort$ durch Q
4. Falls $aktuellWort$ noch Variablen enthält (nicht terminal), GOTO 2

Ausgabe: Das terminale Wort $aktuellWort$

Beachte: Die Berechnung

- ist nicht deterministisch (Auswahl der Regel)
- kann mehr als ein Ergebnis liefern (oder auch keines)
- kann in Endlosschleifen geraten

Rechnung einer Grammatik

Beispiel:

Welche Wörter kann man ableiten?

- $G_a = (\{S\}, \{a\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow aS$$

$$R_2 = S \rightarrow \epsilon$$

Rechnung einer Grammatik

Beispiel:

Welche Wörter kann man ableiten?

- $G_a = (\{S\}, \{a\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow aS$$

$$R_2 = S \rightarrow \epsilon$$

- $G_{ab} = (\{S\}, \{a, b\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow aSb$$

$$R_2 = S \rightarrow \epsilon$$

Rechnung einer Grammatik

Beispiel:

Welche Wörter kann man ableiten?

- $G_a = (\{S\}, \{a\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow aS$$

$$R_2 = S \rightarrow \epsilon$$

- $G_{ab} = (\{S\}, \{a, b\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow aSb$$

$$R_2 = S \rightarrow \epsilon$$

- Sei $G_{gerade} = (\{S, S_0\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow 1S \mid 2S_0 \mid 3S \mid 4S_0 \mid 5S \mid 6S_0 \mid 7S \mid 8S_0 \mid 9S$$

$$R_2 = S_0 \rightarrow S \mid \epsilon$$

Rechnung einer Grammatik

Beispiel:

Welche Wörter kann man ableiten?

- $G_a = (\{S\}, \{a\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow aS$$

$$R_2 = S \rightarrow \epsilon$$

- $G_{ab} = (\{S\}, \{a, b\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow aSb$$

$$R_2 = S \rightarrow \epsilon$$

- Sei $G_{gerade} = (\{S, S_0\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow 1S \mid 2S_0 \mid 3S \mid 4S_0 \mid 5S \mid 6S_0 \mid 7S \mid 8S_0 \mid 9S$$

$$R_2 = S_0 \rightarrow S \mid \epsilon$$

Beispiel

Grammatik $G_{ab} = (\{S\}, \{a, b\}, \{R_1, R_2\}, S)$

$$R_1 = S \rightarrow aSb$$

$$R_2 = S \rightarrow \epsilon$$

Mögliche Ableitung:

$$S \Rightarrow_{R_1} aSb \Rightarrow_{R_1} aaSbb \Rightarrow_{R_1} aaaSbbb \Rightarrow_{R_2} aaabbbb$$

Also: $a^3b^3 \in L(G_{ab})$

Rechnung einer Grammatik

Definition (Ableitung, Rechnung)

Gegeben:

- Grammatik $G = (V, T, R, S)$
- Wörter w, w' aus $(V \cup T)^*$

Rechnung einer Grammatik

Definition (Ableitung, Rechnung)

Gegeben:

- Grammatik $G = (V, T, R, S)$
- Wörter w, w' aus $(V \cup T)^*$

Es gilt

$$w \Longrightarrow_G w' \quad (\text{„}w \text{ geht über in } w'\text{“})$$

falls

$$\exists u, v \in (V \cup T)^* \exists P \rightarrow Q \in R \quad (w = uPv \text{ und } w' = uQv)$$

Rechnung einer Grammatik

Schreibweise für Ableitung

$$w \Longrightarrow_G^* w'$$

falls es Wörter $w_0, \dots, w_n \in (V \cup T)^*$ ($n \geq 0$) gibt mit

- $w = w_0$
- $w_n = w'$
- $w_i \Longrightarrow_G w_{i+1}$ für $0 \leq i < n$

Rechnung einer Grammatik

Schreibweise für Ableitung

$$w \Longrightarrow_G^* w'$$

falls es Wörter $w_0, \dots, w_n \in (V \cup T)^*$ ($n \geq 0$) gibt mit

- $w = w_0$
- $w_m = w'$
- $w_i \Longrightarrow_G w_{i+1}$ für $0 \leq i < n$

Merke: $w \Longrightarrow_G^* w$ gilt stets ($n = 0$)

Rechnung einer Grammatik

Schreibweise für Ableitung

$$w \Longrightarrow_G^* w'$$

falls es Wörter $w_0, \dots, w_n \in (V \cup T)^*$ ($n \geq 0$) gibt mit

- $w = w_0$
- $w_n = w'$
- $w_i \Longrightarrow_G w_{i+1}$ für $0 \leq i < n$

Merke: $w \Longrightarrow_G^* w$ gilt stets ($n = 0$)

Die Folge w_0, \dots, w_n heißt **Ableitung** oder **Rechnung**

- von w_0 nach w_n
- in G
- der Länge n

Beispiel

Beispiel

Wir betrachten die Grammatik $G = (\{S, B\}, \{a, b, c\}, \{R_0, R_1, R_2, R_3\}, S)$

$$R_0 = S \rightarrow aBBc$$

$$R_1 = B \rightarrow b$$

$$R_2 = B \rightarrow ba$$

$$R_3 = BB \rightarrow bBa$$

Beispiel

Beispiel

Wir betrachten die Grammatik $G = (\{S, B\}, \{a, b, c\}, \{R_0, R_1, R_2, R_3\}, S)$

$$R_0 = S \rightarrow aBBc$$

$$R_1 = B \rightarrow b$$

$$R_2 = B \rightarrow ba$$

$$R_3 = BB \rightarrow bBa$$

Drei Möglichkeiten, das Wort *abbac* zu erzeugen:

$$S \xRightarrow{R_0} aBBc \xRightarrow{R_1} abBc \xRightarrow{R_2} abbac$$

$$S \xRightarrow{R_0} aBBc \xRightarrow{R_2} aBbac \xRightarrow{R_1} abbac$$

$$S \xRightarrow{R_0} aBBc \xRightarrow{R_3} abBac \xRightarrow{R_1} abbac$$