

Vorlesung
Grundlagen der Theoretischen Informatik /
Einführung in die Theoretische Informatik I

Bernhard Beckert

Institut für Informatik



Sommersemester 2007

Diese Vorlesungsmaterialien basieren ganz wesentlich auf den Folien zu den Vorlesungen von

Katrin Erk (gehalten an der Universität Koblenz-Landau)

Jürgen Dix (gehalten an der TU Clausthal)

Ihnen beiden gilt mein herzlicher Dank.

– Bernhard Beckert, April 2007

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^+
-------	-------

Syntactic Sugar

grep	Regulärer Ausdruck
------	--------------------

$[abc]$	$a + b + c$
---------	-------------

$[a-d]$	$a + b + c + d$
---------	-----------------

$.$	beliebiges Zeichen aus Σ
-----	---------------------------------

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

Definition 6.6 (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**

Definition 6.7 (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

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

Beispiel 6.8

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

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

Beispiel 6.9 (Einfache Grammatiken)

Welche Wörter kann man ableiten?

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

$$R_1 = S \rightarrow aS$$

$$R_2 = S \rightarrow \varepsilon$$

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

$$R_1 = S \rightarrow aSb$$

$$R_2 = S \rightarrow \varepsilon$$

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

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

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

Definition 6.10 (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

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

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 ($n = 0$)

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

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

Beispiel 6.11 (Indeterminismus)

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:

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

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

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

Warum ist das ein *Feature* und kein *Bug*?

- Erlaubt einfachere Definition von Grammatiken
- Für manche Sprachen gibt es keine eindeutige Grammatiken
- Eine Grammatik beschreibt die **Struktur** der Wörter.
Ein Wort kann mehrere mögliche Strukturen haben.
- Für **natürliche Sprachen** braucht man das unbedingt:
Manche Sätze sind mehrdeutig (in ihrer Grammatik),
also müssen auch die Grammatiken mehrdeutig sein!

Beispiel 6.12 (Mehrdeutige Grammatik natürlicher Sätze)

Time flies like an arrow.
Fruit flies like a banana.

- Beide Sätze haben zwei mögliche grammatische Strukturen.
- Erst unser semantisches Verständnis wählt eine aus.

Definition 6.13 (Erzeugte Sprache)

Gegeben: Eine Grammatik G

Die von G erzeugte Sprache $L(G)$ ist die Menge aller **terminalen** Wörter, die durch G vom Startsymbol S aus erzeugt werden können:

$$L(G) := \{w \in T^* \mid S \xRightarrow{*}_G w\}$$

Definition 6.14 (Äquivalenz)

Zwei Grammatiken G_1, G_2 heißen **äquivalent** gdw

$$L(G_1) = L(G_2)$$