Vorlesung

Grundlagen der Theoretischen Informatik / Einführung in die Theoretische Informatik I

Bernhard Beckert

Institut für Informatik

UNIVERSITÄT KOBLENZ · LANDAU

Sommersemester 2007

Dank

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

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'
$w \mid w'$	$w + w'$
$w*$	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
- Die zu einer Grammatik gehörende Sprache besteht aus den
  – **ableitbaren**
  – **terminalen**
  Wörtern

Grammatik

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**

Grammatik

Definition 6.7 (Regel)

Eine Regel ist ein Element

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

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):

$$P \to_G Q \qquad \text{bzw.} \qquad P \to Q$$

- Abkürzung für mehrere Regeln mit derselben Prämisse:

$$P \to Q_1 \mid Q_2 \mid Q_3 \qquad \text{für} \qquad P \to Q_1,\ P \to Q_2,\ P \to Q_3$$

Konvention (meistens)

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

Grammatik

Beispiel 6.8

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

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

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 6.9 (Einfache Grammatiken)

Welche Wörter kann man ableiten?

- $G_a = (\{S\}, \{a\}, \{R_1, R_2\}, S)$
  $$\begin{aligned} R_1 &= S \rightarrow aS \\ R_2 &= S \rightarrow \varepsilon \end{aligned}$$
- $G_{ab} = (\{S\}, \{a, b\}, \{R_1, R_2\}, S)$
  $$\begin{aligned} R_1 &= S \rightarrow aSb \\ R_2 &= S \rightarrow \varepsilon \end{aligned}$$
- Sei $G_{gerade} = (\{S, S_0\}, \{0,1,2,3,4,5,6,7,8,9\}, \{R_1, R_2\}, S)$
  $$\begin{aligned} R_1 &= S \rightarrow 1S \,|\, 2S_0 \,|\, 3S \,|\, 4S_0 \,|\, 5S \,|\, 6S_0 \,|\, 7S \,|\, 8S_0 \,|\, 9S \\ R_2 &= S_0 \rightarrow S \,|\, \varepsilon \end{aligned}$$

Rechnung einer Grammatik

Definition 6.10 (Ableitung, Rechnung)

Gegeben:

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

Es gilt

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

falls

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

Rechnung einer Grammatik

Schreibweise für Ableitung

$$w \Longrightarrow_G^* w'$$

falls es Wörter $w_0, \ldots, w_n \in (V \cup T)^*$ $(n \geq 0)$ 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 $w_0, \ldots, w_n$ heißt Ableitung oder Rechnung

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

Vorsicht: Indeterminismus

Beispiel 6.11 (Indeterminismus)

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

$$\begin{aligned} R_0 &= S \rightarrow aBBc \\ R_1 &= B \rightarrow b \\ R_2 &= B \rightarrow ba \\ R_3 &= BB \rightarrow bBa \end{aligned}$$

Drei Möglichkeiten, das Wort *abbac* zu erzeugen:

$$\begin{array}{lllllll} S & \Longrightarrow_{R_0} & aBBc & \Longrightarrow_{R_1} & abBc & \Longrightarrow_{R_2} & abbac \\ S & \Longrightarrow_{R_0} & aBBc & \Longrightarrow_{R_2} & aBbac & \Longrightarrow_{R_1} & abbac \\ S & \Longrightarrow_{R_0} & aBBc & \Longrightarrow_{R_3} & abBac & \Longrightarrow_{R_1} & abbac \end{array}$$

Vorsicht: Indeterminismus

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!

Vorsicht: Indeterminismus

Beispiel 6.12 (Mehrdeutige Grammatik natürlichsprachlicher 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.

Erzeugte Sprache, Äquivalenz

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:

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

Definition 6.14 (Äquivalenz)

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

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