Grundlagen der Informatik

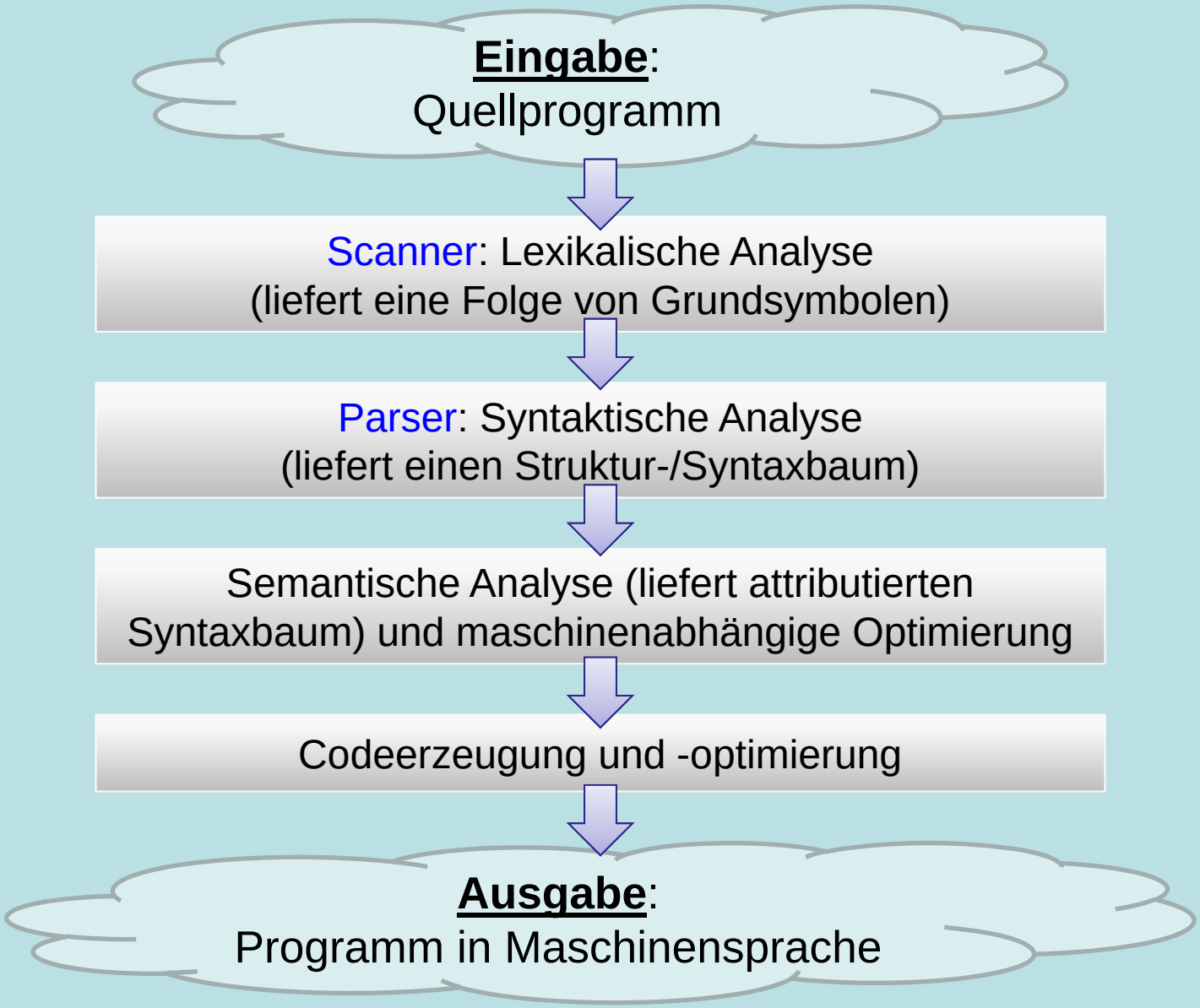
- Lexikalische und Syntaktische Analyse -

Prof. Dr. Klaus Volbert

Hochschule für angewandte Wissenschaften
Fakultät Informatik und Mathematik

Wintersemester 2010/11
Regensburg, 16. November 2010

Compiler


Eingabe:
Quellprogramm
Scanner: Lexikalische Analyse
(liefert eine Folge von Grundsymbolen)
Parser: Syntaktische Analyse
(liefert einen Struktur-/Syntaxbaum)
Semantische Analyse (liefert attributierten
Syntaxbaum) und maschinenabhängige Optimierung
Codeerzeugung und -optimierung
Ausgabe:
Programm in Maschinensprache

Lexikalische und Syntaktische Analyse

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

- Mittels formaler Sprachen soll entschieden werden, ob ein Element zu einer vorgegebenen Klasse gehört oder nicht
- Wie kann die Menge aller lexikalisch und syntaktisch korrekten C/C++- Programme beschrieben werden?
- Sei L die Menge aller syntaktisch korrekten C/C++-Programme, wie kann dann für ein Programm p entschieden werden, ob $p \in L$ (Wortproblem)?
- Sprachen können durch Grammatiken beschrieben werden
- Grammatiken geben Regeln (Produktionen) an, nach denen syntaktisch korrekte Sätze aufgebaut werden dürfen
- Anmerkung: Ein großer Teil der Arbeit eines Informatikers besteht in der Definition neuer Sprachen (z. B. Protokolle)

Begriffe

- Ein Alphabet $\sum$ ist eine endliche Menge von Zeichen ($\sum = \{0,1\}$)
- Ein Wort über einem Alphabet $\sum$ ist eine endliche Folge von Zeichen aus $\sum$, alle Wörter der Länge n: $\sum^n$
- Die Menge aller Wörter über $\sum$ ist $\sum^*$
- Das leere Wort ist ε, es gilt: $\varepsilon \in \sum^*$
- Wörter können zusammengesetzt werden (Konkatenation), es gilt: $\varepsilon u = u\varepsilon = u$, $\sum^+ = \sum\sum^* = \sum^* \backslash \{\varepsilon\}$ („mindestens einmal“)
- Die Länge eines Wortes ist die Anzahl der Zeichen, aus der sich ein Wort zusammensetzt, Notation: $|u|$, es gilt: $|\varepsilon| = 0$
- Eine Sprache L über einem Alphabet $\sum$ ist eine Menge von Wörtern über $\sum$, d.h. $L \subseteq \sum^*$
- Anmerkung:
  - Ein Programm ist ein Wort, d.h. hier weicht der Begriff von dem Wortbegriff in natürlichen Sprachen (z.B. deutsch) ab

Grammatiken zur Erzeugung von Sprachen

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

- Eine Grammatik G ist beschrieben durch ein 4-Tupel $G = (V, \sum, P, S)$ mit:
  - V ist ein endliches Alphabet von Variablen (Nicht-Terminale)
  - $\sum$ ist ein endliches Alphabet von Terminalen
  - $S \in V$ ist das Startsymbol
  - $P \subseteq ((V \cup \sum)^+ \times (V \cup \sum)^*)$ ist eine endliche Menge von Produktionen (Produktionen werden auch Regeln genannt)
- Ableitbarkeit
  - $b \in (V \cup \sum)^*$ ist (direkt) ableitbar aus $a \in (V \cup \sum)^+$, $a \to b$, wenn es eine Regel $(u, v) \in P$ und $\alpha, \beta \in (V \cup \sum)^*$ gibt: $a = \alpha\, u\, \beta$, $b = \alpha\, v\, \beta$
  - b ist aus a (indirekt) ableitbar, $a \overset{*}{\to} b$, wenn b durch endlich viele direkte Ableitungsschritte aus a erzeugbar ist
- Die von G erzeugte Sprache ist

$$L(G) = \{w \in \sum^* \mid S \overset{*}{\to} w\}$$

Beispiele

- $G_1 = (V, \sum, P, S)$ mit
  - $V = \{S, A, B\}$
  - $\sum = \{0,1\}$
  - $P = \{S \to ASB, A \to 0, B \to 1, S \to \varepsilon\}$
- Welche Sprache definiert G_1, also was ist $\mathrm{L} = \mathrm{L}(G_1)$?

- $G_2 = (V, \sum, P, S)$ mit
  - $V = \{S, R, L\}$
  - $\sum = \{0,1\}$
  - $P = \{S \to 01L0, 1L \to L11, 0L \to 0R, R1 \to 1R, R0 \to L0, L0 \to 0\}$
- Was ist $\mathrm{L} = \mathrm{L}(G_2)$?