# Grundlagen der Informatik

## - Lexikalische und Syntaktische Analyse Forts. -

Prof. Dr. Klaus Volbert

HOCHSCHULE REGENSBURG UNIVERSITY OF APPLIED SCIENCES

Hochschule für angewandte Wissenschaften
Fakultät Informatik und Mathematik

Wintersemester 2010/11
Regensburg, 23. 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

# *Chomsky-Hierarchie*

- Eine Grammatik $G = (V, \sum, P, S)$ heißt vom Typ Chomsky-0.
- $G$ heißt kontextsensitiv (vom Typ Chomsky-1), wenn für jede Regel $a \rightarrow b$ gilt: $|a| \leq |b|$ (Ausnahme: $A \rightarrow \varepsilon$ wird zugelassen)
- G heißt kontextfrei (vom Typ Chomsky-2), wenn alle Regeln die Form $A \rightarrow b$ mit $A \in V$ haben
- G heißt regulär (vom Typ Chomsky-3), wenn alle Regeln die Form $A \rightarrow \varepsilon$ oder $A \rightarrow aB$ mit a $\in \sum$ und $A, B \in V$ haben
- Eine Sprache heißt kontextsensitiv, kontextfrei oder regulär, wenn sie durch eine entspr. Grammatik erzeugt werden kann
- Sei $L_i$ die Menge der Typ Chomsky-$i$ Sprachen, dann gilt:

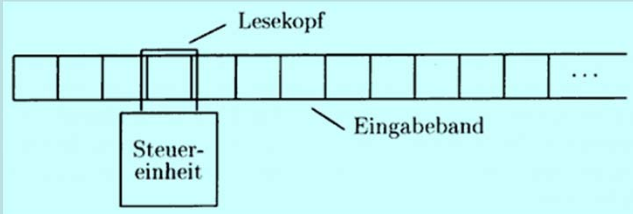
$$L_3 \subseteq L_2 \subseteq L_1 \subseteq L_0$$

(Anmerkung: Chomsky-Hierarchie ist sogar echt)

- Welche besondere Struktur haben reguläre Sprachen?

# *Deterministischer endlicher Automat*

- Ein deterministischer endlicher Automat (DEA/DFA) ist beschrieben durch ein 5-Tupel $\mathrm{M} = (Q, \sum, \delta, q_0, F)$ mit:
  - $Q$ ist eine endliche, nichtleere Menge von Zuständen
  - $\sum$ ist ein endliches, nicht leeres Eingabealphabet
  - $\delta: Q \times \sum \rightarrow Q$ ist die Übergangsfunktion
  - $q_0 \in Q$ ist der Startzustand
  - $F \subseteq Q$ ist die Menge der akzeptierenden Endzustände




- Akzeptanz
  - Ein DFA akzeptiert eine Eingabe $x_1, \ldots, x_n$, wenn gilt:

$$q_0 x_1, \ldots, x_n \overset{*}{\rightarrow} x_1, \ldots, x_n q \text{ mit } q \in F$$

  - Alternativ durch induktive Fortsetzung von $\delta$ auf $Q \times \sum^* \rightarrow Q$:

$$\delta(q, \varepsilon) = q \text{ für n > 0, sonst } \delta(q, x_1, \ldots, x_n) = \delta(\delta(q, x_1, \ldots, x_{n-1}), x_n)$$

  - Damit: Ein DFA akzeptiert $\mathrm{x} = x_1, \ldots, x_n \Leftrightarrow \delta(q_0, x) \in F$

# *Anmerkungen*

- Satz: Eine Sprache L ist genau dann regulär, wenn es einen DFA gibt, der L akzeptiert
- Scanner in Compilern implementieren DFAs zur lexikalischen Analyse
  - Je einen für Zahlen, Schlüsselwörter, Namen, Sonderzeichen, …
  - Die vorgefundenen Symbole (Token) werden in Tabellen abgelegt
- Bei der lexikalischen Analyse reicht es aus, einmalig von „links nach rechts“ über die Eingabe zu gehen, d.h. Zeichen für Zeichen der Eingabe zu lesen und zu verarbeiten
- Kann ein DFA eine Syntaxanalyse durchführen?
  - Können Klammerausdrücke auf Korrektheit geprüft werden?

# *Pumping-Lemma*

- Sei $L$ eine reguläre Sprache, dann gibt es eine Konstante $n \in \mathrm{IN}$, so dass sich jedes $\mathrm{z} \in L$ mit $|\mathrm{z}| \geq n$, so als $\mathrm{z} = \mathrm{uvw}$ schreiben lässt, dass $|uv| \leq n, |v| \geq 1$ und $uv^iw \in L$ für alle $i \geq 0$ gilt
- Beweisidee:
  - Ein Zustand muss mehrmals vorkommen
  - An dem Zustand können beliebige Schleifen durchlaufen werden
- Bemerkung:
  - Eine ähnliche Aussage gilt auch für kontextfreie Sprachen
- Anwendungsbeispiele:
  - $L_1 = \{0^k 1^k \mid k \geq 1\}$ ist nicht regulär
  - $L_2 = \{0^k | \ k \text{ ist Quadratzahl}, k \geq 1\}$ ist nicht regulär
- Was bedeutet das für die Auswertung von Programmen?