Compiler Construction

Chapter 2 – Syntactical Analysis

Prof. Dr. Jörg Kreiker

joerg.kreiker@informatik.hs-fulda.de

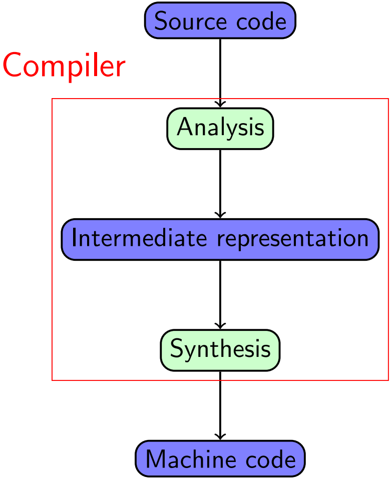
Fachbereich Angewandte Informatik
Hochschule Fulda – University of Applied Sciences

April 27, 2017

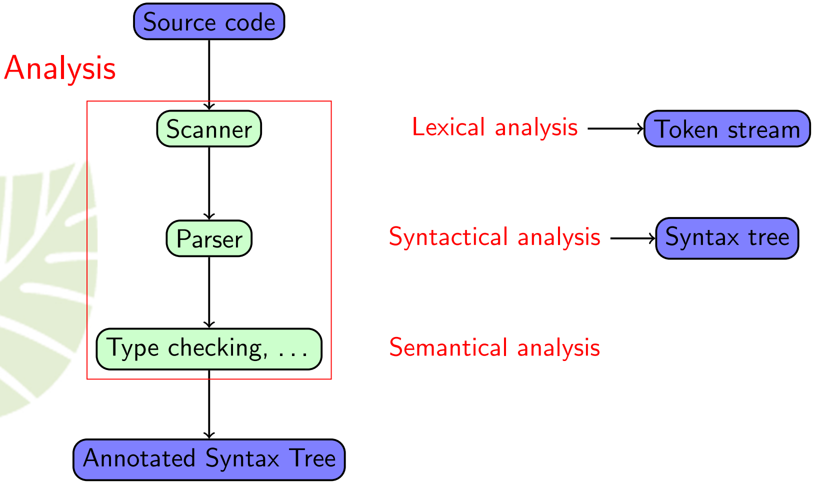
Agenda

1. Context-free grammars
2. Left-derivations
3. Pushdown automata
4. Item Pushdown Automata
5. $LL(1)$ grammars
6. Top-down parsing

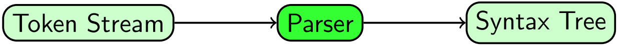
Compiler Overview



Analysis Phase



Syntactical Analysis



- During syntactical analysis, tokens are combined into large program units
- Examples
  - Expressions
  - Statements
  - Branches
  - Loops
- Parsers are generated just like scanners
  - **Specification of hierarchical structure** context-free grammar
  - **Generated Implementation** pushdown automaton + X

Context-free grammars

- Programs may contain an unbounded number of tokens but only a limited number of token classes
- The set T of token classes is our finite alphabet of terminal symbols
- A context-free grammar (CFG) is a quadruple (N, T, P, S), where
  - N is the set of non-terminal symbols
  - T is the set of terminal symbols
  - P is the set of production or rules
  - $S \in N$ is the start symbol
- Productions are of the form

$$A \rightarrow \alpha \qquad \mathrm{w}here \qquad A \in N, \alpha \in (N \cup T)^*$$

Example

Context-free grammar $(\{S\}, \{a, b\}, P, \{S\})$, where P has the two rules

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow \varepsilon \end{aligned}$$

specifies the language $\{a^b b^n \mid n \geq 0\}$

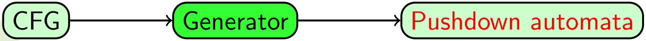
Derivations

- CFG are word replacement systems.
- Rules specify possible replacements.
- A sequence of such replacements is called a derivation
- The language of the CFG is the set of all words (of terminal symbols) that can be derived from the start symbol
- Derviations can be represented as a derivation tree
  - root: start symbol
  - inner nodes: replacements
  - leafs: terminal symbols

Left Derivations

- A CFG is called unique, iff there is at most 1 derivation tree for every word over the respective alphabet
- Grammars describing programming languages should be unique
- A derivation is called a left derivation, iff the leftmost non-terminal is replaced in each replacement
- Left derivations correspond to top-down construction of the syntax tree

Syntactical Analysis



Pushdown automata

A pushdown automaton (PDA) is a tuple $(Q, T, \delta,_{0}, F)$, where

- Q is a finite set of states
- T is the input alphabet
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of accepting states
- $\delta \subseteq Q^+ \times (T \cup \{\varepsilon\}) \times Q$ is a finite set of transitions

PDA computations

- The current configuration of a PDA is a pair $(\gamma, w) \in Q^* \times T^*$, where
- γ is the content of the stack, and
- w is the remainder of the input
- A computation step is characterized by

$$(\alpha\gamma, xw) \vdash (\alpha\gamma', w) \qquad \textit{for} \qquad (\gamma, x, \gamma') \in \delta$$

- The language accepted by a PDA is then

$$\{w \in T^* \mid \exists f \in F : (q_0, w) \vdash^* (f, \varepsilon)\}$$

- A PDA accepts by accepting state AND empty stack