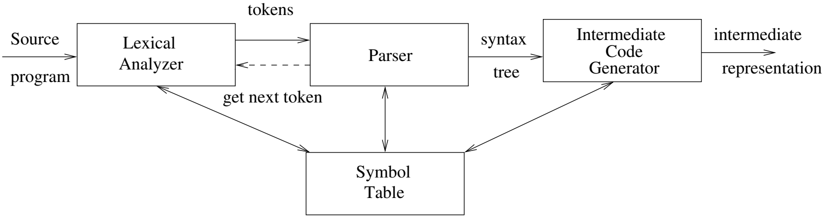
Syntax Analysis

Sudakshina Dutta

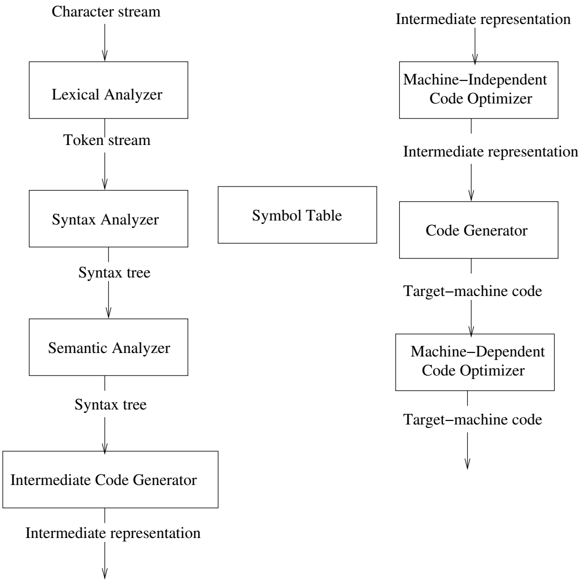
IIT Goa

4^{th} February, 2022

- Every programming language has precise grammar rules that describe the syntactic structure of well-formed programs and are useful in detecting errors
- Parsers obtains strings of tokens from the lexical analyzer and verifies that the string can be generated by the grammar of the source language
- It constructs parse trees/syntax trees and passes it to the rest of compilers for further processing



The Phases of a Compiler



Context-free Grammars

- A CFG is denoted as $G = (N, T, P, S)$
  - N : Finite set of non-terminals
  - T : Finite set of terminals
  - $S \in N$: The start symbol
  - P : Finite set of productions of the form $A \rightarrow \alpha$ where $A \in N$ and $\alpha \in (N \cup T)^*$
- Example :

$$
\begin{gathered}
E \rightarrow E + T | T \\
T \rightarrow T * F | F \\
F \rightarrow (E) | id
\end{gathered}
$$

- In this example, E represents expression consisting of terms separated by $+$ signs, terms consist of factors separated by $*$ signs and F represents factors

Derivations

- Consider the following grammar :
$$E \to E + E | E * E | - E | (E) | id$$
- Example of derivation : $E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(id)$
- If $S \xrightarrow{*} \alpha$, where S is the start symbol of the grammar G, we say α is the sentential form of G and it is generated in zero or more steps
- If $S \xrightarrow{+} \alpha$, then α is generated in one or more steps
- A sentence of G is a sentential form with no nonterminal
- The language generated by a grammar is a set of sentences
  — Context-free grammar generates context-free language

Context-free Languages

- Context-free grammars generate context-free languages
- The language generated by G, denoted $L(G)$, is
  $L(G) = \{w | w \in T^* \ and \ S \Rightarrow^* w\}$
- In other words, a string is in $L(G)$ if
  1. the string consists of terminals
  2. the string can be derived from S
- A string $\alpha \in (N \cup T)^*$ is a sentential form if $S \Rightarrow^* \alpha$
- Two grammars G_1 and G_2 are equivalent, if $L(G_1) = L(G_2)$

Context-free Languages

- Examples :
  1. $L(G_1) = \{a^n b^n | n \geq 0\}$
     - $A \rightarrow aAb|\epsilon$
  2. $L(G_2) = \{x|\ x \text{ has equal no of 0 and 1}\}$
     - $A \rightarrow 0A1|1A0|\epsilon$

Derivations

- **Leftmost Derivation** In this form of derivation, left-most non-terminal in each sentential form is always chosen
- Example:
  $E \xrightarrow{lm} -E \xrightarrow{lm} -(E) \xrightarrow{lm} -(E+E) \xrightarrow{lm} -(id+E) \xrightarrow{lm} -(id+id)$
- **Rightmost Derivation** In this form of derivation, right-most non-terminal in each sentential form is always chosen
- Rightmost derivations are sometimes called canonical derivation