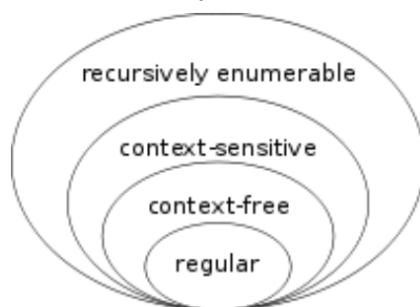


Aula 16 - Análise sintática: Gramática, Linguagens e Ambiguidade

Análise Sintática

- Processo que verifica se uma determinada entrada corresponde ao de uma gramática
- Seja $G1$ uma gramática, $L(G1)$ uma linguagem definida por $G1$ e “ a ” uma sentença de entrada. Então, um analisador sintático é uma ferramenta capaz de dizer se “ a ” $\in L(G1)$
- Hierarquia de Chomsky



- Reconhecedores para cada nível da Hierarquia de Chomsky

Tipo	Nome	Exemplo	Reconhecedor	Complexidade
0	Rekursivamente Enumerável	Estrutura de Frase	Maquina de Turing	<i>Undecidable</i>
1	Linguagem Sensível Cont.	$a^n b^n c^n$	Aut. Linearm. Delimitado	NP-Completo
2	Linguagem Livre de Contexto	$a^n b^n$	Automato a Pilha	$O(n^3)$
3	Linguagem Regular	$a^n b$	Automato Finito	$O(n)$

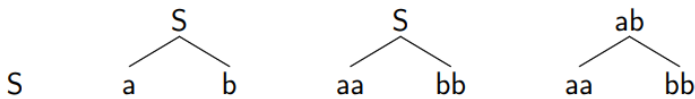
Linguagens Livres de Contexto Determinísticas

- Linguagens livres de contexto determinísticas são um subconjunto das LLC onde as linguagens não são ambíguas
- Todas as linguagens de programação pertencem a esta classe
- Para se especificar uma linguagem de programação, é necessário formalizar sua sintaxe, semântica e alfabeto.
 - Para especificar a sintaxe, usa-se a BNF (Formalismo de Backus-Naur)
 - Para especificar a semântica, usa-se regras informais. Por exemplo, associar o símbolo “*” com a operação de multiplicação
 - Alfabeto (Σ): conjunto finito e não vazio de símbolos
 - Cadeia (palavra ou sentença): sequência finita de símbolos do alfabeto.
 - Σ^* = conjunto de todas as sentenças de um alfabeto
- Uma linguagem é um subconjunto de Σ^*

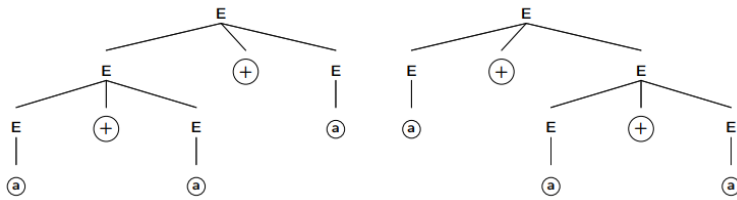
Gramática

- Uma gramática livre de contexto é definida pela 4-tupla $G=\{T, V, P, S\}$ onde
 - T = Símbolos terminais
 - V = Símbolos não terminais / variáveis
 - P = Produção ou regras

- S = Símbolo inicial
- Exemplo
 - $G1 = \{\{a, b\}, \{S\}, \{S \rightarrow ab|aSb\}, S\}$
 - Derivações: $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$
 - Linguagem: $L(G1) = \{a^n b^n, n \geq 1\}$
- Árvore de derivação
 - Uma árvore de derivação é uma alternativa gráfica para mostrar o processo de derivação de uma sentença em uma gramática
 - $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$



- Gramática Ambígua
 - Uma gramática é dita ambígua se existe uma sentença para a qual existe mais de uma árvore de derivação
 - Exemplo: $G2 = \{E \rightarrow a|E + E\}, \alpha = a + a + a$



- Para eliminar a ambiguidade de uma gramática $G1$, deve-se reescrever a gramática para uma nova gramática $G2$ não ambígua tal que $L(G1)=L(G2)$. Exemplo: $G1 = \{A \rightarrow Aa|aA|a\}$ para $G2 = \{A \rightarrow Aa|a\}$

Contexto em compiladores

- Em compiladores, o alvo é um subconjunto de linguagens livres de contexto chamadas “Linguagens livres de contexto determinísticas”, que não são ambíguas, são o conjunto de linguagens do qual fazem parte todas as linguagens de programação e os reconhecedores tem complexidade $O(n)$.
- Seja α uma sentença e G uma gramática para uma LLCD. O que será estudado aqui são mecanismos para reconhecer se $\alpha \in L(G)$.
- Para tal, existem duas abordagens: top-down e bottom-up.