

Construção de Compiladores

Período Especial

Aula 15: Análise Sintática:

Gramáticas, Linguagens e Ambiguidade

Bruno Müller Junior

Departamento de Informática
UFPR

2020

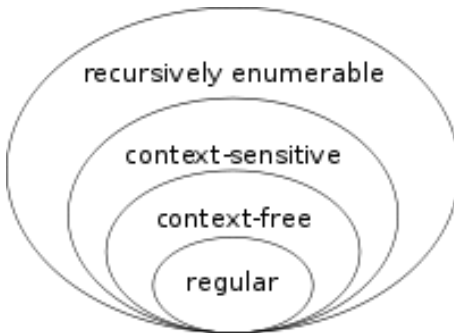
Análise Sintática

- A análise sintática (*parsing*) é um processo que verifica se uma determinada entrada (sentença) corresponde ao de uma gramática.
 - Seja $G1$ uma gramática;
 - Seja $L(G1)$ a linguagem definida por $G1$;
 - Seja α uma sentença de entrada.
 - Então, formalmente, um analisador sintático é uma ferramenta capaz de dizer se:

$$\alpha \in L(G1)$$

Hierarquia de Chomsky

- é uma classificação de gramáticas formais descrita em 1959 pelo linguista Noam Chomsky.





Reconhecedores

Tipo	Nome	Exemplo	Reconhecedor	Complexidade
0	Recursivamente 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.
- A teoria (e prática) de compiladores trata desta classe.
- Todas as linguagens de programação pertencem a esta classe.
- O que será estudado



Definições

- Para se especificar uma linguagem de programação, é necessário formalizar sua sintaxe, semântica e alfabeto.
 - Para especificar a sintaxe, usa-se a BNF.
 - Para especificar a semântica, usa-se regras informais. Por exemplo, associar o símbolo “*” com a operação de multiplicação.
 - alfabeto: conjunto finito e não vazio de símbolos, por exemplo:
 $\Sigma = \{a, b, *, (,)\}$
- Cadeia (palavra ou sentença): sequência finita de símbolos de Σ .
- Um caso importante é o conjunto de todas as sentenças de um alfabeto, que é indicada por Σ^*

Linguagens

- Uma linguagem é um subconjunto de Σ^* .
- Exemplos de linguagens para $\Sigma = \{a, b\}$.
 - $L_1 = \emptyset$
 - $L_2 = \{\alpha \in \Sigma, |\alpha| \leq 2\}$
 - $L_3 = \{a^n, n \geq 2\}$
 - $L_4 = \{a^n b^n, n \geq 1\}$
- As linguagens L_1 e L_2 são linguagens finitas.
- As linguagens L_3 e L_4 são linguagens infinitas.



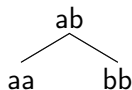
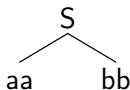
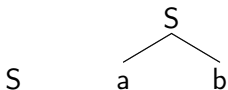
Gramática

- Uma gramática livre de contexto é definida pela 4-tupla $G = \{T, V, P, S\}$, onde:
 - T Símbolos terminais;
 - V Símbolos não terminais (ou “variáveis”)
 - P Produções ou regras: é uma relação finita de N para $(T \cup N)^*$
 - S Símbolo inicial
- Exemplo: $G_1 = (\{a, b\}, \{S\}, \{S \rightarrow ab|aSb\}, S)$
- Derivações: $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$
- Linguagem: $L(G_1) = \{a^n b^n, n \geq 1\}$



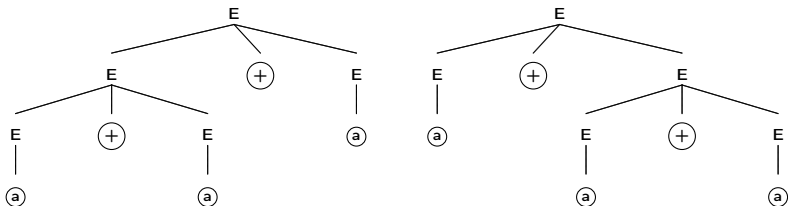
Árvore de Derivação

- Uma árvore de derivação é uma alternativa gráfica para mostrar o processo de derivação de uma sentença em uma gramática.
- Seja $G_1 = (\{a, b\}, \{S\}, \{S \rightarrow ab | aSb\}, S)$ e as derivações $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$



Gramática Ambígua

- Uma gramática é dita ambígua se existe uma sentença para a qual existe mais de uma árvore de derivação.
- Exemplo: $G_2 = \{E \rightarrow a \mid E + E\}$, $\alpha = a + a + a$





Linguagem Ambígua

- Para eliminar a ambiguidade de uma gramática G_1 , deve-se reescrever a gramática para uma nova gramática G_2 não ambígua tal que $L(G_1)=L(G_2)$. Exemplo:

$$G_1 = \{A \rightarrow Aa|aA|a\}$$

$$G_2 = \{A \rightarrow Aa|a\}$$

- Quando não existe uma G_2 não ambígua, dizemos que a linguagem é ambígua. Exemplo:
 $L = \{a^i b^j c^k | i, j, k \geq 1 \text{ e } i = j \text{ ou } j = k\}$

Contexto

- Gramáticas, linguagens e ambiguidades (entre outros) são o alvo de uma área da teoria da computação que lida com linguagens formais e autômatos.
- Em compiladores, o alvo é um subconjunto de linguagens livres de contexto chamadas “Linguagens livres de contexto determinísticas”, que:
 - 1 não são ambíguas;
 - 2 são o conjunto de linguagens do qual fazem parte todas as linguagens de programação.
 - 3 os reconhecedores tem complexidade $O(n)$;

Objetivo

- 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*.
- Elas serão vistas nas próximas aulas.