



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

COMPILADORES

Tipos de Analisadores
Hierarquia de Chomsky

Docentes: Ruben Moisés Manhiça
Cristiliano Arlindo Maculuve

Maputo, 28 de abril de 2023



Conteúdo da Aula

1. Analise Sintática (Continuação);
2. Topologia





Análise Sintática

Cada linguagem de programação possui as regras que descrevem a estrutura sintática dos programas bem-formados.

Em Pascal, por exemplo,

um programa é constituído por blocos,
um bloco por comandos,
um comando por expressões,
uma expressão por *tokens*
e assim por diante.





Gramática

A sintaxe das construções de uma linguagem de programação pode ser descrita pelas **gramáticas livres de contexto (tipo 2)**.

As gramáticas oferecem vantagens significativas tanto para os projetistas de linguagens quanto para os escritores de compiladores.





Gramática

- Uma gramática oferece, para uma linguagem de programação, uma especificação sintática precisa e fácil de entender.
- Para certas classes de gramáticas, pode-se construir automaticamente um analisador sintático que determine se um programa-fonte está sintaticamente bem-formado.
- Como benefício adicional, o processo de construção do analisador pode revelar ambiguidades sintáticas bem como outras construções difíceis de se analisar gramaticalmente.
- Uma gramática propriamente projetada implica uma estrutura de linguagem de programação útil à tradução correta de programas-fonte em códigos-objeto e também à detecção de erros.





Gramática

As linguagens evoluíram ao longo de um certo período de tempo, adquirindo novas construções e realizando tarefas adicionais.

Essas novas construções podem ser mais facilmente incluídas quando existe uma implementação baseada numa descrição gramatical da linguagem.





Papel do Analisador Sintático

No modelo dos compiladores, o analisador sintático

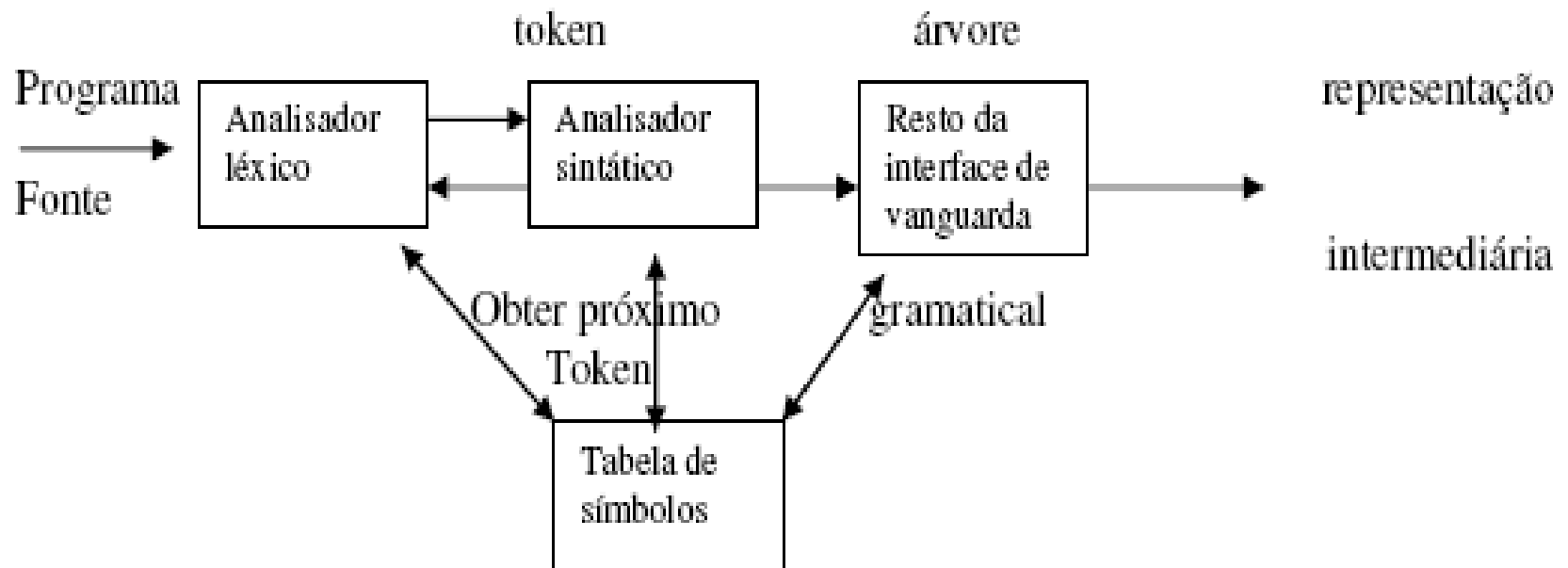
Obtém uma cadeia de *tokens* proveniente do analisador léxico, como mostrado na Figura a seguir;

Verifica se a cadeia pode ser gerada pela gramática da linguagem-fonte;

Espera-se que o analisador sintático relate quaisquer erros de sintaxe de uma forma inteligível.



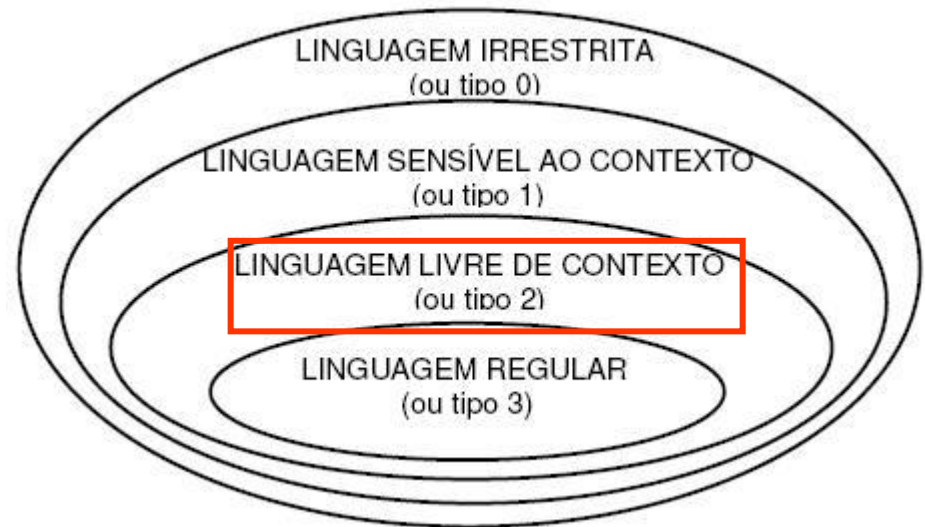
Papel do Analisador Sintático





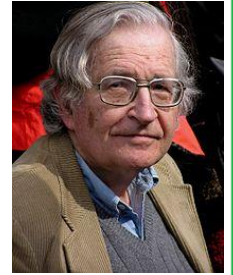
Análise Sintática

- A análise sintática é mais complexa em natureza do que a análise léxica
 - Precisamos de uma linguagem mais avançada
- Hierarquia de Chomsky





Hierarquia de Chomsky



Hierarquia de Chomsky é a classificação de gramáticas formais descrita em 1959 pelo linguista Noam Chomsky. Esta classificação possui 4 níveis, sendo que os dois últimos níveis (os níveis 2 e 3) são amplamente utilizados na descrição de linguagem de programação e na implementação de interpretadores e compiladores.





Tipos de Gramática

1) Gramática do tipo 0 (Irrestritas)

São aquelas às quais nenhuma limitação é imposta, exceto a existência de pelo menos 1 símbolo não terminal do lado esquerdo das regras.

Todo o universo das linguagens é gerado por gramáticas deste tipo.

As produções são do tipo

$$\alpha \rightarrow \beta \text{ onde } \alpha \in V^* V_n V^* \text{ e } \beta \in V^* \quad V = \{V_n \cup V_t\}$$





Tipos de Gramática

2) Gramática do tipo 1 (Sensíveis ao Contexto)

Se nas regras de produção for imposta a restrição de nenhuma substituição é aplicada se causar a redução no tamanho da cadeia gerada, cria-se a classe de gramáticas sensíveis ao contexto.

As produções são do tipo

$$\alpha \rightarrow \beta \text{ onde } \alpha \in V^+ V_{\Pi} V^+ \text{ e } \beta \in V^+ \text{ e } |\beta| \geq |\alpha|$$

Exemplo :

$aA bcD \rightarrow a bcD bcD$ é tipo 1 $A bc \rightarrow abD bc$ é tipo 1 $AC \rightarrow A\lambda$ não é tipo 1





Tipos de Gramática

3) Gramática do tipo 2 (Livres de Contexto)

Apresentam uma restrição adicional de que as regras de produção tem apenas 1 elemento não terminal do lado $\epsilon \alpha \rightarrow \beta$, $\alpha \in V_n$ e $\beta \in V^+$

Exemplo :

$$G = (\{A,B,C\}, \{a,b,c\}, P, A)$$

$$P: \begin{cases} A \rightarrow BC \mid a \\ B \rightarrow b \\ C \rightarrow c \end{cases}$$

$$L(G) = \{a, b, c\}$$





Tipos de Gramática

4) Gramática do tipo 3 (Regulares)

Possui a restrição de que as produções substituem um não terminal por uma cadeia de um terminal seguido (direita) (ou precedido (esquerda)), ou não, por um único não terminal.

Produções do tipo $A \rightarrow \alpha B$; $A \rightarrow \alpha$ ou $A \rightarrow B\alpha$; $A \rightarrow \alpha$

Exemplo $G = (V_n, V_t, P, S)$

$$P: \begin{cases} S \rightarrow aS \\ S \rightarrow bA \\ A \rightarrow c \end{cases} \quad L(G) = \{ w/w \in \{a, b, c\}^* \text{ e } w = a^n bc, n \geq 0 \}$$





Tipos de análises sintáticas

- Descendentes (top-down) ou
- Ascendentes (bottom-up).

Assume-se que a saída de um analisador sintático seja uma representação da árvore gramatical para o fluxo de *tokens* produzido pelo analisador léxico.

Na prática existe um certo número de tarefas que poderiam ser conduzidas durante a análise sintática, quais sejam:

Coletar informações sobre os vários *tokens* na tabela de símbolos;

Realizar a verificação de tipos e outras formas de análise semântica;

Gerar o código intermediário;





Tipos de análises sintáticas

Tanto os métodos descendentes como os ascendentes constroem a árvore de derivação *da esquerda para direita*.

A escolha das regras deve se basear na cadeia a ser reconhecida, que é lida da esquerda para a direita.

Imaginem um compilador que começasse a partir do fim do código-fonte!





Análise Sintática Ascendente

- Na análise sintática ascendente a construção da árvore de derivação para uma determinada cadeia (lexema) começa pelas folhas da árvore e segue em direção de sua raiz.
- Caso seja obtida uma árvore cuja raiz tem como rótulo o símbolo inicial da gramática, e na qual a sequência dos rótulos das folhas forma a cadeia dada, então a cadeia é uma sentença da linguagem, e a árvore obtida é a sua árvore de derivação.





Análise Sintática Ascendente

Embora a árvore de derivação seja usada para descrever os métodos de análise, na prática ela não precisa ser efetivamente construída.

A única estrutura de dados necessária para o processo de análise sintática é uma **pilha**.

Guarda informação sobre os nós da árvore de derivação relevantes em cada fase do processo





Análise Sintática Ascendente

Definição:

Redução de uma cadeia → partir das folhas em direção à raiz de uma árvore de derivação.

A cada passo, procura-se reduzir uma cadeia (ou subcadeia) ao seu símbolo de origem, objetivando-se atingir o símbolo inicial da gramática.

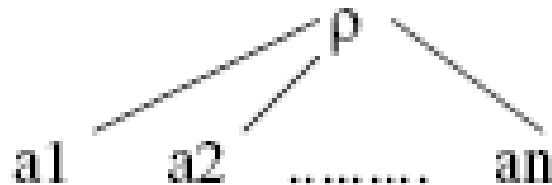




Análise Sintática Ascendente

Etapas da redução de uma cadeia

- 1) Toma-se uma determinada cadeia β .
- 2) Procura-se por uma sub-cadeia $\mathbf{a_1, a_2, \dots, a_n}$ de β que possa ser substituída pelo seu símbolo de origem ρ .
Ou seja, $\rho \rightarrow \mathbf{a_1, a_2, \dots, a_n}$. Assim $\mathbf{a_1, a_2, \dots, a_n} \beta b = a_1 \rho a_2$.





Análise Sintática Ascendente

3) Repetir o passo (2) até que β = **símbolo inicial** da gramática.

Caso isso não seja possível, tem-se que a cadeia analisada não pertence a linguagem especificada.





Análise Sintática Ascendente

Exemplo:

$G = (\{E, T, F\}, \{a, b, +, *, (,)\}, P, E)$

$P: E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow a \mid b \mid (E)$

Verificando se a cadeia $a+b*a$ pertence a linguagem utilizando-se de análise sintática ascendente.

$\underline{a}+b*a \Rightarrow \underline{F}+b*a \Rightarrow \underline{T}+b*a \Rightarrow E+\underline{b}*a \Rightarrow E+\underline{F}*a \Rightarrow E+T*\underline{a} \Rightarrow E+T*\underline{F} \Rightarrow$

$\begin{array}{ccccccc} | & | & | & | & | & | & | \\ a & F & T & b & F & & a \end{array}$

$E+T \Rightarrow E$

$\begin{array}{cc} | & | \\ T * F & E + T \end{array}$



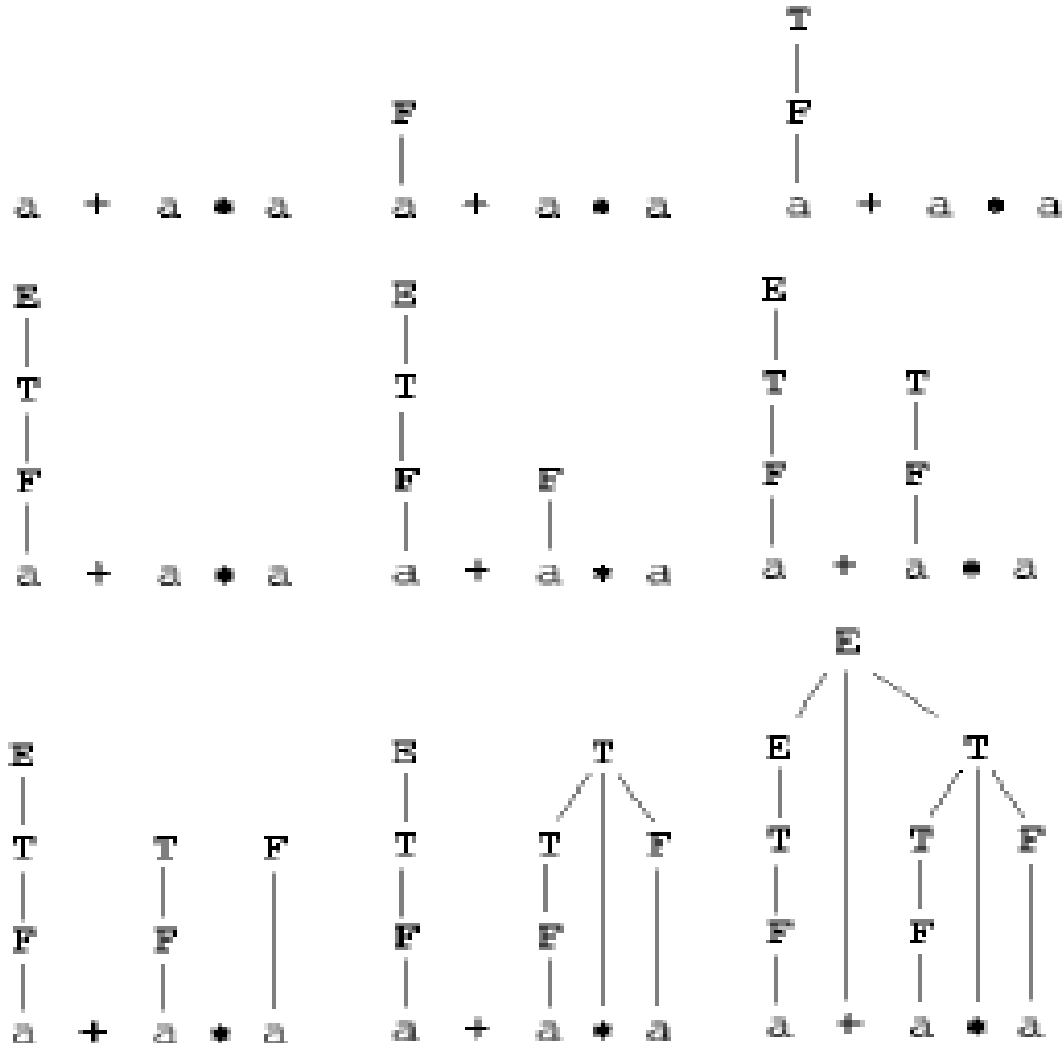


Análise Sintática Ascendente

Outro exemplo a
cadeia
 $x = a + a * a$ e
produção

1. $E \rightarrow E + T$
2. $E \rightarrow T$
3. $T \rightarrow T * F$
4. $T \rightarrow F$
5. $F \rightarrow (E)$
6. $F \rightarrow a$

Nesse caso, a
ordem das regras
corresponde
a uma derivação à
direita, porém de
trás para frente





Análise Sintática Ascendente

Problemas:

- Os maiores problemas na análise sintática ascendente residem na dificuldade de determinação de qual parcela da cadeia (sub-cadeia) que deverá ser reduzida;
- Qual será a produção a ser utilizada na redução, para o caso de existirem mais do que uma possibilidade.





Análise Sintática Descendente

Na análise sintática descendente tem-se procedimento inverso ao da ascendente.

Aqui a análise parte da raiz da árvore de derivação e segue em direção as folhas, ou seja,

a partir do símbolo inicial da gramática vai-se procurando substituir os símbolos não-terminais de forma a obter nas folhas da árvore a cadeia desejada.





Análise Sintática Descendente

Etapas da redução de uma cadeia

- 1) Toma-se uma determinada cadeia b e o símbolo inicial β da gramática.
- 2) Procura-se por uma regra de derivação que permita derivar β em sua totalidade ou pelo menos se aproximar desta.
- 3) Repetir o passo (2) até que a cadeia não apresente mais símbolos não terminais, verificando se a cadeia obtida coincide com β .

Não havendo coincidência, a cadeia analisada não pertence a linguagem especificada.





Análise Sintática Descendente

Exemplo

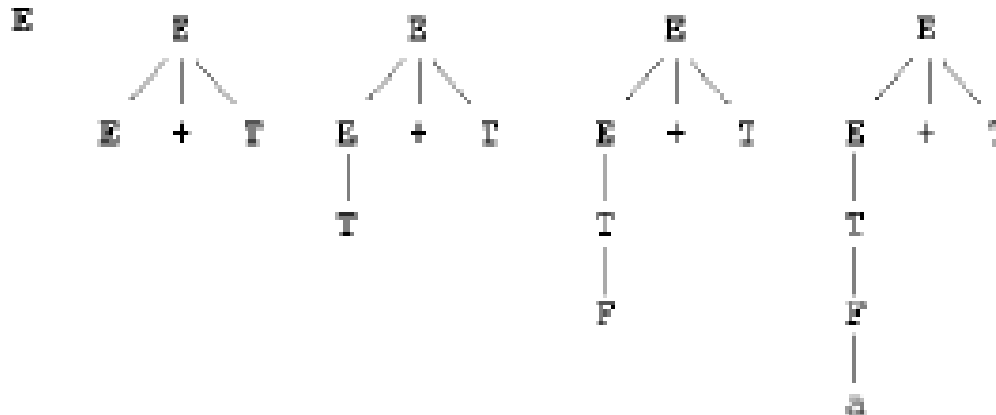
Considere a seguinte gramática

1. $E \rightarrow E + T$
2. $E \rightarrow T$
3. $T \rightarrow T * F$
4. $T \rightarrow F$
5. $F \rightarrow (E)$
6. $F \rightarrow a$

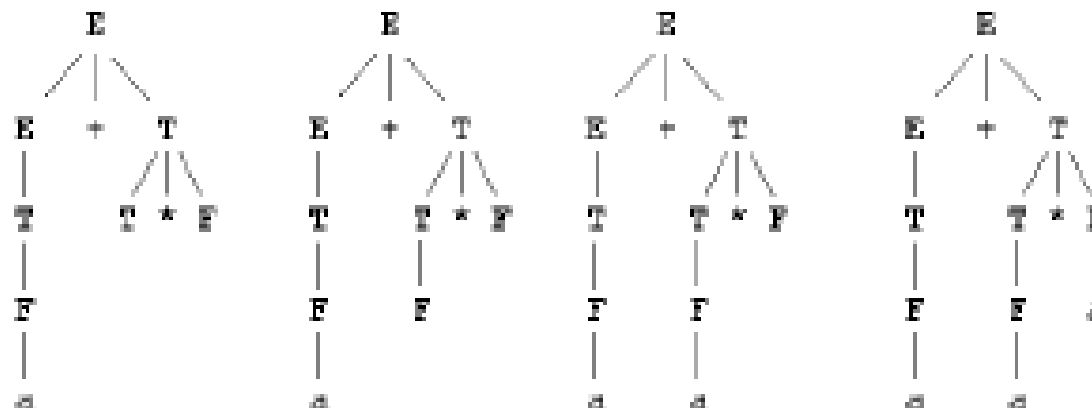
e a cadeia $x = a+a*a$



Análise Sintática Descendente



$E \rightarrow E + T$
 $E \rightarrow T$
 $T \rightarrow T * F$
 $T \rightarrow F$
 $F \rightarrow (E)$
 $F \rightarrow a$



Note que as regras são consideradas na mesma ordem em que as regras seriam usadas em uma derivação à esquerda





Análise Sintática Descendente

Problemas:

Embora apresente os mesmos problemas que a análise sintática ascendente, implementa facilmente compiladores para linguagens obtidas de gramáticas LL(1)

(análise da cadeia da esquerda para a direita e derivações esquerdas observando apenas o primeiro símbolo da cadeia para decidir qual regra de derivação será utilizada).





TPC

- Escolham 2 Colegas para vir falar das Funções
 - First();
 - Follow();

Recompensa: + 1 Valor no teste 1 ou 2





Para a proxima aula

- Resolução de Exercicios;
- Funções First() e Follow();



FIM!!!

Duvidas e Questões?

