

Compiladores e Computabilidade

Prof. Leandro C. Fernandes
UNIP – Universidade Paulista, 2018



(2ª fase da Análise)

ANÁLISE SINTÁTICA (PARSING)

A análise sintática



A análise sintática deve:

- Comprovar que a seqüência de tokens cumpre as regras gramaticais, gerando sua árvore gramatical.
- Tornar clara a estrutura hierárquica da evolução da sentença.
 - Exemplo: $A / B * C$ em Fortran: $(A/B) * C$
em APL: $A / (B * C)$
- Identificar erros de Sintaxe
 - Exemplo: $A * / B;$
- Recuperar erros de sintaxe;
 - Não retardar, de forma significativa, o processamento de programas corretos.

Observações/Curiosidade

- 60% dos programas compilados estão corretos tanto sintaticamente quanto semanticamente;
- 80% dos enunciados com erros apresentam apenas um erro;
- 13% dos enunciados com erros apresentam apenas dois erros;
- 90% dos erros envolvem um único token.



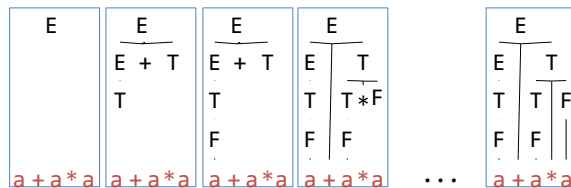
Especificando a estrutura sintática da Linguagem

- Gramáticas regulares não podem lidar com estruturas aninhadas ou com recursões centrais
 Ex: `Expr` \rightarrow ... "(" `Expr` ")" ...
 `Cmd` \rightarrow "do" `Cmd` "while" "(" `Expr` ")"
- Solução: Gramáticas Livres de Contexto
- Vantagens na utilização de gramáticas:
 - especificações sintáticas precisas de linguagens;
 - podemos usar um gerador automático de *parser*;
 - o processo de construção pode levar a identificar ambiguidades;
 - facilidade de ampliar/modificar a linguagem.

Tipos de Analisadores Sintáticos

Métodos Descendentes (*Top Down*):

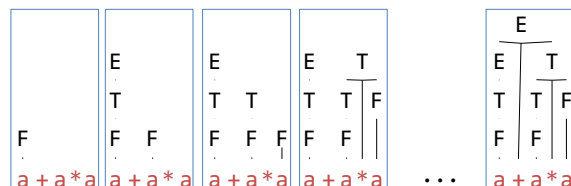
- Constroem a árvore sintática de cima para baixo (da raiz para as folhas), ou seja, do símbolo inicial da gramática para a sentença.
 - Analisadores Descendentes Recursivos
 - Analisadores LL(k)



Tipos de Analisadores Sintáticos

Métodos Ascendentes (*Bottom-up*):

- Constroem a árvore sintática de baixo para cima (das folhas para a raiz), ou seja, reduz os símbolo da sentença até alcançar o símbolo inicial da gramática.
 - Analisadores LR
 - Analisadores LALR



Tipos de Analisadores Sintáticos

- Tanto para a análise ascendente, quanto para a descendente a entrada é examinada da esquerda para a direita e, usualmente, um símbolo por vez;
 - Trabalham com subclasses de gramáticas. Em geral, as gramáticas são LL e LR
 - Na prática, utilizam-se LL(1) e LR(1)
- Muitos compiladores são dirigidos pela sintaxe (*parsen driver*)
 - Nele o analisador sintático é o módulo principal e realiza chamadas para o analisador léxico.



RECUPERAÇÃO DE ERROS

Recuperação de Erros

- Modo pânico ou Desespero:
 - Para imediatamente diante do primeiro erro; ou
 - identificado um erro, o analisador sintático descarta símbolos de entrada, até que seja encontrado um token pertencente ao subconjunto de tokens de sincronização;
 - tokens de sincronização: delimitadores etc.

Recuperação de Erros

- Recuperação de Frases:
 - ao descobrir um erro, o analisador sintático pode realizar uma correção local na entrada restante (substituir por alguma cadeia que permita a análise prosseguir)
 - exemplo: substituir uma vírgula inadequada por um ponto e vírgula, remover um “:” excedente;

Recuperação de Erros

- Produções de Erro:
 - aumenta-se a gramática, de forma a acomodar os erros mais comuns. Quando uma produção de erro é identificada pelo analisador, diagnósticos apropriados são apresentados;

Recuperação de Erros

- Correção Global:
 - usa algoritmos de escolha da sequência mínima de mudanças necessárias para se obter a correção global, com custo adequado.
 - exemplo: dada uma cadeia x , o parser procura árvores gramaticais que permitam transformar x em y (cadeia correta) com um mínimo de modificações.