Análise Sintática
(Cap. 04)
Conceitos e requisitos para um analisador sintático

Análise Sintática

- Métodos e técnicas para descrever e reconhecer a estrutura sintática de programas
  - Ex: em linguagem C, o Analisador Sintático (AS) deve reconhecer que um programa é composto de funções; funções são compostas de declarações e comandos, etc.

Análise Sintática

- Tratamento de erros sintáticos
  - Informar a presença de erros de forma clara e precisa
  - Recuperar cada erro com rapidez para detectar outros erros
  - Acrescentar custo mínimo no processamento de programas corretos
- Estratégias de recuperação de erros: modo pânico (ao detectar erro, o AS descarta símbolos de entrada até encontrar token de sincronismo), recuperação em nível de frase

Análise Sintática

- Utiliza Gramáticas Livres de Contexto (GLC) para descrever a sintaxe das construções de linguagem de programação como expressões e comandos
- Derivações: a partir do símbolo inicial de uma regra, substitui-se um não-terminal pelo corpo de uma de suas produções
  - O não-terminal em cada passo é escolhido:
    - Em derivações **mais à esquerda (lm)**, o não-terminal mais à esquerda em cada forma setencial sempre é escolhido
    - Em derivações **mais à direita (rm)**, o não-terminal mais à direita em cada forma setencial sempre é escolhido

Análise Sintática

- Ambiguidade: Uma gramática que produz mais de uma árvore de derivação para alguma sentença é considerada ambígua
- Uma GLC é ambígua se permitir mais de 1 derivação **mais à esquerda** ou mais de 1 derivação **mais à direita** para a mesma sentença

Análise Sintática

- Eliminação de recursão à esquerda
  - Métodos de análise descendente não tratam recursões à esquerda
  - Eliminação de recursão à esquerda imediata:
    - 1º: agrupe as produções
      - A -> Aa1 | Aa2 | ... | Aan | B1 | B2 | ... | Bn
      - Onde nenhum Bi (1<= i <= n) começa com A
    - 2º: substitua as produções-A por:
      - A-> B1A' | B2A' | ... |BnA'
      - A' -> a1A' | a2A' | ... | AnA' | ε

Análise Sintática

- Algoritmo para eliminar recursão à esquerda de dois ou mais passos:
  - Entrada: gramática G
  - Saída: gramática equivalente a G

    1) organizar não-terminais em ordem crescente A1, A2, ..., An

    2) para (cada i de 1 até n){

    3) para (cada j de 1 até i-1){

    4) substituir cada prod. da forma Ai-> Ajy por produções Ai-> d1y | d2y | ... | dky, onde Aj-> d1 | d2 | ... | dn são produções Aj

    5) eliminar todas as recursões à esquerda imediatas nas produções-Ai

Análise Sintática

- Aplicar algoritmo anterior para a gramática:
  - S-> Aa | b
  - A-> Ac | Sd | epsilon
  - K-> Md | Kb
  - M-> N | O | M
  - N-> Ke | z

Análise Sintática

- Fatoração à esquerda: procura resolver a ambiguidade inicial (da esquerda para a direita) entre regras de produção
- Algoritmo:
  - Entrada: gramática G
  - Saída: gramática G, fatorada à esquerda
    - 1) para cada não-terminal A, encontrar prefixo **a** mais longo e comum a duas ou mais de suas alternativas
    - 2) Se **a!=**ε, então existe um prefixo comum não-trivial, substitua todas as produções-A, A->aB1 | aB2 | ... | aBn | y, onde y representa todas as alternativas que não começam com **a**, por:
      - A-> aA' | y
      - A'-> B1 | B2 | ... | Bn