Tradutores

Análise Sintática – *Top Down*

Sumário

- Introdução
- Analisador Descendente com *Backtracking*
- Analisador Preditivo
  - Analisador Preditivo Recursivo
  - Analisador Preditivo Não Recursivo
- Recuperação de Erros

Análise *Top-Down*

- Construção da árvore sintática a partir da raiz em direção as folhas
  - Caminhamento em pré-ordem
  - Encontra a derivação mais a equerda para a entrada analisada
- Problema chave
  - Encontrar, a cada passo da análise, que produção deve ser aplicada para um não terminal

Análise *Top-Down*

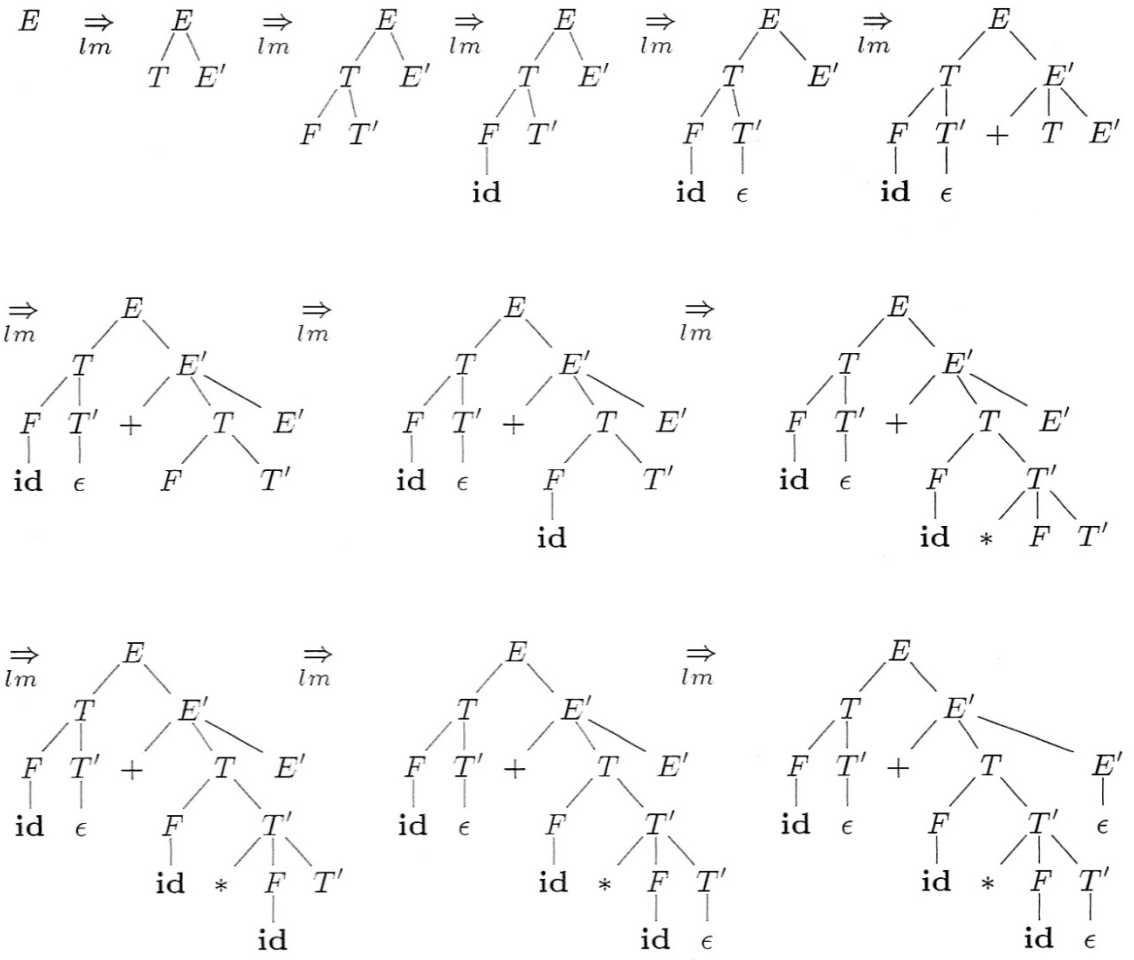
$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \varepsilon$

$F \rightarrow (E) \mid id$



Análise *Top-Down*

- Gramáticas LL(*k*)
  - Classe de gramáticas utilizadas na contruçaão de analisadores *top-down* utilizando *k* símbolos de *lookahead*
  - LL(1)
    - Tipo específico → apenas um símbolo de *lookahead*

Analisador Recursivo Descendente

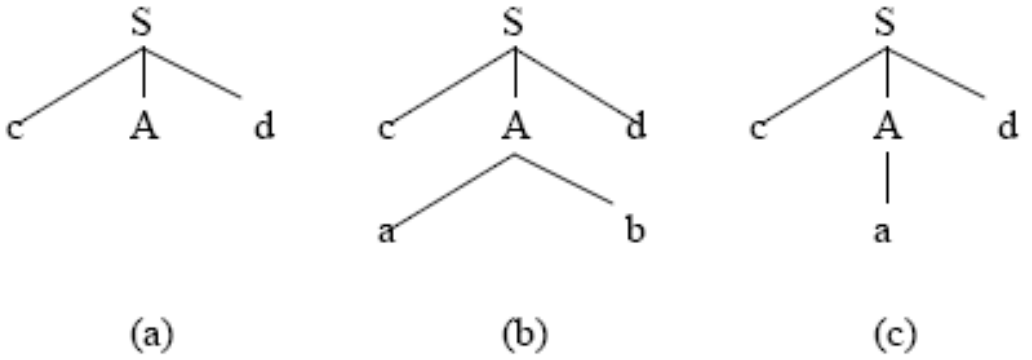
- Consiste de um conjunto de funções: uma para cada não terminal.
- Execução inicia pelo símbolo inicial.

```
void A(){
    Escolher uma produção A → X1,X2,…,Xk
   for ( i = 1 to k ) {
       if ( Xi é um não terminal ){
              executar a função Xi();
       }
       else if ( Xi é igual ao símbolo sendo analisado ){
              avançar o ponteiro da entrada para o próximo símbolo
       }
       else /* erro */ ;
   }
```

Analisador Recursivo com *backtracking*

- *Backtracking*
  - Voltar atrás no processo de reconhecimento e tentar produções alternativas
  - Entrada: *cad*

$$S \rightarrow cAd$$
$$A \rightarrow ab \mid a$$



Analisador Recursivo com *backtracking*

- Entrada `[a]`

$$
\begin{aligned}
S &\rightarrow a \\
&\mid [L] \\
L &\rightarrow S\,;\,L \\
&\mid S
\end{aligned}
$$

Exercício: Construir as etapas de construção da árvore de derivação .

Analisador Recursivo com *backtracking*

- Uma gramática recursiva a esquerda pode gerar um ciclo infinito de expansão de não terminais.

$$A \rightarrow Aa$$
$$\mid b.$$

- Um analisador *top-down* para esta gramática pode entrar em um laço infinito, expandindo repetidamente o terminal *A*

Analisador Preditivo Recursivo

- Tendo-se uma gramática sem recursão a esquerda e fatorada à esquerda, pode-se gerar um analisador que não necessita de *backtracking*
- Produções de um mesmo não terminal devem ter seus lados direitos iniciados por diferentes seqüências de caracteres

Analisador Preditivo Recursivo

- Função FIRST(β)
  - Se β =*=> ε, então ε é um elemento de FIRST(β)
  - Se β =*=> aB, então a é um elemento de FIRST(β)
- Sendo a um não terminal e β uma forma sentencial qualquer, podendo ser vazia.
- Dado um não terminal *A*, definido por várias alternativas que não iniciam por terminais, a implementação de um analisador preditivo recursivo para *A* exige que os conjuntos FIRST que iniciam as várias alternativas de produção sejam disjuntos.

Analisador Preditivo Recursivo

- Terminais *if*, *while* e *begin* permitem que a escolha da alternativa seja feita antecipadamente

$$
\begin{aligned}
comando \rightarrow\ & if\ expr\ then\ comando\ else\ comando \\
\mid\ & while\ expr\ do\ comando \\
\mid\ & begin\ listaComandos\ end.
\end{aligned}
$$

Analisador Preditivo Recursivo

comando → *condicional*
| *iterativo*
| *atribuição*

condicional → *if expr then comando*

iterativo → *repeat lista until expr*
| *while expr do comando*

atribuição → *id := expr*

```
procedure COMANDO;
begin
if token = 'if'
  then CONDICIONAL
  else if token = 'while' or 'repeat'
        then ITERATIVO
        else if token = 'id'
              then ATRIBUIÇÃO
              else ERRO
end;
```

Analisador Preditivo Recursivo

- FIRST(condicional) = {if}
- FIRST(iterativo) = {while, repeat}
- FIRST(atribuição) = {id}

Exercício: escrever as rotinas CONDICIONAL, ITERATIVO e ATRIBUICAO

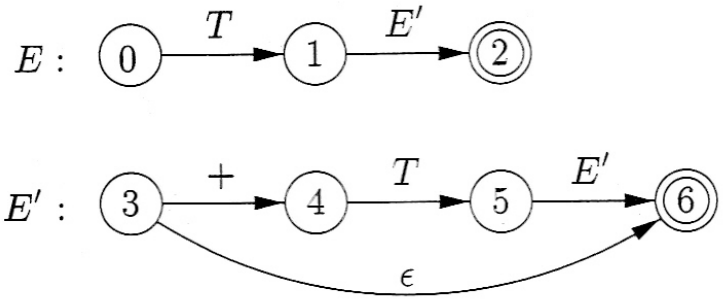
Analisador Preditivo

- Diagramas de transição
  - Úteis para a visualização de analisadores preditivos
  - Gramática deve ter a recursão a esquerda eliminada e também deve ser fatorada a esquerda
  - Uma transição em um token significa que deve-se seguir tal alternativa caso o token seja o próximo símbolo da entrada
  - Uma transição em um não terminal significa que este deve ser expandido

Analisador Preditivo

- Diagramas de transição
  - Etapas para a construção
  - Para cada não terminal *A*
    - Criar um estado inicial e um estado final
    - Para cada produção $A \rightarrow X_1X_2 \ldots X_k$, criar um caminho do estado inicial ao estado final com arcos X_1, X_2,… X_k. Se $A \rightarrow \varepsilon$, inserir também um arco rotulado por ε

$$E \rightarrow TE'$$
$$E' \rightarrow +TE' \mid \varepsilon$$
$$T \rightarrow FT'$$
$$T' \rightarrow *FT' \mid \varepsilon$$
$$F \rightarrow (E) \mid id$$



Análise Top Down

- Exercício
  - Considere a seguinte gramática:

    E → EE+

    E → EE*

    E → num

    1. Eliminar a recursão a esquerda
    2. Fatorar a esquerda