ANÁLISE SINTÁTICA DESCENDENTE NÃO RECURSIVA

A análise sintática descendente não recursiva, diferentemente da recursiva, trabalha com uma pilha explícita. Os movimentos desta pilha são ditados por um algoritmo conforme explicitado abaixo.

Inicialmente o analisador está em uma configuração onde existe $S, onde S é o símbolo inicial da gramática, e $\alpha$$ no buffer de entrada :

```
Aponte pra o primeiro símbolo de α$;
Repita
        Seja X o conteúdo do topo da pilha;
        Seja a o próximo símbolo de entrada;
        Se X é um terminal ou $
        Então se X=a
                Então desempilhar(X); avançar-entrada;
                Senão erro (esperado X);
        Senão se M[X,a]<> vazio
                Então desempilhar(X); empilhar(M[X,a])
                Senão erro()
Até X=$
```

Observação : M[X,a] é o resultado da consulta à tabela do analisador sintático descendente não recursivo.

Exemplo : Dada a gramática e a tabela da página 3:

Sigma = {+,*,(,),ide}
N = {E,T,F}
S = E
P = { E ==> TE'
E'==> +TE' | ε
T ==> FT'
T'==> *FT' | ε
F ==> (E) | ide | num}

Analise a seguinte cadeia mostrando a configuração da pilha em cada passagem e indique se existe ou não erro sintático :

ide * (ide+ide)$

CONSTRUÇÃO DA TABELA DO ANALISADOR SINTÁTICO DESCENDENTE

Para construir a tabela do analisador sintático descendente não recursivo usa-se duas funções : FIRST(α) e FOLLOW(A), onde α é uma cadeia qualquer e A é um não-terminal.

FIRST(α) é um conjunto de símbolos que iniciam as cadeias derivadas de α .

FOLLOW(A) é o conjunto de símbolos a que aparecem imediatamente a direita de A, nas derivações do tipo S ➔ αAaβ, onde αe β são cadeias quaisquer.

Definição de FIRST(α)

1- Se α é um terminal então FIRST(α)={α}
2- Se α é um não-terminal e α ➔aβ então adicionar a ao FIRST(α), onde a é um terminal. Se α ➔ε então adicionar ε ao FIRST(α).
3- Se α é um não terminal e α ➔ $Y_1Y_2... Y_{k...}Y_n$ adicionar os FIRST(Y_1)-{ε} aos FIRST(α).
   Se Y_1 ➔ ε então α ➔ $Y_2Y_3... Y_{k...}Y_n$, adicionar os FIRST(Y_2)-{ε} aos FIRST(α).
   ....
   Se Y_{k-1} ➔ ε então α ➔ $Y_{k...}Y_n$, adicionar os FIRST(Y_k)-{ε} aos FIRST(α).
   ....
   Se Y_{n-1} ➔ ε então α ➔ Y_n, adicionar os FIRST(Y_n)-{ε} aos FIRST(α).
   Obs : $Y_1Y_2... Y_{k...}Y_n$ são não-terminais.

Definição de FOLLOW(A)

O fim de uma sentença a ser analisada é indicada pelo símbolo $.

1- $ pertence a FOLLOW(A), onde A é o símbolo inicial da gramática.
2- Se existir a produção A➔αBβ então adicionar FIRST(β)-{ε} ao FOLLOW(B). OBS: FIRST($Y_1Y_2... Y_{k...}Y_n$)=FIRST(Y_1)-{ε} se Y_1 não derivar ε.
3- Se existir a produção A➔αB ou A➔αBβ, onde FIRST(β) contém ε (β ➔ε), então adicionar os FOLLOW(A) aos FOLLOW(B).

MONTAGEM DA TABELA M[A,a]

1. Para cada produção A$\rightarrow\alpha$ da gramática faça os passos 2 e 3.
2. Para cada terminal a que pertence a FIRST(α), adicione A$\rightarrow\alpha$ a M[A,a].
3. Se ε pertence a FIRST(α) adicione A$\rightarrow\alpha$ a M[A,b], para cada terminal b pertencente a FOLLOW(A). Se ε pertence a FIRST(α) e $ a FOLLOW(A), adicione A$\rightarrow\alpha$ a M[A,$].
4. Faça cada entrada indefinida de M ser um estado de erro.

Dada a gramática do exemplo anterior monte a tabela M[A,a]

Aplicando a segunda regra :

	Ide	Num	+	*	(	)	$
E	E→TE'	E→TE'			E→TE'		
E'			E'→+TE'				
T	T→FT'	T→FT'			T→FT'		
T'				T'→*FT'			
F	F→ide	F→num			F→(E)		

Aplicando a terceira regra :

	Ide	Num	+	*	(	)	$
E	E→TE'	E→TE'			E→TE'		
E'			E'→+TE'			E'→ε	E'→ε
T	T→FT'	T→FT'			T→FT'		
T'			T'→ε	T'→*FT'		T'→ε	T'→ε
F	F→ide	F→num			F→(E)		

Aplicando a quarta regra :

	Ide	Num	+	*	(	)	$
E	E→TE'	E→TE'	Erro	Erro	E→TE'	Erro	Erro
E'	Erro	Erro	E'→+TE'	Erro	Erro	E'→ε	E'→ε
T	T→FT'	T→FT'	Erro	Erro	T→FT'	Erro	Erro
T'	Erro	Erro	T'→ε	T'→*FT'	Erro	T'→ε	T'→ε
F	F→ide	F→num	Erro	Erro	F→(E)	Erro	Erro