

LISTA DE EXERCICIOS DE COMPILADORES

Gramática 1

$F \rightarrow T \text{ id } (P); F | T \text{ id } (); F | T \text{ id } (P); | T \text{ id } ();$
 $T \rightarrow \text{int} | \text{void} | \text{char}$

Gramática 2

$C \rightarrow \text{se } B \text{ então } S | \text{se } B \text{ então } S \text{ senão } S$
 $B \rightarrow (\text{id}) | (\text{id} == \text{id}) | (\text{id} == \text{num}) | \text{num}$
 $S \rightarrow \text{outra } S | \text{outra}$

Gramática 3

$\text{FUNCTION} \rightarrow \text{FUNCTION TYPE id } (\text{PARAM}); |$
 $\text{FUNCTION TYPE id } (); |$
 $\text{TYPE id } (\text{PARAM}); |$
 $\text{TYPE id } ();$
 $\text{TYPE} \rightarrow \text{int} | \text{void} | \text{char}$
 $\text{PARAM} \rightarrow \text{PARAM, TYPE id} |$
 TYPE id

Gramática 4

$\text{EXPRESSION} \rightarrow \text{EXPRESSION plus TERM} |$
 TERM
 $\text{TERM} \rightarrow \text{TERM times FACTOR} |$
 FACTOR
 $\text{FACTOR} \rightarrow (\text{EXPRESSION}) | \text{identifier} | \text{number}$

1. Para as gramáticas acima construa a tabela sintática do analisador descendente preditivo se recursão.
2. Construa uma derivação válida para cada uma das gramáticas acima.
3. Considere a tabela de tokens abaixo e construa o AF do analisador léxico para cada uma das gramáticas acima.

| Token | Padrão |
|------------|--|
| id | letra(letra U digito) ⁺ |
| times | * |
| plus | + |
| identifier | letra(letra U digito U _) [*] |
| number | (digito ⁺ U digito [*] .digito ⁺) |

4. Escreva pseudocódigos dos Procedimentos para implementar o Analisador preditivo com recursão para a gramática 2.

5. Explique:
 1. O que é uma derivação ? O que é uma derivação mais-à-esquerda e uma derivação mais-à-direita?
 2. O que é uma árvore de derivação ?
 3. Quando uma gramática é ambígua ?
 4. O que é Back-end e Front-end em compilação ?
 5. O funcionamento de um analisador sintático preditivo sem recursão.
 6. Como o analisador léxico diferencia palavras reservadas e identificadores ?

6. A tabela sintática abaixo é de um analisador sintático descendente preditivo sem recursão para a gramática 5. Exemplifique seu funcionamento com a seguinte cadeia de entrada.

algoritmo id (vetor [num] inteiro)

inicio

sentença

sentença

fim

Gramática 5

1. $A \rightarrow \text{algoritmo id (P) D A |}$
2. $A \rightarrow \epsilon$
3. $P \rightarrow \text{T id, P |}$
4. $P \rightarrow \epsilon$
5. $T \rightarrow \text{C B}$
6. $C \rightarrow \text{ponteiro |}$
7. $C \rightarrow \text{vetor [num] I |}$
8. $C \rightarrow \epsilon$
9. $I \rightarrow \text{[num] I |}$
10. $I \rightarrow \epsilon$
11. $B \rightarrow \text{numérico |}$
12. $B \rightarrow \text{caracter}$
13. $D \rightarrow \text{inicio S fim}$
14. $S \rightarrow \text{sentença R}$
15. $R \rightarrow \text{S |}$
16. $R \rightarrow \epsilon$

| | algoritmo | id | (|) | , | ponteiro | vetor | [num] | inteiro | caracter | inicio | fim | sentença | \$ |
|-----------|-----------|----|----------|---|---|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|----------|
| A | 1 | | | | | | | | | | | | | 2 |
| P | | | 4 | | | 3 | 3 | | 3 | 3 | | | | |
| T | | | | | | 5 | 5 | | 5 | 5 | | | | |
| C | | | | | | 6 | 7 | | 8 | 8 | | | | |
| I | | | | | | | | 9 | 10 | 10 | | | | |
| B | | | | | | | | | 11 | 12 | | | | |
| D | | | | | | | | | | | 13 | | | |
| S | | | | | | | | | | | | | 14 | |
| S' | | | | | | | | | | | | 16 | 15 | |