



SCC-206 INTRODUÇÃO À COMPILAÇÃO

Prof. Thiago A. S. Pardo

1ª. LISTA DE EXERCÍCIOS

- 1) O que é um compilador?
- 2) Qual a diferença entre um interpretador e um compilador?
- 3) Liste as principais fases de um compilador e descreva brevemente suas funcionalidades.
- 4) Qual a diferença entre as notações BNF e EBNF?
- 5) Liste as características que diferenciam as gramáticas da hierarquia de Chomsky. Qual o tipo de linguagem das linguagens de programação? Justifique e exemplifique.
- 6) Quais são as classes de tokens que um analisador léxico reconhece? Exemplifique.
- 7) Para que serve a tabela de palavras reservadas em um compilador?
- 8) Qual a função das rotinas de tratamento de erros em um compilador? Que tipos de erro podem ser detectados durante a análise léxica? Exemplifique.
- 9) Construa um autômato finito para o reconhecimento de números reais.
- 10) Implemente uma sub-rotina correspondente ao autômato anterior.
- 11) Sejam as gramáticas:

$G_1 = (V_n, V_t, P, S)$, em que P consiste das regras

$$\begin{aligned} S &\rightarrow A \mid B \mid \lambda \\ A &\rightarrow A+B \mid A-B \mid 1 \mid 2 \mid 3 \mid \lambda \\ B &\rightarrow A \mid C \\ C &\rightarrow (A) \end{aligned}$$

$G_2 = (V_n, V_t, P, S)$, em que P consiste das regras

$$\begin{aligned} S &\rightarrow \lambda \mid abA \mid abB \mid abC \\ A &\rightarrow aSaa \mid b \\ B &\rightarrow bSbb \mid c \\ C &\rightarrow cSc \mid d \end{aligned}$$

$G_3 = (V_n, V_t, P, S)$, em que P consiste das regras

$S \rightarrow bAb$
 $A \rightarrow CB \mid a$
 $B \rightarrow Aa$
 $C \rightarrow c \mid \lambda$

(a) Descreva as linguagens de G_1 , G_2 e G_3 , dando alguns exemplos de cadeias possíveis para cada linguagem.

(b) Calcule os conjuntos primeiro e seguidor para todo não terminal.

12) Simule a saída da análise léxica para o programa em LALG abaixo:

```
program nome1;  
{exemplo 1}  
var a, a1, b: integer;  
begin  
  read(a, b);  
  a1:= a + b;  
  while a1>a do  
  begin  
    write(a1);  
    a1:= a1-1;  
  end;  
  if a<> b then write(a);  
end.
```

13) As gramáticas abaixo são LL(1)? Transforme as que não são.

(a) $S \rightarrow ABc$
 $A \rightarrow a\lambda$
 $B \rightarrow b\lambda$

(b) $S \rightarrow Ab$
 $A \rightarrow aB\lambda$
 $B \rightarrow b\lambda$

(c) $S \rightarrow ABBA$
 $A \rightarrow a\lambda$
 $B \rightarrow b\lambda$

(d) $S \rightarrow aSe\lambda B$
 $B \rightarrow bBe\lambda C$
 $C \rightarrow cBe\lambda d$

14) Construa os grafos sintáticos para os trechos da gramática abaixo, em número reduzido quando adequado.

```
<corpo> ::= <dc> begin <comandos> end
<comandos> ::= <cmd> ; <comandos> | λ
<cmd> ::= read ( <variaveis> ) |
        write ( <variaveis> ) |
        while <condicao> do <cmd> |
        if <condicao> then <cmd> <pfalsa> |
        ident := <expressao> |
        ident <lista_arg> |
        begin <comandos> end
```

15) Para os grafos anteriores, produza os procedimentos recursivos da análise sintática descendente preditiva recursiva com o tratamento adequado de erros.

16) Qual a diferença entre a análise sintática descendente preditiva recursiva e a não recursiva? Por que são chamadas ‘preditivas’?

17) Considere a gramática abaixo:

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid id \end{aligned}$$

(a) Verifique se é LL(1). Se não for, transforme-a.

(b) Faça a análise sintática descendente preditiva não recursiva para a cadeia $id*id+id$

18) Altere a gramática da LALG para que ela reconheça o comando “case”.

19) Quais as vantagens da separação da análise léxica da sintática?

20) Explique brevemente o funcionamento da análise ‘empilha e reduz’.

21) Observe a gramática abaixo:

$$\begin{aligned} S &\rightarrow (L) \mid a \\ L &\rightarrow L, S \mid S \end{aligned}$$

- (i) que linguagem essa gramática gera?
- (ii) construa uma derivação mais à direita para $(a, (a, a))$ e mostre o *handle* de cada forma sentencial à direita
- (iii) mostre os passos de um analisador sintático empilha e reduz correspondentes à derivação mais à direita de (a)

22) A tabela abaixo mostra as relações de precedência de operadores para a gramática anterior.

	a	()	,	\$
a			>	>	>
(<	<	=	<	
)			>	>	>
,	<	<	>	>	
\$	<	<			

Usando essas relações, mostre os passos da análise sintática das sentenças abaixo:

(a, a)

(a, (a, a))

(a, ((a, a), (a,a)))