

PROF. RICARDO FERREIRA MARTINS  
LISTA 02

**Q. 01** – Dada a gramática:

```
E -> E + T | E - T | T
T -> T * F | F
F -> ( E ) | num
```

Escreva derivações e árvores sintáticas para as expressões a seguir (assuma que os numerais são instâncias do terminal *num*):

- $3 + 4 * 5 - 6$
- $3 * (4 - 5 + 6)$
- $3 - (4 + 5 * 6)$

**Q. 02** – Escreva uma gramática para expressões *booleanas* contendo as constantes **true** e **false** e os operadores **&&**, **||** e **!**, além de parênteses. O operador **||** tem precedência menor que **&&**, e este menor que **!**. A gramática não pode ser ambígua.

**Q. 03** – Analise se as regras abaixo possuem características de ambiguidade.

```
CMD -> if ( exp ) CMD | CASAM-CMD
CASAM-CMD -> if ( exp ) CASAM-CMD else CMD | outro
```

**Q. 04** – Na gramática a seguir os não-terminais estão em maiúsculas e os terminais em minúsculas, todos separados por espaços:

```
PROGRAMA -> begin LISTACMD end
LISTACMD -> LISTACMD CMD
          | CMD
CMD       -> do id := num to num begin LISTACMD end
          | read id
          | write EXP
          | id := EXP
EXP       -> EXP + EXP | EXP - EXP | num | id | ( EXP )
```

Verifique se essa gramática pode ser aplicada ao método LL(1).

**Q. 05** – Uma gramática LL(1) pode ser ambígua? Justifique.  
Uma gramática não ambígua precisa ser LL(1)? Justifique.

**Q. 06** – Dadas as duas gramáticas a seguir:

```
LEXP -> ATOMO | LISTA
ATOMO -> num | id
LISTA -> ( LEXP-SEQ )
LEXP-SEQ -> LEXP-SEQ LEXP | LEXP

DECL -> TIPO VAR-LISTA
TIPO -> int | float
VAR-LISTA -> id , VAR-LISTA | id
```

Remova a recursão esquerda, caso seja necessário.

**Q. 07** – Encontre o FIRST e o FOLLOW dos seguintes conjuntos gramaticais:

```
S' → S $
S → ε
S → X S
B → \ begin { WORD }
B → \ end { WORD }
X → B S E
X → { S }
X → WORD
X → begin
X → end
X → \ WORD
```

**Q. 08** – Considere a gramática:

```
expr → expr or termo | termo
termo → termo and fator | fator
fator → not fator | (expr) | true | false
```

- i) Construa uma árvore gramatical (*parsing*) para a seguinte sentença:  
not (true or false and true)
- ii) Esta gramática é ambígua? Justifique sua resposta.

**Q. 09** – Dada a gramática:

```
CMD -> ATRIB
CMD -> CHAMADA
CMD -> outro
ATRIB -> id := exp
CHAMADA -> id ( exp )
```

Essa gramática é LL(1)? Justifique.