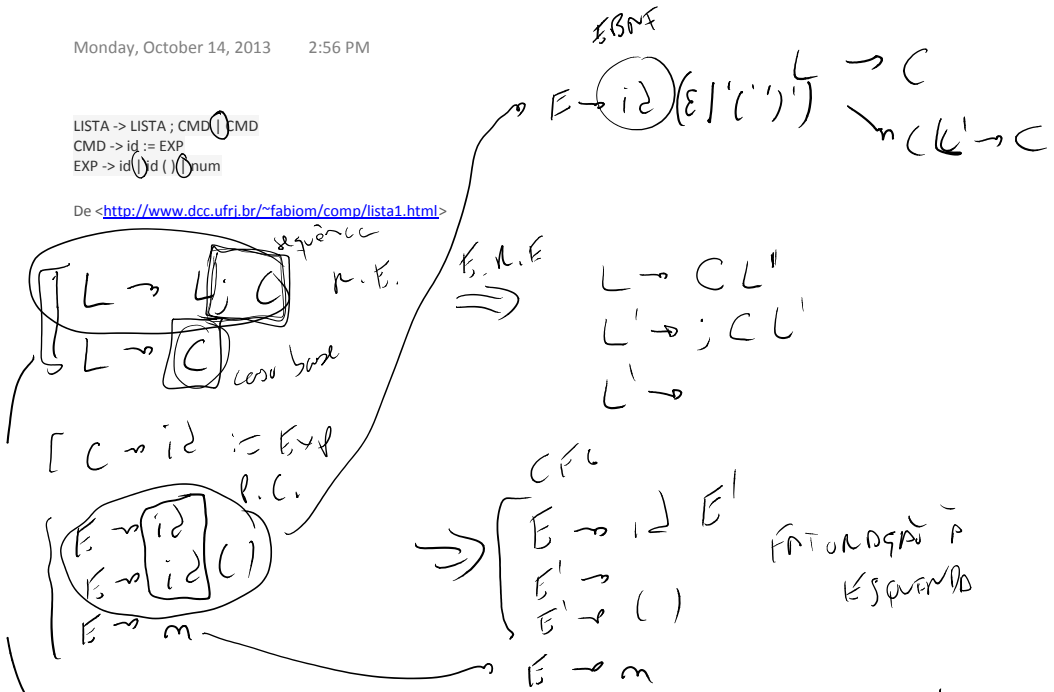
Monday, October 14, 2013 2:56 PM

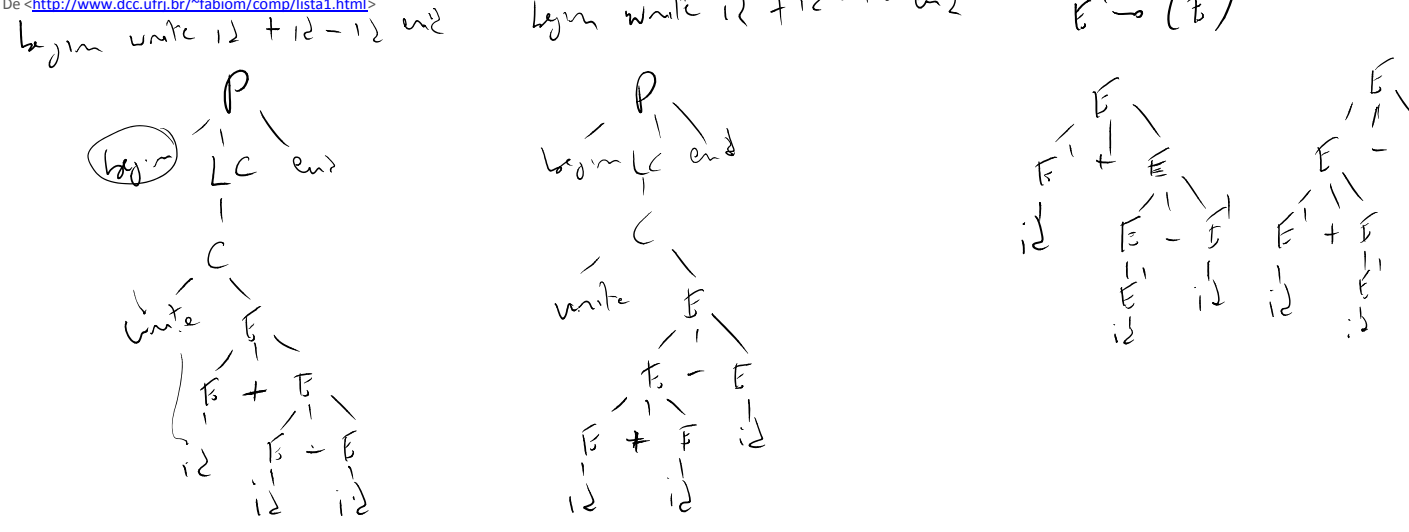
LISTA -> LISTA ; CMD | CMD
CMD -> id := EXP
EXP -> id | id ( ) | num

De <http://www.dcc.ufrj.br/~fabiom/comp/lista1.html>



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 )

De <http://www.dcc.ufrj.br/~fabiom/comp/lista1.html>



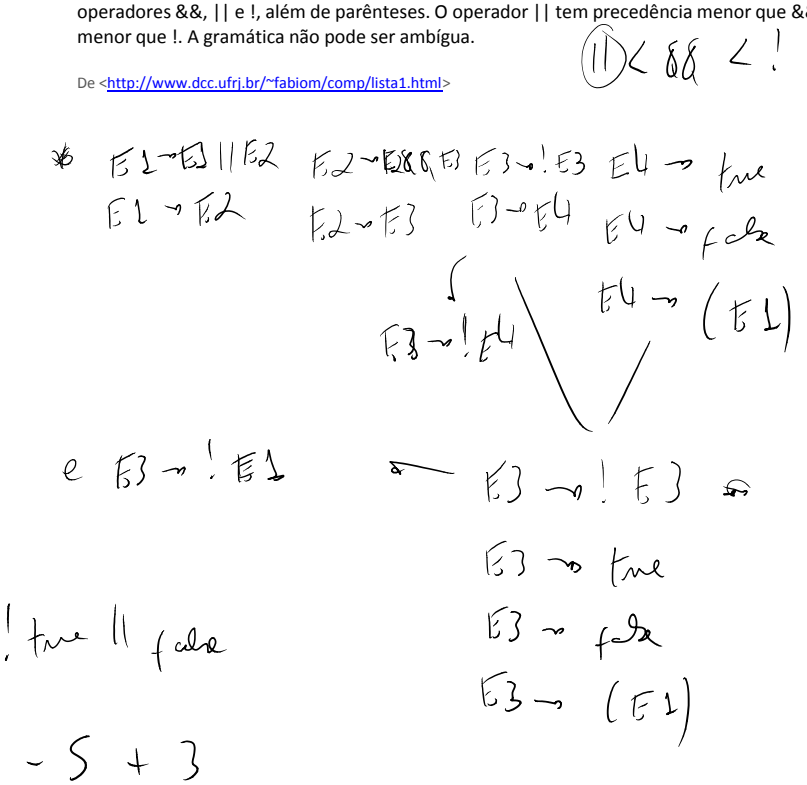
9.

A -> A A | ( A ) | *vazio*

De <http://www.dcc.ufrj.br/~fabiom/comp/lista1.html>

11. Escreva uma gramática para expressões boolenas 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.

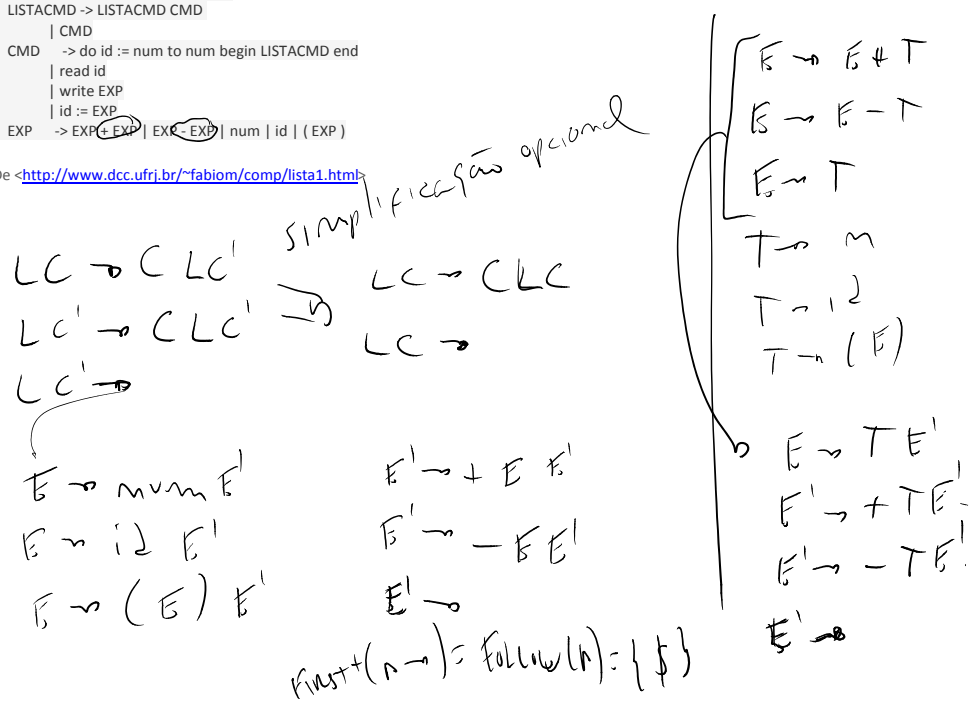
De <http://www.dcc.ufrj.br/~fabiom/comp/lista1.html>



14.

```
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 )
```

De <http://www.dcc.ufrj.br/~fabiom/comp/lista1.html>



16. Dada a gramática A -> ( A ) A | *vazio*, escreva pseudocódigo para analisá-la de forma recursiva com retrocesso local e sem retrocesso (LL(1)).

De <http://www.dcc.ufrj.br/~fabiom/comp/lista1.html>

```
Arvore res = new Arvore();
int atual = pos;
try {
    Arvore n = new Arvore();
    n.filho(match('('));
    n.filho(A());
    n.filho(match(')'));
    n.filho(A());
    res.addAll(n.filhos);
} catch (Falha f) {
    pos = atual;
}
return res;
}

Arvore A() { // sem retrocesso
    Arvore res = new Arvore();
    if (la == '(') {
        res.filho(match('('));
        res.filho(A());
        res.filho(match(')'));
        res.filho(A());
    } // else { }
    return res;
}
```