

$AB$        $B$   
 $\neg AB$        $\neg$   
 $\neg B$

① Lexema: conjunto de caracteres do programa fonte.

Token: é uma unidade Lética que contém um nome e valor.

Padrão: forma que os lexemas podem assumir.

②

```
Token nextToken()
{
```

```
    string lexeme = "";
    while( (isLetter(input.charAt(pos)) || isDigit(input.charAt(pos))) &&
           !(lexeme.isEmpty() && isDigit(input.charAt(pos))))
        lexeme += input.charAt(pos++);
    if(!lexeme.isEmpty())
        return Token(ID, lexeme);
```

CONTINUA

```
}
```

L2:

→ Não pode ser Ambígua.

$S \rightarrow A B i$   
 $\quad | A B j$       X  
 $\quad | \dots$

→ Não pode recursiva à esquerda.

$S \rightarrow S A i$       X  
 $\quad | \dots$

→ Não pode:  $FIRST(S) = FOLLOW(S)$

③ a) Não, pois tem recursão à esq. em 1, 3.

1.  $Expr \rightarrow Term Expr'$

2.  $Expr' \rightarrow \epsilon | Expr'$

3.       $| \epsilon$

4.  $Term \rightarrow Factor Term'$

5.  $Term' \rightarrow \&\& Term'$

6.       $| \epsilon$

7.  $Factor \rightarrow ! Factor$

8.       $| (Expr)$

9.       $| true$

10.       $| false$

b)

	first	follow	first +
Expr	{!, (, true, false}	{\$, )}	_____
Expr'	{ , ε}	{\$, )}	{ε, \$,  }
Term	{!, (, true, false}	{ , \$, )}	_____
Term'	{& &, ε}	{ , \$, )}	{ε,  , \$, )}
factor	{!, (, true, false}	{& &,  , \$, )}	_____