

# Analizador Sintático

João Pedro Marin Comini RA: 161020551

Pedro Vitor Haramoto RA: 161024157



# Analizador Sintático

O analisador léxico é a segunda fase de um compilador e sua tarefa é analisar uma estrutura de entrada (tokens) para determinar sua estrutura gramatical segundo uma determinada gramática formal. Ou seja, é o processo de determinar se uma cadeia de símbolos pode ser gerada por uma gramática.

# A gramática

A gramática é dita LL(1) se toda predição pode ser feita analisando um único token à frente.

# A análise

Conceitualmente, o analisador LL(1) constrói uma derivação mais à esquerda para o programa, partindo do símbolo inicial

A cada passo da derivação, o prefixo de terminais da forma sentencial tem que casar com um prefixo da entrada

Caso exista mais de uma regra para o não-terminal que vai gerar o próximo passo da derivação, o analisador usa o primeiro token após esse prefixo para escolher qual regra usar

Esse processo continua até todo o programa ser derivado ou acontecer um erro (o prefixo de terminais da forma sentencial não casa com um prefixo do programa)

# A análise

PROG -> CMD ; PROG

PROG ->

CMD -> id = EXP

CMD -> print EXP

EXP -> id

EXP -> num

EXP -> ( EXP + EXP )

id = ( num + id ) ;

print num ;

# Exemplo

```
1  {comentarios}  
2  program ex1;  
3  var  
4  a, b, c, i : integer;  
5  a, b, c, i : integer;  
6  d : integer;  
7  e : array[1..4,1..4] of integer;  
8  begin  
9  
10     a := (4*(7+8))+88;  
11     b := 9*a;  
12     c := a;
```

```
[['program', 'PALAVRA_RESERVADA', 2], ['ex1', 'ID', 2], [';', 'ERR_SIMB_PONT', 2]]  
['var']  
['a', ',', 'b', ',', 'c', ',', 'i', ':', 'integer', ';']  
['a', ',', 'b', ',', 'c', ',', 'i', ':', 'integer', ';']  
['d', ':', 'real', ';']  
['e', ':', 'array', '[', '1', '..', '4', '1', '..', '4', 'of', 'integer', ';']  
['begin']  
['a', ':=', '(', '4', '*', '(', '7', '+', '8', ')', ')', '+', '88', ';']  
['b', ':=', '9', '*', 'a', ';']  
['c', ':=', 'a', ';']
```