

## Linguagem LALG (pascal simplificado)

1. <programa> ::= program ident ; <corpo> .
2. <corpo> ::= <dc> begin <comandos> end
3. <dc> ::= <dc\_c> <dc\_v> <dc\_p>
4. <dc\_c> ::= const ident = <numero> ; <dc\_c> |  $\lambda$
5. <dc\_v> ::= var <variaveis> : <tipo\_var> ; <dc\_v> |  $\lambda$
6. <tipo\_var> ::= real | integer
7. <variaveis> ::= ident <mais\_var>
8. <mais\_var> ::= , <variaveis> |  $\lambda$
9. <dc\_p> ::= procedure ident <parametros> ; <corpo\_p> <dc\_p> |  $\lambda$
10. <parametros> ::= ( <lista\_par> ) |  $\lambda$
11. <lista\_par> ::= <variaveis> : <tipo\_var> <mais\_par>
12. <mais\_par> ::= ; <lista\_par> |  $\lambda$
13. <corpo\_p> ::= <dc\_loc> begin <comandos> end ;
14. <dc\_loc> ::= <dc\_v>
15. <lista\_arg> ::= ( <argumentos> ) |  $\lambda$
16. <argumentos> ::= ident <mais\_ident>
17. <mais\_ident> ::= ; <argumentos> |  $\lambda$
18. <pfalsa> ::= else <cmd> |  $\lambda$
19. <comandos> ::= <cmd> ; <comandos> |  $\lambda$
20. <cmd> ::= read ( <variaveis> ) |  
          write ( <variaveis> ) |  
          while ( <condicao> ) do <cmd> |  
          if <condicao> then <cmd> <pfalsa> |  
          ident := <expressão> |  
          ident <lista\_arg> |  
          begin <comandos> end
21. <condicao> ::= <expressao> <relacao> <expressao>
22. <relacao> ::= = | < > | >= | <= | > | <
23. <expressao> ::= <termo> <outros\_termos>
24. <op\_un> ::= + | - |  $\lambda$
25. <outros\_termos> ::= <op\_ad> <termo> <outros\_termos> |  $\lambda$
26. <op\_ad> ::= + | -
27. <termo> ::= <op\_un> <fator> <mais\_fatores>
28. <mais\_fatores> ::= <op\_mul> <fator> <mais\_fatores> |  $\lambda$
29. <op\_mul> ::= \* | /
30. <fator> ::= ident | <numero> | ( <expressao> )
31. <numero> ::= numero\_int | numero\_real

Além disso, inclusão do comando **for**.

Comentários de única linha, entre chaves { }

Identificadores e números são itens léxicos da forma:

- ident: sequência de letras e dígitos, começando por letra

- número inteiro: sequência de dígitos (0 a 9)

- número real: pelo menos um dígito, seguido de um ponto decimal, seguido de uma sequência de um ou mais dígitos

### Exemplos de programas LALG

```
program nome1;
{exemplo 1}
var a, al, b: integer;
begin
  read(a);
  al:= a*2;
  while (al>0) do
  begin
    write(al);
    al:= al-1;
  end;
  for b:=1 to 10 do
  begin
    b:=b+2;
    a:=a-1;
  end;
  if a<> b then write(a);
end.
```

```
program nome2;
{exemplo 2}
var a: real;
var b: integer;
procedure nomep(x: real);
var a, c: integer;
begin
  read(c, a);
  if a<x+c then
  begin
    a:= c+x;
    write(a);
  end
  else c:= a+x;
end;
begin {programa principal}
  read(b);
  nomep(b);
end.
```