

## Linguagem LALG

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

Comentários entre chaves { }

Identificadores e números são itens léxicos da forma:

- ident: seqüência de letras e dígitos, começando por letra
- número inteiro: seqüência de dígitos (0 a 9)
- número real: pelo menos um dígito seguido de um ponto decimal e de uma seqüência de um ou mais dígitos

### Exemplos de programas LALG

```
program nome1;  
{exemplo 1}  
var a, a1, b: integer;  
begin  
  read(a, b);  
  a1:= a + b;  
  while a1>a do  
  begin  
    write(a1);  
    a1:= a1-1;  
  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.
```