Disciplina: Compiladores — 6ª fase

Professora: Christine Vieira

GRAMÁTICA

1. PROGRAMA ::= begin (ident) ; BLOCO end
2. BLOCO ::= DCLVAR DCLPROC CORPO
3. DCLVAR ::= var LID : TIPO ; LDVAR
4. DCLVAR ::= î
5. LID ::= ident REPIDENT
6. REPIDENT ::= , ident REPIDENT
7. REPIDENT ::= î
8. LDVAR ::= LID : TIPO ; LDVAR
9. LDVAR ::= î
10. TIPO ::= integer
11. TIPO ::= string
12. TIPO ::= float
13. DCLPROC ::= procedure ident DEFPAR { BLOCO } DCLPROC
14. DCLPROC ::= î
15. DEFPAR ::= (ident : TIPO REPDEFPAR)
16. DEFPAR ::= î
17. REPDEFPAR ::= ; ident : TIPO REPDEFPAR
18. REPDEFPAR ::= î
19. CORPO ::= begin COMANDO REPCOMANDO fim
20. COMANDO ::= if EXPRESSAO then CORPO ELSEPARTE
21. COMANDO ::= while EXPRESSAO do CORPO
22. COMANDO ::= do CORPO while EXPRESSAO
23. COMANDO ::= read (ident)
24. COMANDO ::= call ident PARAMETROS
25. COMANDO ::= write (ITEMSAIDA REPITEM)
26. COMANDO ::= case EXPRESSAO of CONDCASE fim
27. COMANDO ::= ident := EXPRESSAO ;
28. COMANDO ::= î
29. REPCOMANDO ::= ; COMANDO REPCOMANDO
30. REPCOMANDO ::= î
31. PARAMETROS ::= (EXPRESSAO REPPAR)
32. PARAMETROS ::= î
33. REPPAR ::= , EXPRESSAO REPPAR
34. REPPAR ::= î
35. ELSEPARTE ::= senao COMANDO
36. ELSEPARTE ::= î
37. VARIAVEL ::= ident
38. REPVARIAVEL ::= , VARIAVEL REPVARIAVEL
39. REPVARIAVEL ::= î

40. ITEMSAIDA ::= literal
41. ITEMSAIDA ::= EXPRESSAO
42. REPITEM ::= , ITEMSAIDA REPITEM
43. REPITEM ::= î
44. CONDCASE ::= COND : CORPO CONTCASE
45. COND ::= ident
46. COND ::= ninteiro
47. COND::= nfloat
48. CONTCASE ::= ; CONDCASE
49. CONTCASE ::= î
50. EXPRESSAO ::= EXPSIMP REPEXPSIMP
51. EXPSIMP ::= + TERMO REPEXP
52. EXPSIMP ::= - TERMO REPEXP
53. EXPSIMP ::= TERMO REPEXP
54. REPEXPSIMP ::= = EXPSIMP
55. REPEXPSIMP ::= < EXPSIMP
56. REPEXPSIMP ::= > EXPSIMP
57. REPEXPSIMP ::= >= EXPSIMP
58. REPEXPSIMP ::= <= EXPSIMP
59. REPEXPSIMP ::= <> EXPSIMP
60. REPEXPSIMP ::= î
61. TERMO ::= FATOR REPTERMO
62. REPEXP ::= + TERMO REPEXP
63. REPEXP ::= - TERMO REPEXP
64. REPEXP ::= or TERMO REPEXP
65. REPEXP ::= î
66. REPTERMO ::= * FATOR REPTERMO
67. REPTERMO ::= / FATOR REPTERMO
68. REPTERMO ::= and FATOR REPTERMO
69. REPTERMO ::= î
70. FATOR ::= ninteiro
71. FATOR ::= string
72. FATOR ::= nfloat
73. FATOR ::= (EXPRESSAO)
74. FATOR ::= not FATOR
75. FATOR ::= VARIAVEL

TERMINAIS CODIFICADOS

1. write
2. while
3. var
4. then
5. string
6. senao
7. read
8. procedure
9. or
10. of
11. not
12. ninteiro
13. nfloat
14. literal
15. integer
16. if
17. ident
18. î
19. float
20. fim
21. end
22. do
23. case
24. call
25. begin
26. and
27. >=
28. >
29. =
30. <>
31. <=
32. <
33. +
34. }
35. {
36. ;
37. :=
38. :
39. /
40. ,
41. *
42.)
43. (
44. $
45. -

NÃO TERMINAIS CODIFICADOS

46. <PROGRAMA>
47. <BLOCO>
48. <DCLVAR>
49. <DCLPROC>
50. <CORPO>
51. <LID>
52. <TIPO>
53. <LDVAR>
54. <REPIDENT>
55. <DEFPAR>
56. <REPDEFPAR>
57. <COMANDO>
58. <REPCOMANDO>
59. <EXPRESSAO>
60. <ELSEPARTE>
61. <PARAMETROS>
62. <ITEMSAIDA>
63. <REPITEM>
64. <CONDCASE>
65. <REPPAR>
66. <VARIAVEL>
67. <REPVARIAVEL>
68. <COND>
69. <CONTCASE>
70. <EXPSIMP>
71. <REPEXPSIMP>
72. <TERMO>
73. <REPEXP>
74. <FATOR>
75. <REPTERMO>

TABELA DE PARSING

```
tabParsing[46][25] = 1;
tabParsing[47][3] = 2;
tabParsing[47][8] = 2;
tabParsing[47][25] = 2;
tabParsing[48][3] = 3;
tabParsing[48][8] = 4;
tabParsing[48][25] = 4;
tabParsing[49][8] = 13;
tabParsing[49][25] = 14;
tabParsing[50][25] = 19;
tabParsing[51][17] = 5;
tabParsing[52][5] = 11;
tabParsing[52][15] = 10;
tabParsing[52][19] = 12;
tabParsing[53][8] = 9;
tabParsing[53][17] = 8;
tabParsing[53][21] = 9;
tabParsing[53][34] = 9;
tabParsing[54][38] = 7;
tabParsing[54][40] = 6;
tabParsing[55][35] = 16;
tabParsing[55][43] = 15;
tabParsing[56][36] = 17;
tabParsing[56][42] = 18;
tabParsing[57][1] = 25;
tabParsing[57][2] = 28;
tabParsing[57][6] = 28;
tabParsing[57][7] = 23;
tabParsing[57][16] = 20;
tabParsing[57][17] = 27;
tabParsing[57][20] = 28;
tabParsing[57][21] = 28;
tabParsing[57][22] = 22;
tabParsing[57][23] = 26;
tabParsing[57][24] = 24;
tabParsing[57][34] = 28;
tabParsing[57][36] = 28;
tabParsing[58][20] = 30;
tabParsing[58][36] = 29;
tabParsing[59][5] = 50;
tabParsing[59][11] = 50;
tabParsing[59][12] = 50;
tabParsing[59][13] = 50;
tabParsing[59][17] = 50;
tabParsing[59][33] = 50;
tabParsing[59][43] = 50;
tabParsing[59][45] = 50;
tabParsing[60][2] = 36;
tabParsing[60][6] = 36;
tabParsing[60][20] = 36;
tabParsing[60][21] = 36;
tabParsing[60][34] = 36;
```

```
tabParsing[60][36] = 36;
tabParsing[61][2] = 32;
tabParsing[61][6] = 32;
tabParsing[61][20] = 32;
tabParsing[61][21] = 32;
tabParsing[61][34] = 32;
tabParsing[61][36] = 32;
tabParsing[61][43] = 31;
tabParsing[62][5] = 41;
tabParsing[62][11] = 41;
tabParsing[62][12] = 41;
tabParsing[62][13] = 41;
tabParsing[62][14] = 40;
tabParsing[62][17] = 41;
tabParsing[62][33] = 41;
tabParsing[62][43] = 41;
tabParsing[62][45] = 41;
tabParsing[63][40] = 42;
tabParsing[63][42] = 43;
tabParsing[64][12] = 44;
tabParsing[64][13] = 44;
tabParsing[64][17] = 44;
tabParsing[65][40] = 33;
tabParsing[65][42] = 34;
tabParsing[66][17] = 37;
tabParsing[67][40] = 38;
tabParsing[68][12] = 46;
tabParsing[68][13] = 47;
tabParsing[68][17] = 45;
tabParsing[69][20] = 49;
tabParsing[69][36] = 48;
tabParsing[70][5] = 53;
tabParsing[70][11] = 53;
tabParsing[70][12] = 53;
tabParsing[70][13] = 53;
tabParsing[70][17] = 53;
tabParsing[70][33] = 51;
tabParsing[70][43] = 53;
tabParsing[70][45] = 52;
tabParsing[71][2] = 60;
tabParsing[71][4] = 60;
tabParsing[71][6] = 60;
tabParsing[71][10] = 60;
tabParsing[71][20] = 60;
tabParsing[71][21] = 60;
tabParsing[71][22] = 60;
tabParsing[71][27] = 57;
tabParsing[71][28] = 56;
tabParsing[71][29] = 54;
tabParsing[71][30] = 59;
tabParsing[71][31] = 58;
tabParsing[71][32] = 55;
tabParsing[71][34] = 60;
tabParsing[71][36] = 60;
```

```
tabParsing[71][40] = 60;
tabParsing[71][42] = 60;
tabParsing[72][5] = 61;
tabParsing[72][11] = 61;
tabParsing[72][12] = 61;
tabParsing[72][13] = 61;
tabParsing[72][17] = 61;
tabParsing[72][43] = 61;
tabParsing[73][2] = 65;
tabParsing[73][4] = 65;
tabParsing[73][6] = 65;
tabParsing[73][9] = 64;
tabParsing[73][10] = 65;
tabParsing[73][20] = 65;
tabParsing[73][21] = 65;
tabParsing[73][22] = 65;
tabParsing[73][27] = 65;
tabParsing[73][28] = 65;
tabParsing[73][29] = 65;
tabParsing[73][30] = 65;
tabParsing[73][31] = 65;
tabParsing[73][32] = 65;
tabParsing[73][33] = 62;
tabParsing[73][34] = 65;
tabParsing[73][36] = 65;
tabParsing[73][40] = 65;
tabParsing[73][42] = 65;
tabParsing[73][45] = 63;
tabParsing[74][5] = 71;
tabParsing[74][11] = 74;
tabParsing[74][12] = 70;
tabParsing[74][13] = 72;
tabParsing[74][17] = 75;
tabParsing[74][43] = 73;
tabParsing[75][2] = 69;
tabParsing[75][4] = 69;
tabParsing[75][6] = 69;
tabParsing[75][9] = 69;
tabParsing[75][10] = 69;
tabParsing[75][20] = 69;
tabParsing[75][21] = 69;
tabParsing[75][22] = 69;
tabParsing[75][26] = 68;
tabParsing[75][27] = 69;
tabParsing[75][28] = 69;
tabParsing[75][29] = 69;
tabParsing[75][30] = 69;
tabParsing[75][31] = 69;
tabParsing[75][32] = 69;
tabParsing[75][33] = 69;
tabParsing[75][34] = 69;
tabParsing[75][36] = 69;
tabParsing[75][39] = 67;
tabParsing[75][40] = 69;
```

```
tabParsing[75][41] = 66;
tabParsing[75][42] = 69;
tabParsing[75][45] = 69;
```

PRODUÇÕES CODIFICADAS

1- 25 43 17 42 36 47 21
2- 48 49 50
3- 3 51 38 52 36 53
4- 18
5- 17 54
6- 40 17 54
7- 18
8- 51 38 52 36 53
9- 18
10- 15
11- 5
12- 19
13- 8 17 55 35 47 34 49
14- 18
15- 43 17 38 52 56 42
16- 18
17- 36 17 38 52 56
18- 18
19- 25 57 58 20
20- 16 59 4 50 60
21- 2 59 22 50
22- 22 50 2 59
23- 7 43 17 42
24- 24 17 61
25- 1 43 62 63 42
26- 23 59 10 64 20
27- 17 37 59 36
28- 18
29- 36 57 58
30- 18
31- 43 59 65 42
32- 18
33- 40 59 65
34- 18
35- 6 57
36- 18
37- 17
38- 40 66 67
39- 18
40- 14
41- 59
42- 40 62 63
43- 18
44- 68 38 50 69
45- 17
46- 12
47- 13
48- 36 64
49- 18
50- 70 71
51- 33 72 73
52- 45 72 73

53- 72 73
54- 29 70
55- 32 70
56- 28 70
57- 27 70
58- 31 70
59- 30 70
60- 18
61- 74 75
62- 33 72 73
63- 45 72 73
64- 9 72 73
65- 18
66- 41 74 75
67- 39 74 75
68- 26 74 75
69- 18
70- 12
71- 5
72- 13
73- 43 59 42
74- 11 74
75- 66