P R O D U C T I O N S

1. <program> ::= PROGRAM IDENTIFIER (IDENTIFIER <identifier_list>) ; <block> .
2. <block> ::= <label_declaration_part> <constant_definition_part> <type_definition_part> <variable_declaration_part> <proc_and_func_declaration_part> <statement_part>
3. <label_declaration_part> ::= LABEL UNSIGNEDINTEGER <label_declaration_part_remaind>
4. <label_declaration_part> ::= EPSILON
5. <label_declaration_part_remaind> ::= , UNSIGNEDINTEGER <label_declaration_part_remaind>
6. <label_declaration_part_remaind> ::= ;
7. <constant_definition_part> ::= CONST IDENTIFIER = <constant> ; <constant_definition_part_remai>
8. <constant_definition_part> ::= EPSILON
9. <constant_definition_part_remai> ::= IDENTIFIER = <constant> ; <constant_definition_part_remai>
10. <constant_definition_part_remai> ::= EPSILON
11. <constant> ::= <non_identifier_constant>
12. <constant> ::= IDENTIFIER
13. <non_identifier_constant> ::= PLUSMINUS <non_identifier_constant_remain>
14. <non_identifier_constant> ::= UNSIGNEDINTEGER
15. <non_identifier_constant> ::= UNSIGNEDREAL
16. <non_identifier_constant> ::= STRING
17. <non_identifier_constant_remain> ::= IDENTIFIER
18. <non_identifier_constant_remain> ::= UNSIGNEDINTEGER
19. <non_identifier_constant_remain> ::= UNSIGNEDREAL
20. <type_definition_part> ::= TYPE IDENTIFIER = <type> ; <type_definition_part_remainder>
21. <type_definition_part> ::= EPSILON
22. <type_definition_part_remainder> ::= IDENTIFIER = <type> ; <type_definition_part_remainder>
23. <type_definition_part_remainder> ::= EPSILON
24. <type> ::= <simple_type>
25. <type> ::= PACKED <unpacked_structure_type>
26. <type> ::= <unpacked_structure_type>
27. <type> ::= POINTER IDENTIFIER
28. <simple_type> ::= (IDENTIFIER <identifier_list>)

29. <simple_type> ::= <non_identifier_constant> .. <constant>
30. <simple_type> ::= IDENTIFIER <simple_type_remainder>
31. <simple_type_remainder> ::= .. <constant>
32. <simple_type_remainder> ::= EPSILON
33. <unpacked_structure_type> ::= ARRAY [<simple_type> <simple_type_list>] OF <type>
34. <unpacked_structure_type> ::= RECORD <field_list> END
35. <unpacked_structure_type> ::= FILE OF <type>
36. <unpacked_structure_type> ::= SET OF <simple_type>
37. <field_list> ::= <record_section> <field_list_remainder>
38. <field_list> ::= CASE IDENTIFIER <tag_field_remainder> OF <variant> <variant_list>
39. <record_section> ::= IDENTIFIER <identifier_list> : <type>
40. <record_section> ::= EPSILON
41. <field_list_remainder> ::= ; <field_list>
42. <field_list_remainder> ::= EPSILON
43. <tag_field_remainder> ::= : IDENTIFIER
44. <tag_field_remainder> ::= EPSILON
45. <variant> ::= <constant> <constant_list> : (<field_list>)
46. <variant> ::= EPSILON
47. <variant_list> ::= ; <variant> <variant_list>
48. <variant_list> ::= EPSILON
49. <variable_declaration_part> ::= VAR IDENTIFIER <identifier_list> : <type> ; <variable_declaration_part_rema>
50. <variable_declaration_part> ::= EPSILON
51. <variable_declaration_part_rema> ::= IDENTIFIER <identifier_list> : <type> ; <variable_declaration_part_rema>
52. <variable_declaration_part_rema> ::= EPSILON
53. <proc_and_func_declaration_part> ::= <proc_and_func_declaration> ; <proc_and_func_declaration_part>
54. <proc_and_func_declaration_part> ::= EPSILON
55. <proc_and_func_declaration> ::= PROCEDURE IDENTIFIER <formal_parameter_part> ; <block>
56. <proc_and_func_declaration> ::= FUNCTION IDENTIFIER <formal_parameter_part> : IDENTIFIER ; <block>
57. <formal_parameter_part> ::= (<formal_parameter> <formal_parameter_list>)
58. <formal_parameter_part> ::= EPSILON

59. <formal_parameter> ::= IDENTIFIER <identifier_list> : IDENTIFIER
60. <formal_parameter> ::= VAR IDENTIFIER <identifier_list> : IDENTIFIER
61. <formal_parameter> ::= FUNCTION IDENTIFIER <identifier_list> : IDENTIFIER
62. <formal_parameter> ::= PROCEDURE IDENTIFIER <identifier_list>
63. <statement_part> ::= BEGIN <statement> <compound_statement_remainder>
64. <compound_statement_remainder> ::= ; <statement> <compound_statement_remainder>
65. <compound_statement_remainder> ::= END
66. <statement> ::= UNSIGNEDINTEGER : <unlabeled_statement>
67. <statement> ::= <unlabeled_statement>
68. <unlabeled_statement> ::= IDENTIFIER <simple_statement_remainder>
69. <unlabeled_statement> ::= BEGIN <statement> <compound_statement_remainder>
70. <unlabeled_statement> ::= IF <expression> THEN <statement> <else_clause>
71. <unlabeled_statement> ::= CASE <expression> OF <case_element> <case_element_list>
72. <unlabeled_statement> ::= WHILE <expression> DO <statement>
73. <unlabeled_statement> ::= REPEAT <statement> <repeat_statement_list> <expression>
74. <unlabeled_statement> ::= FOR IDENTIFIER := <expression> <for_statement_remainder>
75. <unlabeled_statement> ::= WITH IDENTIFIER <variable_selectors> <with_variable_list> <statement>
76. <unlabeled_statement> ::= GOTO UNSIGNEDINTEGER
77. <unlabeled_statement> ::= EPSILON
78. <simple_statement_remainder> ::= <variable_selectors> := <expression>
79. <simple_statement_remainder> ::= <actual_parameter_part>
80. <simple_statement_remainder> ::= EPSILON
81. <variable_selectors> ::= [<expression> <expression_list>] <variable_selectors>
82. <variable_selectors> ::= . IDENTIFIER <variable_selectors>
83. <variable_selectors> ::= POINTER <variable_selectors>
84. <variable_selectors> ::= EPSILON
85. <actual_parameter_part> ::= (<expression> <expression_list>)
86. <else_clause> ::= ELSE <statement>
87. <case_element> ::= <constant> <constant_list> : <statement>
88. <case_element> ::= EPSILON

89. <case_element_list> ::= ; <case_element> <case_element_list>
90. <case_element_list> ::= END
91. <repeat_statement_list> ::= ; <statement> <repeat_statement_list>
92. <repeat_statement_list> ::= UNTIL
93. <for_statement_remainder> ::= TO <expression> DO <statement>
94. <for_statement_remainder> ::= DOWNTO <expression> DO <statement>
95. <with_variable_list> ::= , IDENTIFIER <variable_selectors> <with_variable_list>
96. <with_variable_list> ::= DO
97. <expression_list> ::= , <expression>
98. <expression_list> ::= EPSILON
99. <expression> ::= <simple_expression> <simple_expression_remainder>
100. <simple_expression_remainder> ::= = <simple_expression>
101. <simple_expression_remainder> ::= RELATIONOPERATOR <simple_expression>
102. <simple_expression_remainder> ::= EPSILON
103. <simple_expression> ::= PLUSMINUS <term>
104. <simple_expression> ::= <term> <term_list>
105. <term_list> ::= PLUSMINUS <term> <term_list>
106. <term_list> ::= OR <term> <term_list>
107. <term_list> ::= EPSILON
108. <term> ::= <factor> <factor_list>
109. <factor_list> ::= MULTIPLYINGOPERATOR <factor> <factor_list>
110. <factor_list> ::= EPSILON
111. <factor> ::= NOT <factor>
112. <factor> ::= IDENTIFIER <identifier_remainder>
113. <factor> ::= STRING
114. <factor> ::= UNSIGNEDINTEGER
115. <factor> ::= UNSIGNEDREAL
116. <factor> ::= NIL
117. <factor> ::= [<set_range>]
118. <factor> ::= (<expression>)

119. <identifier_remainder> ::= <variable_selectors>
120. <identifier_remainder> ::= <actual_parameter_part>
121. <set_range> ::= <set_element> <set_element_list>
122. <set_range> ::= EPSILON
123. <set_element> ::= <expression> <set_element_remainder>
124. <set_element_remainder> ::= .. <expression>
125. <set_element_remainder> ::= EPSILON
126. <set_element_list> ::= , <set_element> <set_element_list>
127. <set_element_list> ::= EPSILON
128. <formal_parameter_list> ::= ; <formal_parameter> <formal_parameter_list>
129. <formal_parameter_list> ::= EPSILON
130. <simple_type_list> ::= , <simple_type> <simple_type_list>
131. <simple_type_list> ::= EPSILON
132. <constant_list> ::= , <constant> <constant_list>
133. <constant_list> ::= EPSILON
134. <identifier_list> ::= , IDENTIFIER <identifier_list>
135. <identifier_list> ::= EPSILON

NONTERMINAL SYMBOLS	REFERENCES
1 <term>	103 104 105 106 108
2 <type>	20 22 24 25 26 27 33 35 39 49 51
3 <block>	1 2 55 56
4 <factor>	108 109 111 112 113 114 115 116 117 118
5 <program>	1
6 <variant>	38 45 46 47
7 <constant>	7 9 11 12 29 31 45 87 132
8 <set_range>	117 121 122
9 <statement>	63 64 66 67 69 70 72 73 75 86 87 91 93 94
10 <term_list>	104 105 106 107
11 <expression>	70 71 72 73 74 78 81 85 93 94 97 99 118 123 124
12 <field_list>	34 37 38 41 45
13 <else_clause>	70 86
14 <factor_list>	108 109 110
15 <set_element>	121 123 126
16 <simple_type>	24 28 29 30 33 36 130
17 <case_element>	71 87 88 89
18 <variant_list>	38 47 48
19 <constant_list>	45 87 132 133
20 <record_section>	37 39 40
21 <statement_part>	2 63
22 <expression_list>	81 85 97 98
23 <identifier_list>	1 28 39 49 51 59 60 61 62 134 135
24 <formal_parameter>	57 59 60 61 62 128
25 <set_element_list>	121 126 127
26 <simple_type_list>	33 130 131
27 <case_element_list>	71 89 90
28 <simple_expression>	99 100 101 103 104
29 <variable_selectors>	75 78 81 82 83 84 95 119
30 <with_variable_list>	75 95 96
31 <tag_field_remainder>	38 43 44
32 <unlabeled_statement>	66 67 68 69 70 71 72 73 74 75 76 77
33 <field_list_remainder>	37 41 42
34 <identifier_remainder>	112 119 120
35 <type_definition_part>	2 20 21
36 <actual_parameter_part>	79 85 120
37 <formal_parameter_list>	57 128 129
38 <formal_parameter_part>	55 56 57 58
39 <repeat_statement_list>	73 91 92
40 <set_element_remainder>	123 124 125
41 <simple_type_remainder>	30 31 32
42 <label_declaration_part>	2 3 4
43 <for_statement_remainder>	74 93 94
44 <non_identifier_constant>	11 13 14 15 16 29
45 <unpacked_structure_type>	25 26 33 34 35 36
46 <constant_definition_part>	2 7 8
47 <proc_and_func_declaration>	53 55 56
48 <variable_declaration_part>	2 49 50
49 <simple_statement_remainder>	68 78 79 80
50 <simple_expression_remainder>	99 100 101 102
51 <compound_statement_remainder>	63 64 65 69
52 <constant_definition_part_remai>	7 9 10 *** SYMBOL TRUNCATED ***
53 <label_declaration_part_remaind>	3 5 6 *** SYMBOL TRUNCATED ***
54 <non_identifier_constant_remain>	13 17 18 19 *** SYMBOL TRUNCATED ***
55 <proc_and_func_declaration_part>	2 53 54
56 <type_definition_part_remainder>	20 22 23
57 <variable_declaration_part_rema>	49 51 52 *** SYMBOL TRUNCATED ***

TERMINAL SYMBOLS	REFERENCES
58 .	1 82
59 (	1 28 45 57 85 118
60)	1 28 45 57 85 118
61 ;	1 6 7 9 20 22 41 47 49 51 53 55 56 64 89 91 128
62 ,	5 95 97 126 130 132 134
63 :	39 43 45 49 51 56 59 60 61 66 87
64 =	7 9 20 22 100
65 [	33 81 117
66]	33 81 117
67 ..	29 31 124
68 :=	74 78
69 DO	72 93 94 96
70 IF	70
71 OF	33 35 36 38 71
72 OR	106
73 TO	93
74 END	34 65 90
75 FOR	74
76 NIL	116
77 NOT	111
78 SET	36
79 VAR	49 60
80 CASE	38 71
81 ELSE	86
82 FILE	35
83 GOTO	76
84 THEN	70
85 TYPE	20
86 WITH	75
87 ARRAY	33
88 BEGIN	63 69
89 CONST	7
90 LABEL	3
91 UNTIL	92
92 WHILE	72
93 DOWNTO	94
94 PACKED	25
95 RECORD	34
96 REPEAT	73
97 STRING	16 113
98 POINTER	27 83
99 PROGRAM	1
100 FUNCTION	56 61
101 PLUSMINUS	13 103 105
102 PROCEDURE	55 62
103 IDENTIFIER	1 7 9 12 17 20 22 27 28 30 38 39 43 49 51 55 56 59 60 61 62 68 74 75 82 95 112 134
104 UNSIGNEDREAL	15 19 115
105 UNSIGNEDINTEGER	3 5 14 18 66 76 114
106 RELATIONOPERATOR	101
107 MULTIPLYINGOPERATOR	109
-EPSILON-	4 8 10 21 23 32 40 42 44 46 48 50 52 54 58 77 80 84 88 98

102 107 110 122 125 127 129 131 133 135

NULLABLE PRODUCTION NUMBERS --- 37 67 119

SELECTION SETS

PROD#	SELECTION SET ITEMS
1	--- PROGRAM
2	--- VAR TYPE BEGIN CONST LABEL FUNCTION PROCEDURE
3	--- LABEL
4	--- VAR TYPE BEGIN CONST FUNCTION PROCEDURE
5	--- ,
6	--- ;
7	--- CONST
8	--- VAR TYPE BEGIN FUNCTION PROCEDURE
9	--- IDENTIFIER
10	--- VAR TYPE BEGIN FUNCTION PROCEDURE
11	--- STRING PLUSMINUS UNSIGNEDREAL UNSIGNEDINTEGER
12	--- IDENTIFIER
13	--- PLUSMINUS
14	--- UNSIGNEDINTEGER
15	--- UNSIGNEDREAL
16	--- STRING
17	--- IDENTIFIER
18	--- UNSIGNEDINTEGER
19	--- UNSIGNEDREAL
20	--- TYPE
21	--- VAR BEGIN FUNCTION PROCEDURE
22	--- IDENTIFIER
23	--- VAR BEGIN FUNCTION PROCEDURE
24	--- (STRING PLUSMINUS IDENTIFIER UNSIGNEDREAL UNSIGNEDINTEGER
25	--- PACKED
26	--- SET FILE ARRAY RECORD
27	--- POINTER
28	--- (

```
29  --- STRING  PLUSMINUS  UNSIGNEDREAL  UNSIGNEDINTEGER
30  --- IDENTIFIER
31  --- ..
32  --- )  ;  ,  ]  END
33  --- ARRAY
34  --- RECORD
35  --- FILE
36  --- SET
37  --- )  ;  END  IDENTIFIER
38  --- CASE
39  --- IDENTIFIER
40  --- )  ;  END
41  --- ;
42  --- )  END
43  --- :
44  --- OF
45  --- STRING  PLUSMINUS  IDENTIFIER  UNSIGNEDREAL  UNSIGNEDINTEGER
46  --- )  ;  END
47  --- ;
48  --- )  END
49  --- VAR
50  --- BEGIN  FUNCTION  PROCEDURE
51  --- IDENTIFIER
52  --- BEGIN  FUNCTION  PROCEDURE
53  --- FUNCTION  PROCEDURE
54  --- BEGIN
55  --- PROCEDURE
56  --- FUNCTION
57  --- (
58  --- ;  :
```

```
   59  --- IDENTIFIER
   60  --- VAR
   61  --- FUNCTION
   62  --- PROCEDURE
   63  --- BEGIN
   64  --- ;
   65  --- END
   66  --- UNSIGNEDINTEGER
   67  --- ;  IF  END  FOR  CASE  ELSE  GOTO  WITH  BEGIN  UNTIL  WHILE  REPEAT  IDENTIFIER
   68  --- IDENTIFIER
   69  --- BEGIN
   70  --- IF
   71  --- CASE
   72  --- WHILE
   73  --- REPEAT
   74  --- FOR
   75  --- WITH
   76  --- GOTO
   77  --- ;  END  ELSE  UNTIL
   78  --- .  [  :=  POINTER
   79  --- (
   80  --- ;  END  ELSE  UNTIL
   81  --- [
   82  --- .
   83  --- POINTER
   84  --- )  ;  ,  =  ]  ..  :=  DO  OF  OR  TO  END  ELSE  THEN  UNTIL  DOWNTO  PLUSMINUS  RELATIONOPERATOR  MULTIPLYI
NGOPERATOR
   85  --- (
   86  --- ELSE
   87  --- STRING  PLUSMINUS  IDENTIFIER  UNSIGNEDREAL  UNSIGNEDINTEGER
   88  --- ;  END
```

```
 89  --- ;
 90  --- END
 91  --- ;
 92  --- UNTIL
 93  --- TO
 94  --- DOWNTO
 95  --- ,
 96  --- DO
 97  --- ,
 98  --- )  ]
 99  --- (  [  NIL  NOT  STRING  PLUSMINUS  IDENTIFIER  UNSIGNEDREAL  UNSIGNEDINTEGER
100  --- =
101  --- RELATIONOPERATOR
102  --- )  ;  ,  ]  ..  DO  OF  TO  END  ELSE  THEN  UNTIL  DOWNTO
103  --- PLUSMINUS
104  --- (  [  NIL  NOT  STRING  IDENTIFIER  UNSIGNEDREAL  UNSIGNEDINTEGER
105  --- PLUSMINUS
106  --- OR
107  --- )  ;  ,  =  ]  ..  DO  OF  TO  END  ELSE  THEN  UNTIL  DOWNTO  RELATIONOPERATOR
108  --- (  [  NIL  NOT  STRING  IDENTIFIER  UNSIGNEDREAL  UNSIGNEDINTEGER
109  --- MULTIPLYINGOPERATOR
110  --- )  ;  ,  =  ]  ..  DO  OF  OR  TO  END  ELSE  THEN  UNTIL  DOWNTO  PLUSMINUS  RELATIONOPERATOR
111  --- NOT
112  --- IDENTIFIER
113  --- STRING
114  --- UNSIGNEDINTEGER
115  --- UNSIGNEDREAL
116  --- NIL
117  --- [
118  --- (
```

```
119  ---  .  )  ;  ,  =  [  ]  ..  DO  OF  OR  TO  END  ELSE  THEN  UNTIL  DOWNTO  POINTER  PLUSMINUS  RELATIONOPERATOR
MULTIPLYINGOPERATOR
120  ---  (
121  ---  (  [  NIL  NOT  STRING  PLUSMINUS  IDENTIFIER  UNSIGNEDREAL  UNSIGNEDINTEGER
122  ---  ]
123  ---  (  [  NIL  NOT  STRING  PLUSMINUS  IDENTIFIER  UNSIGNEDREAL  UNSIGNEDINTEGER
124  ---  ..
125  ---  ,  ]
126  ---  ,
127  ---  ]
128  ---  ;
129  ---  )
130  ---  ,
131  ---  ]
132  ---  ,
133  ---  :
134  ---  ,
135  ---  )  ;  :
```

P R O C E D U R E S

```
    1.  REPLACE(  IDENTIFIER
                  (
                  IDENTIFIER
                 <identifier_list>
                  )
                  ;
                 <block>
                  . )
        ADVANCE
-----------------------------------------
    2.  REPLACE( <label_declaration_part>
                 <constant_definition_part>
                 <type_definition_part>
                 <variable_declaration_part>
                 <proc_and_func_declaration_part>
                 <statement_part> )
        RETAIN
-----------------------------------------
    3.  REPLACE(  UNSIGNEDINTEGER
                 <label_declaration_part_remaind> )
        ADVANCE
-----------------------------------------
    4.  POP
        RETAIN
-----------------------------------------
    5.  REPLACE(  UNSIGNEDINTEGER
                 <label_declaration_part_remaind> )
        ADVANCE
-----------------------------------------
    6.  POP
        ADVANCE
-----------------------------------------
    7.  REPLACE(  IDENTIFIER
                  =
                 <constant>
                  ;
                 <constant_definition_part_remai> )
        ADVANCE
-----------------------------------------
    8.  POP
        RETAIN
-----------------------------------------
    9.  REPLACE(  =
                 <constant>
                  ;
                 <constant_definition_part_remai> )
        ADVANCE
-----------------------------------------
   10.  POP
        RETAIN
-----------------------------------------
   11.  REPLACE( <non_identifier_constant> )
        RETAIN
-----------------------------------------
   12.  POP
        ADVANCE
-----------------------------------------
```

```
13.  REPLACE( <non_identifier_constant_remain> )
     ADVANCE
-----------------------------------------
14.  POP
     ADVANCE
-----------------------------------------
15.  POP
     ADVANCE
-----------------------------------------
16.  POP
     ADVANCE
-----------------------------------------
17.  POP
     ADVANCE
-----------------------------------------
18.  POP
     ADVANCE
-----------------------------------------
19.  POP
     ADVANCE
-----------------------------------------
20.  REPLACE(  IDENTIFIER
               =
               <type>
               ;
               <type_definition_part_remainder> )
     ADVANCE
-----------------------------------------
21.  POP
     RETAIN
-----------------------------------------
22.  REPLACE(  =
               <type>
               ;
               <type_definition_part_remainder> )
     ADVANCE
-----------------------------------------
23.  POP
     RETAIN
-----------------------------------------
24.  REPLACE( <simple_type> )
     RETAIN
-----------------------------------------
25.  REPLACE( <unpacked_structure_type> )
     ADVANCE
-----------------------------------------
26.  REPLACE( <unpacked_structure_type> )
     RETAIN
-----------------------------------------
27.  REPLACE(  IDENTIFIER )
     ADVANCE
-----------------------------------------
28.  REPLACE(  IDENTIFIER
               <identifier_list>
               ) )
     ADVANCE
-----------------------------------------
29.  REPLACE( <non_identifier_constant>
               ..
               <constant> )
```

```
        RETAIN
-----------------------------------------
   30.  REPLACE( <simple_type_remainder> )
        ADVANCE
-----------------------------------------
   31.  REPLACE( <constant> )
        ADVANCE
-----------------------------------------
   32.  POP
        RETAIN
-----------------------------------------
   33.  REPLACE(  [
                  <simple_type>
                  <simple_type_list>
                   ]
                   OF
                  <type> )
        ADVANCE
-----------------------------------------
   34.  REPLACE( <field_list>
                  END )
        ADVANCE
-----------------------------------------
   35.  REPLACE(  OF
                  <type> )
        ADVANCE
-----------------------------------------
   36.  REPLACE(  OF
                  <simple_type> )
        ADVANCE
-----------------------------------------
   37.  REPLACE( <record_section>
                  <field_list_remainder> )
        RETAIN
-----------------------------------------
   38.  REPLACE(  IDENTIFIER
                  <tag_field_remainder>
                   OF
                  <variant>
                  <variant_list> )
        ADVANCE
-----------------------------------------
   39.  REPLACE( <identifier_list>
                   :
                  <type> )
        ADVANCE
-----------------------------------------
   40.  POP
        RETAIN
-----------------------------------------
   41.  REPLACE( <field_list> )
        ADVANCE
-----------------------------------------
   42.  POP
        RETAIN
-----------------------------------------
   43.  REPLACE(  IDENTIFIER )
        ADVANCE
-----------------------------------------
   44.  POP
```

```
        RETAIN
-----------------------------------------
   45.  REPLACE( <constant>
                 <constant_list>
                  :
                  (
                 <field_list>
                  ) )
        RETAIN
-----------------------------------------
   46.  POP
        RETAIN
-----------------------------------------
   47.  REPLACE( <variant>
                 <variant_list> )
        ADVANCE
-----------------------------------------
   48.  POP
        RETAIN
-----------------------------------------
   49.  REPLACE(  IDENTIFIER
                 <identifier_list>
                  :
                 <type>
                  ;
                 <variable_declaration_part_rema> )
        ADVANCE
-----------------------------------------
   50.  POP
        RETAIN
-----------------------------------------
   51.  REPLACE( <identifier_list>
                  :
                 <type>
                  ;
                 <variable_declaration_part_rema> )
        ADVANCE
-----------------------------------------
   52.  POP
        RETAIN
-----------------------------------------
   53.  REPLACE( <proc_and_func_declaration>
                  ;
                 <proc_and_func_declaration_part> )
        RETAIN
-----------------------------------------
   54.  POP
        RETAIN
-----------------------------------------
   55.  REPLACE(  IDENTIFIER
                 <formal_parameter_part>
                  ;
                 <block> )
        ADVANCE
-----------------------------------------
   56.  REPLACE(  IDENTIFIER
                 <formal_parameter_part>
                  :
                  IDENTIFIER
                  ;
```

```
                <block> )
        ADVANCE
-----------------------------------------
   57.  REPLACE( <formal_parameter>
                 <formal_parameter_list>
                  ) )
        ADVANCE
-----------------------------------------
   58.  POP
        RETAIN
-----------------------------------------
   59.  REPLACE( <identifier_list>
                  :
                  IDENTIFIER )
        ADVANCE
-----------------------------------------
   60.  REPLACE(  IDENTIFIER
                 <identifier_list>
                  :
                  IDENTIFIER )
        ADVANCE
-----------------------------------------
   61.  REPLACE(  IDENTIFIER
                 <identifier_list>
                  :
                  IDENTIFIER )
        ADVANCE
-----------------------------------------
   62.  REPLACE(  IDENTIFIER
                 <identifier_list> )
        ADVANCE
-----------------------------------------
   63.  REPLACE( <statement>
                 <compound_statement_remainder> )
        ADVANCE
-----------------------------------------
   64.  REPLACE( <statement>
                 <compound_statement_remainder> )
        ADVANCE
-----------------------------------------
   65.  POP
        ADVANCE
-----------------------------------------
   66.  REPLACE(  :
                 <unlabeled_statement> )
        ADVANCE
-----------------------------------------
   67.  REPLACE( <unlabeled_statement> )
        RETAIN
-----------------------------------------
   68.  REPLACE( <simple_statement_remainder> )
        ADVANCE
-----------------------------------------
   69.  REPLACE( <statement>
                 <compound_statement_remainder> )
        ADVANCE
-----------------------------------------
   70.  REPLACE( <expression>
                  THEN
                 <statement>
```

```
                <else_clause> )
        ADVANCE
-----------------------------------------
   71.  REPLACE( <expression>
                  OF
                 <case_element>
                 <case_element_list> )
        ADVANCE
-----------------------------------------
   72.  REPLACE( <expression>
                  DO
                 <statement> )
        ADVANCE
-----------------------------------------
   73.  REPLACE( <statement>
                 <repeat_statement_list>
                 <expression> )
        ADVANCE
-----------------------------------------
   74.  REPLACE(  IDENTIFIER
                  :=
                 <expression>
                 <for_statement_remainder> )
        ADVANCE
-----------------------------------------
   75.  REPLACE(  IDENTIFIER
                 <variable_selectors>
                 <with_variable_list>
                 <statement> )
        ADVANCE
-----------------------------------------
   76.  REPLACE(  UNSIGNEDINTEGER )
        ADVANCE
-----------------------------------------
   77.  POP
        RETAIN
-----------------------------------------
   78.  REPLACE( <variable_selectors>
                  :=
                 <expression> )
        RETAIN
-----------------------------------------
   79.  REPLACE( <actual_parameter_part> )
        RETAIN
-----------------------------------------
   80.  POP
        RETAIN
-----------------------------------------
   81.  REPLACE( <expression>
                 <expression_list>
                  ]
                 <variable_selectors> )
        ADVANCE
-----------------------------------------
   82.  REPLACE(  IDENTIFIER
                 <variable_selectors> )
        ADVANCE
-----------------------------------------
   83.  REPLACE( <variable_selectors> )
        ADVANCE
```

```
---------------------------------------
   84.  POP
        RETAIN
---------------------------------------
   85.  REPLACE( <expression>
                  <expression_list>
                   ) )
        ADVANCE
---------------------------------------
   86.  REPLACE( <statement> )
        ADVANCE
---------------------------------------
   87.  REPLACE( <constant>
                  <constant_list>
                   :
                  <statement> )
        RETAIN
---------------------------------------
   88.  POP
        RETAIN
---------------------------------------
   89.  REPLACE( <case_element>
                  <case_element_list> )
        ADVANCE
---------------------------------------
   90.  POP
        ADVANCE
---------------------------------------
   91.  REPLACE( <statement>
                  <repeat_statement_list> )
        ADVANCE
---------------------------------------
   92.  POP
        ADVANCE
---------------------------------------
   93.  REPLACE( <expression>
                   DO
                  <statement> )
        ADVANCE
---------------------------------------
   94.  REPLACE( <expression>
                   DO
                  <statement> )
        ADVANCE
---------------------------------------
   95.  REPLACE(  IDENTIFIER
                  <variable_selectors>
                  <with_variable_list> )
        ADVANCE
---------------------------------------
   96.  POP
        ADVANCE
---------------------------------------
   97.  REPLACE( <expression> )
        ADVANCE
---------------------------------------
   98.  POP
        RETAIN
---------------------------------------
   99.  REPLACE( <simple_expression>
```

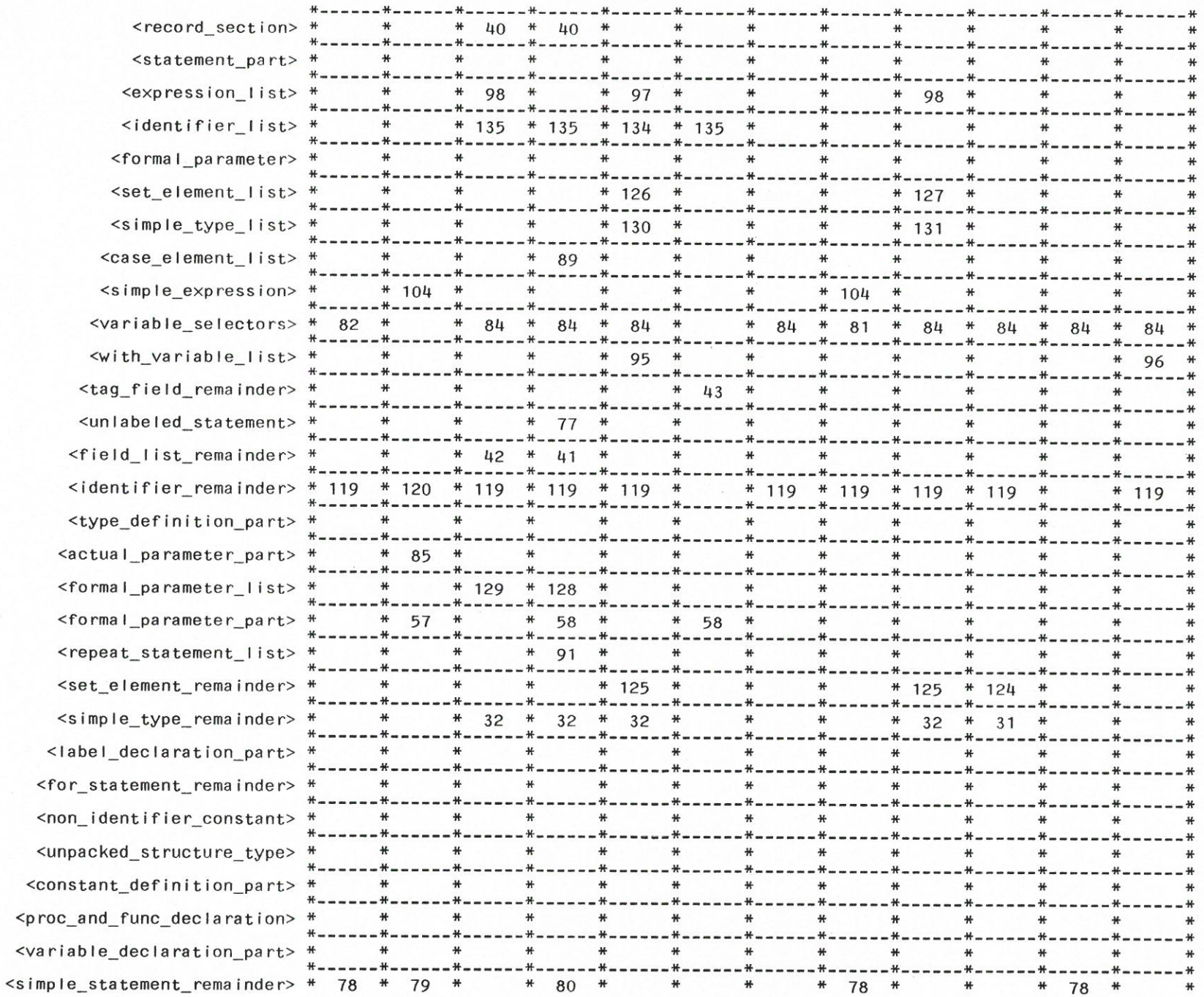
```
                <simple_expression_remainder> )
        RETAIN
-----------------------------------------
  100.  REPLACE( <simple_expression> )
        ADVANCE
-----------------------------------------
  101.  REPLACE( <simple_expression> )
        ADVANCE
-----------------------------------------
  102.  POP
        RETAIN
-----------------------------------------
  103.  REPLACE( <term> )
        ADVANCE
-----------------------------------------
  104.  REPLACE( <term>
                <term_list> )
        RETAIN
-----------------------------------------
  105.  REPLACE( <term>
                <term_list> )
        ADVANCE
-----------------------------------------
  106.  REPLACE( <term>
                <term_list> )
        ADVANCE
-----------------------------------------
  107.  POP
        RETAIN
-----------------------------------------
  108.  REPLACE( <factor>
                <factor_list> )
        RETAIN
-----------------------------------------
  109.  REPLACE( <factor>
                <factor_list> )
        ADVANCE
-----------------------------------------
  110.  POP
        RETAIN
-----------------------------------------
  111.  REPLACE( <factor> )
        ADVANCE
-----------------------------------------
  112.  REPLACE( <identifier_remainder> )
        ADVANCE
-----------------------------------------
  113.  POP
        ADVANCE
-----------------------------------------
  114.  POP
        ADVANCE
-----------------------------------------
  115.  POP
        ADVANCE
-----------------------------------------
  116.  POP
        ADVANCE
-----------------------------------------
  117.  REPLACE( <set_range>
```

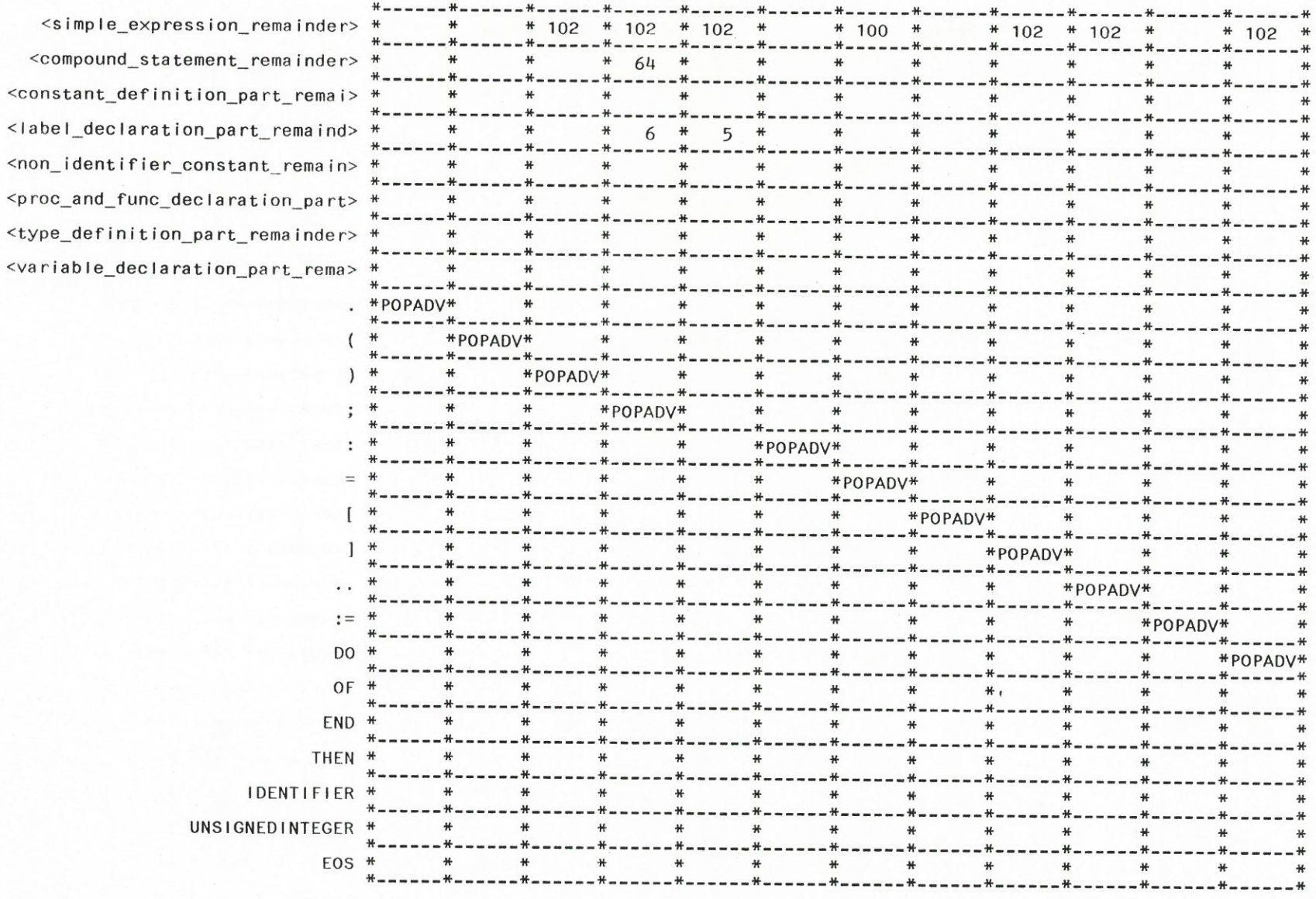
```
                  ] )
       ADVANCE
-------------------------------------
 118.  REPLACE( <expression>
                  ) )
       ADVANCE
-------------------------------------
 119.  REPLACE( <variable_selectors> )
       RETAIN
-------------------------------------
 120.  REPLACE( <actual_parameter_part> )
       RETAIN
-------------------------------------
 121.  REPLACE( <set_element>
                <set_element_list> )
       RETAIN
-------------------------------------
 122.  POP
       RETAIN
-------------------------------------
 123.  REPLACE( <expression>
                <set_element_remainder> )
       RETAIN
-------------------------------------
 124.  REPLACE( <expression> )
       ADVANCE
-------------------------------------
 125.  POP
       RETAIN
-------------------------------------
 126.  REPLACE( <set_element>
                <set_element_list> )
       ADVANCE
-------------------------------------
 127.  POP
       RETAIN
-------------------------------------
 128.  REPLACE( <formal_parameter>
                <formal_parameter_list> )
       ADVANCE
-------------------------------------
 129.  POP
       RETAIN
-------------------------------------
 130.  REPLACE( <simple_type>
                <simple_type_list> )
       ADVANCE
-------------------------------------
 131.  POP
       RETAIN
-------------------------------------
 132.  REPLACE( <constant>
                <constant_list> )
       ADVANCE
-------------------------------------
 133.  POP
       RETAIN
-------------------------------------
 134.  REPLACE(  IDENTIFIER
                <identifier_list> )
```

```
        ADVANCE
-----------------------------------
  135.  POP
        RETAIN
-----------------------------------
```

TRANSITION TABLE

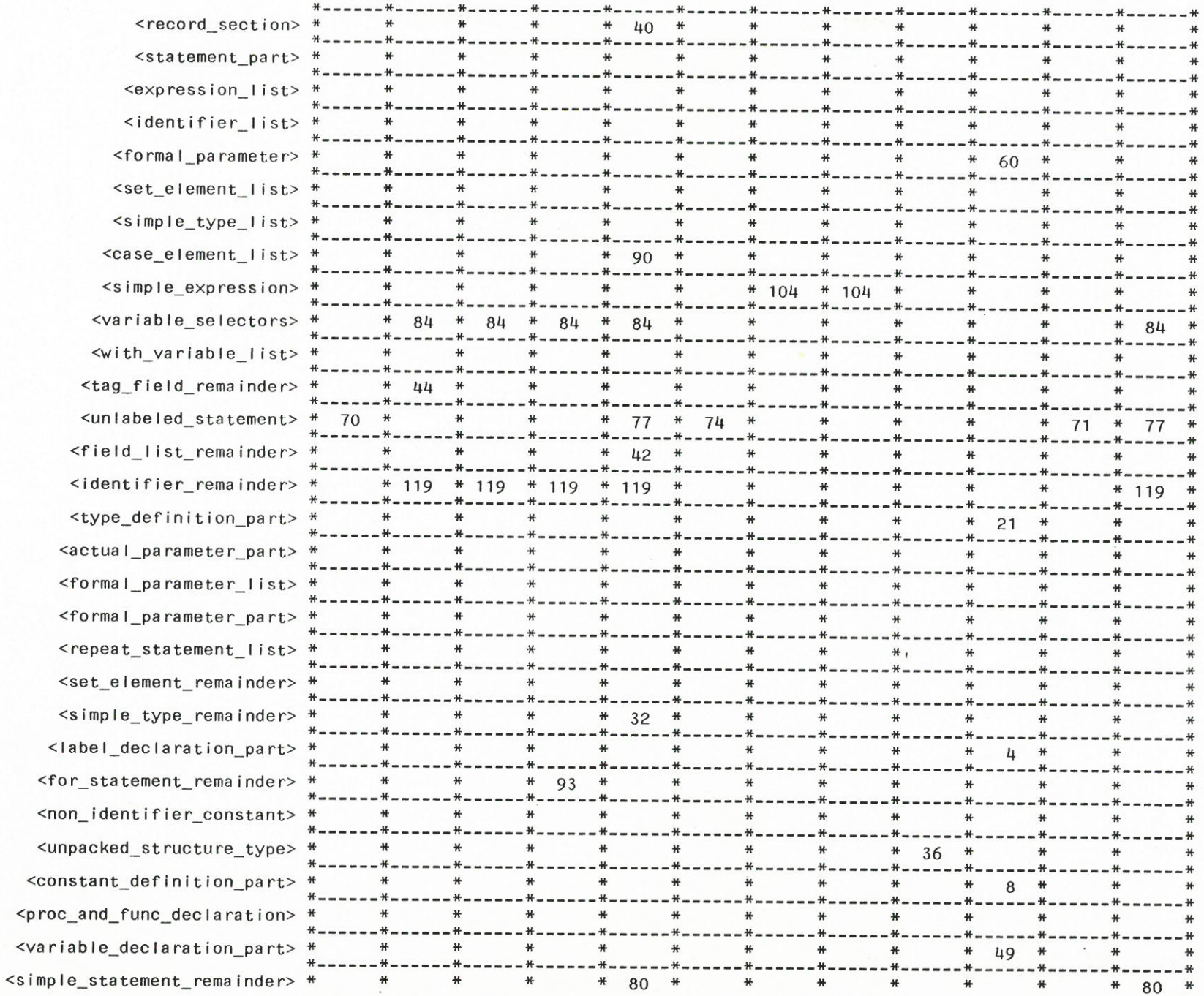
	.	(	)	;	,	:	=	[	]	.	: =	DO
<term>		108						108				
<type>		24										
<block>												
<factor>		118						117				
<program>												
<variant>			46	46								
<constant>												
<set_range>		121						121	122			
<statement>				67								
<term_list>			107	107	107		107		107	107		107
<expression>		99						99				
<field_list>			37	37								
<else_clause>												
<factor_list>			110	110	110		110		110	110		110
<set_element>		123						123				
<simple_type>		28										
<case_element>				88								
<variant_list>			48	47								
<constant_list>					132	133						

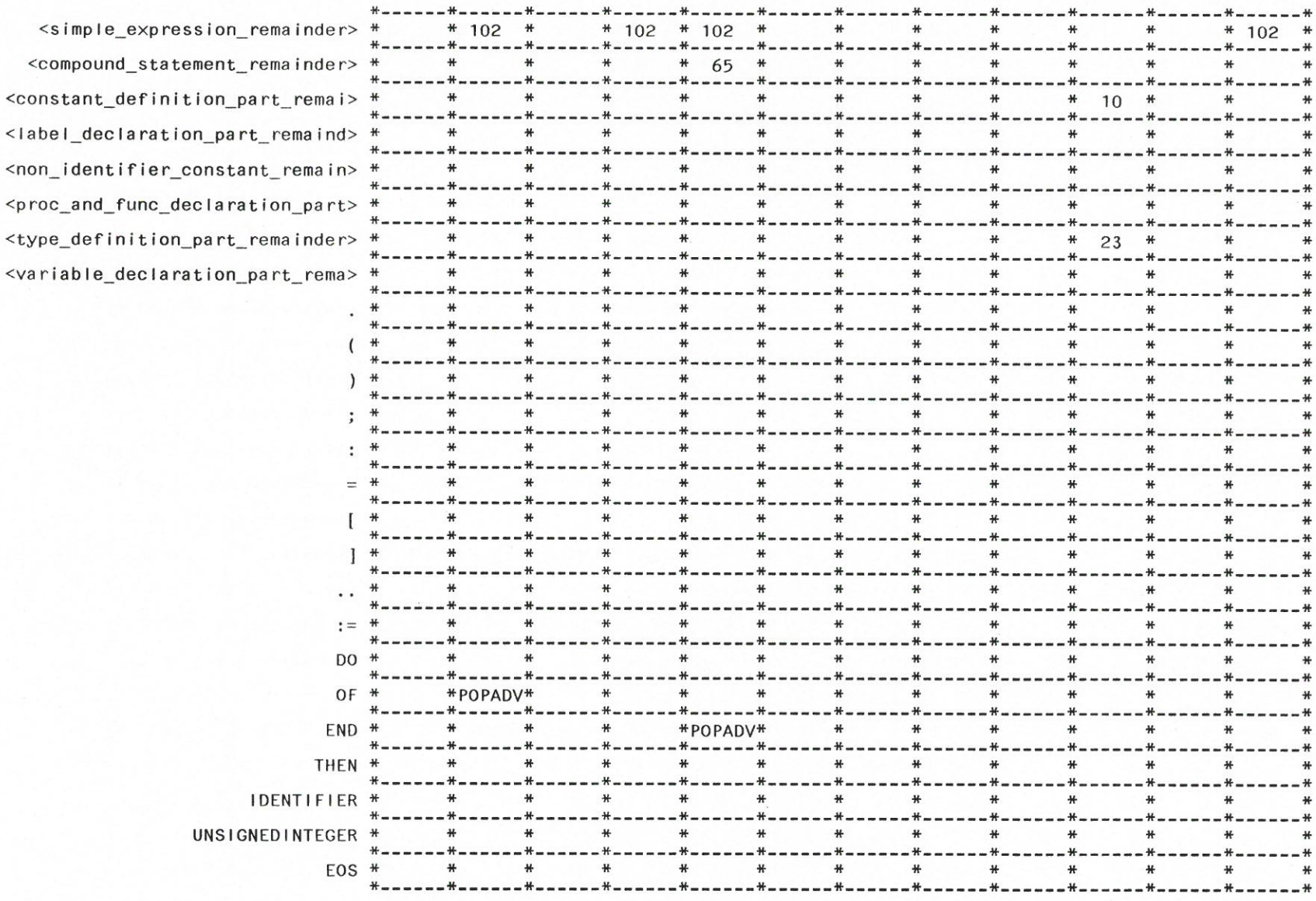

<record_section> 40 40
<statement_part>
<expression_list> 98 97 98
<identifier_list> 135 135 134 135
<formal_parameter>
<set_element_list> 126 127
<simple_type_list> 130 131
<case_element_list> 89
<simple_expression> 104 104
<variable_selectors> 82 84 84 84 84 84 81 84 84 84 84
<with_variable_list> 95 96
<tag_field_remainder> 43
<unlabeled_statement> 77
<field_list_remainder> 42 41
<identifier_remainder> 119 120 119 119 119 119 119 119 119 119 119
<type_definition_part>
<actual_parameter_part> 85
<formal_parameter_list> 129 128
<formal_parameter_part> 57 58 58
<repeat_statement_list> 91
<set_element_remainder> 125 125 124
<simple_type_remainder> 32 32 32 32 31
<label_declaration_part>
<for_statement_remainder>
<non_identifier_constant>
<unpacked_structure_type>
<constant_definition_part>
<proc_and_func_declaration>
<variable_declaration_part>
<simple_statement_remainder> 78 79 80 78


<simple_expression_remainder>
<compound_statement_remainder>
<constant_definition_part_remai>
<label_declaration_part_remaind>
<non_identifier_constant_remain>
<proc_and_func_declaration_part>
<type_definition_part_remainder>
<variable_declaration_part_rema>
.
(
)
;
:
=
[
]
..
:=
DO
OF
END
THEN
IDENTIFIER
UNSIGNEDINTEGER
EOS
102 102 102 100 102 102 102
64
6 5
POPADV

TRANSITION TABLE

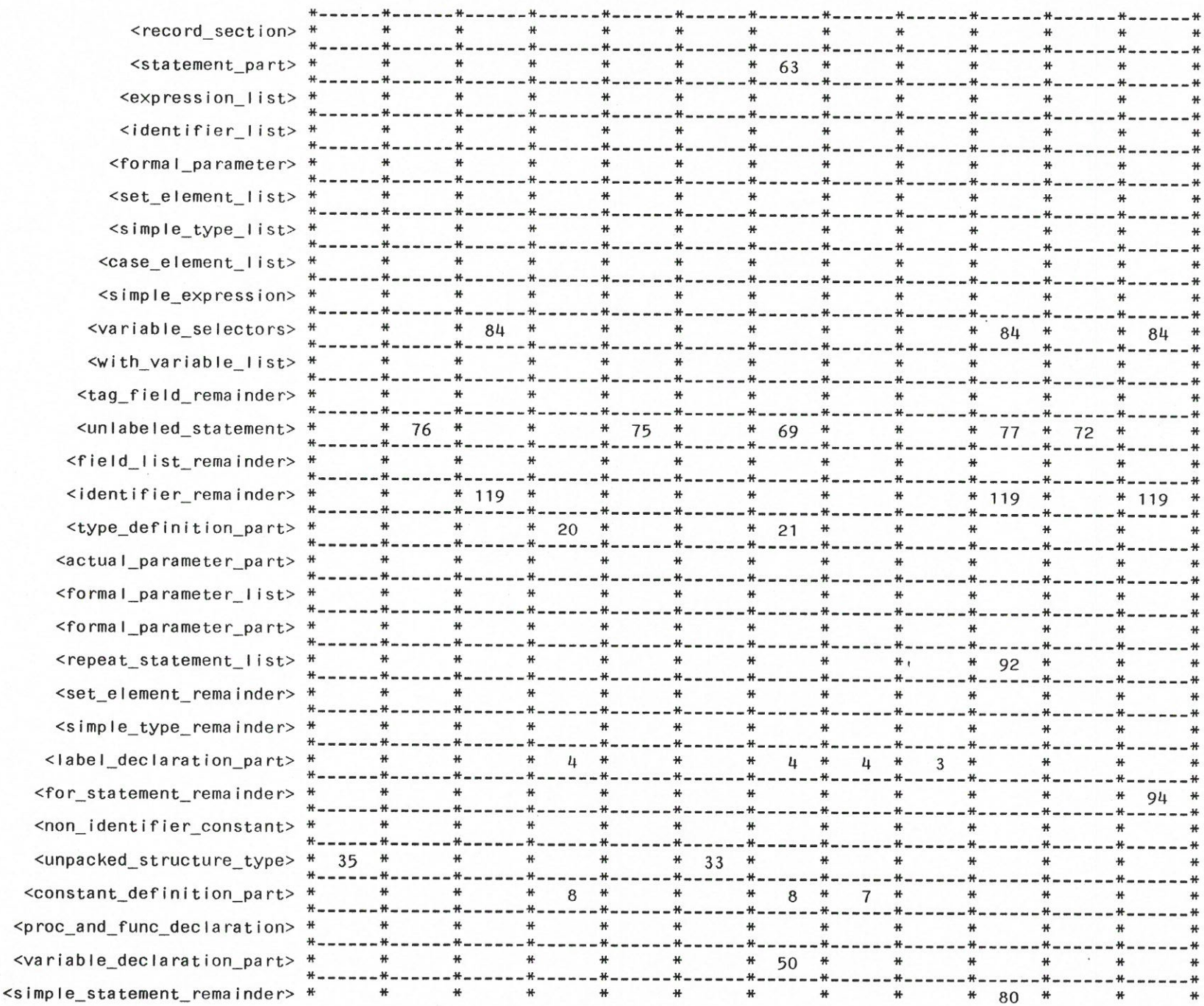
	IF	OF	OR	TO	END	FOR	NIL	NOT	SET	VAR	CASE	ELSE
<term>							108	108				
<type>									26			
<block>										2		
<factor>							116	111				
<program>												
<variant>					46							
<constant>												
<set_range>							121	121				
<statement>	67				67	67					67	67
<term_list>		107	106	107	107							107
<expression>							99	99				
<field_list>					37						38	
<else_clause>												86
<factor_list>		110	110	110	110							110
<set_element>							123	123				
<simple_type>												
<case_element>					88							
<variant_list>					48							
<constant_list>												

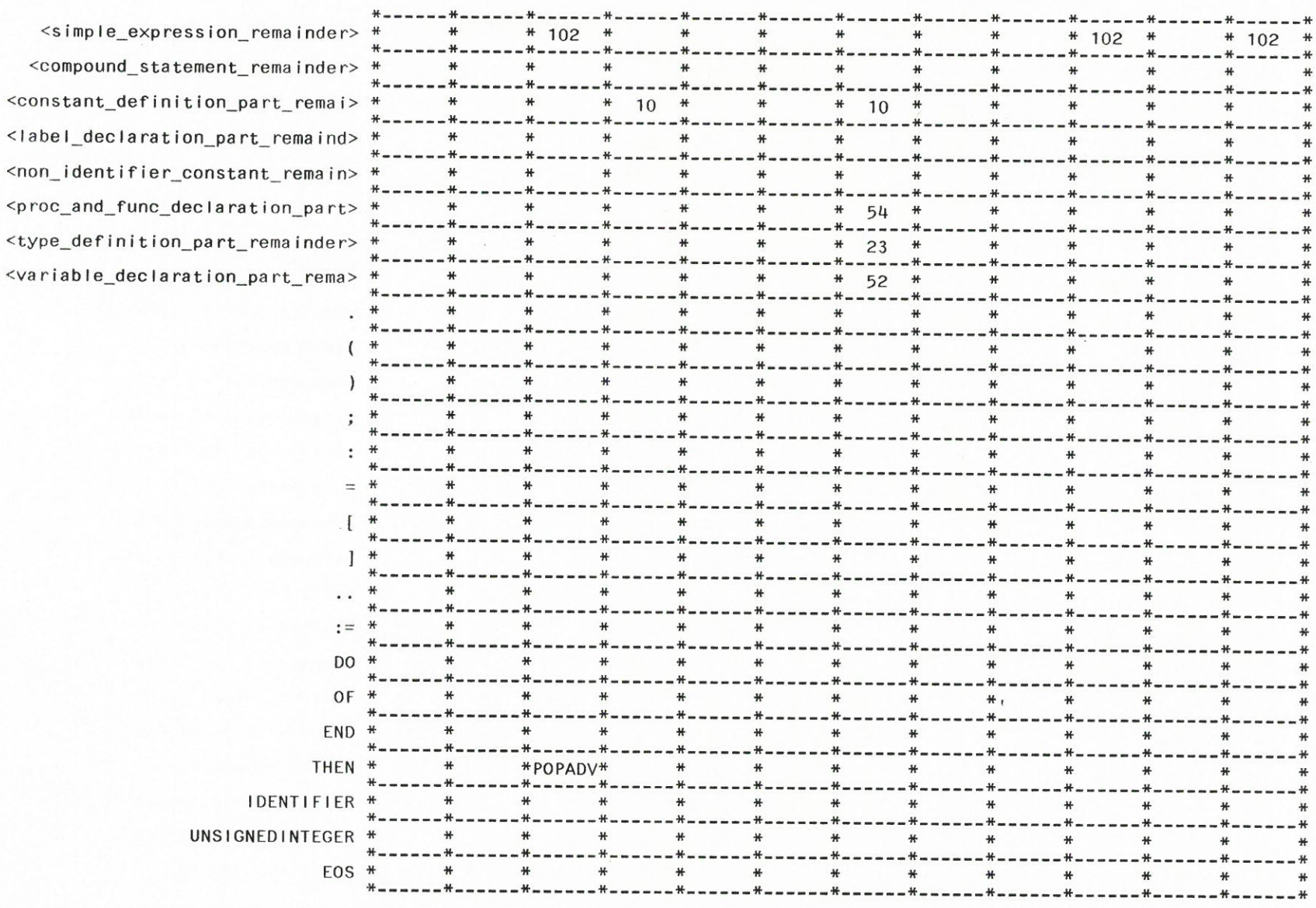

<record_section>
<statement_part>
<expression_list>
<identifier_list>
<formal_parameter>
<set_element_list>
<simple_type_list>
<case_element_list>
<simple_expression>
<variable_selectors>
<with_variable_list>
<tag_field_remainder>
<unlabeled_statement>
<field_list_remainder>
<identifier_remainder>
<type_definition_part>
<actual_parameter_part>
<formal_parameter_list>
<formal_parameter_part>
<repeat_statement_list>
<set_element_remainder>
<simple_type_remainder>
<label_declaration_part>
<for_statement_remainder>
<non_identifier_constant>
<unpacked_structure_type>
<constant_definition_part>
<proc_and_func_declaration>
<variable_declaration_part>
<simple_statement_remainder>


<simple_expression_remainder>
<compound_statement_remainder>
<constant_definition_part_remai>
<label_declaration_part_remaind>
<non_identifier_constant_remain>
<proc_and_func_declaration_part>
<type_definition_part_remainder>
<variable_declaration_part_rema>
.
(
)
;
:
=
[
]
..
:=
DO
OF
END
THEN
IDENTIFIER
UNSIGNEDINTEGER
EOS
102
102
102
102
65
10
23
POPADV
POPADV

TRANSITION TABLE

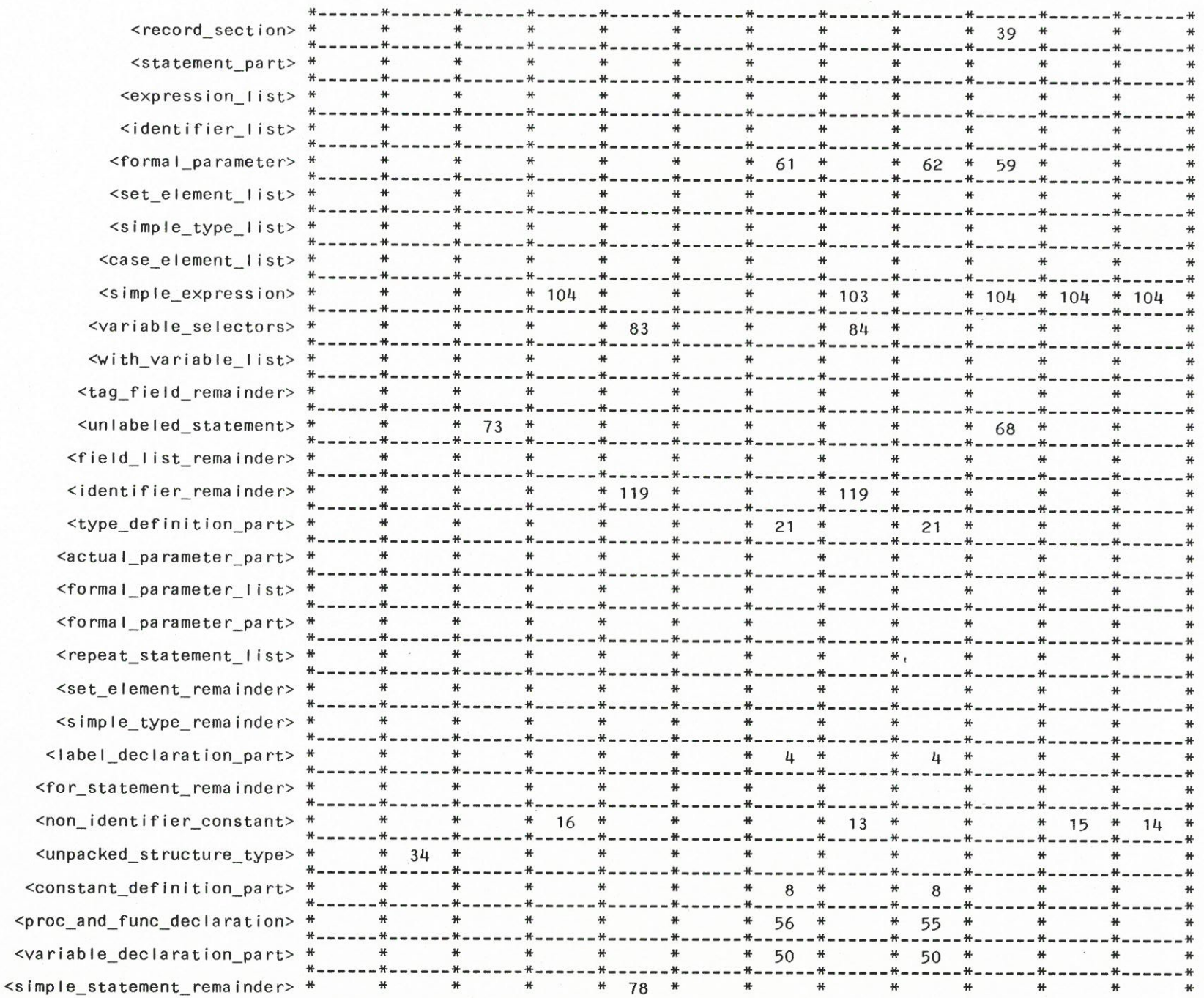
	FILE	GOTO	THEN	TYPE	WITH	ARRAY	BEGIN	CONST	LABEL	UNTIL	WHILE	DOWNTO
<term>												
<type>	26					26						
<block>				2			2	2	2			
<factor>												
<program>												
<variant>												
<constant>												
<set_range>												
<statement>		67			67		67			67	67	
<term_list>			107							107		107
<expression>												
<field_list>												
<else_clause>												
<factor_list>			110							110		110
<set_element>												
<simple_type>												
<case_element>												
<variant_list>												
<constant_list>												

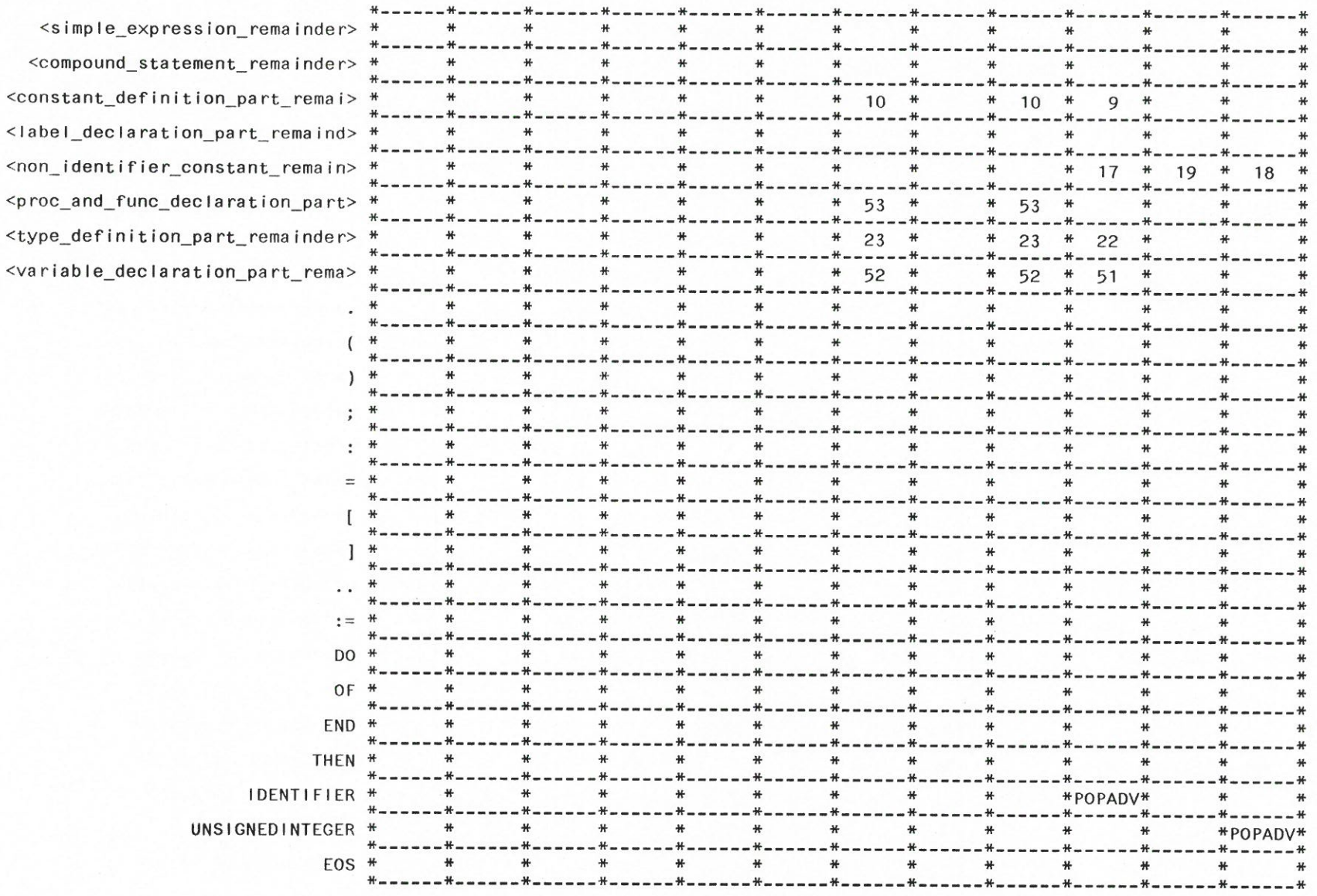

<record_section>
<statement_part>
<expression_list>
<identifier_list>
<formal_parameter>
<set_element_list>
<simple_type_list>
<case_element_list>
<simple_expression>
<variable_selectors>
<with_variable_list>
<tag_field_remainder>
<unlabeled_statement>
<field_list_remainder>
<identifier_remainder>
<type_definition_part>
<actual_parameter_part>
<formal_parameter_list>
<formal_parameter_part>
<repeat_statement_list>
<set_element_remainder>
<simple_type_remainder>
<label_declaration_part>
<for_statement_remainder>
<non_identifier_constant>
<unpacked_structure_type>
<constant_definition_part>
<proc_and_func_declaration>
<variable_declaration_part>
<simple_statement_remainder>


<simple_expression_remainder>
<compound_statement_remainder>
<constant_definition_part_remai>
<label_declaration_part_remaind>
<non_identifier_constant_remain>
<proc_and_func_declaration_part>
<type_definition_part_remainder>
<variable_declaration_part_rema>
.
(
)
;
:
=
[
]
..
:=
DO
OF
END
THEN
IDENTIFIER
UNSIGNEDINTEGER
EOS
102
102
102
10
10
54
23
52
POPADV

TRANSITION TABLE

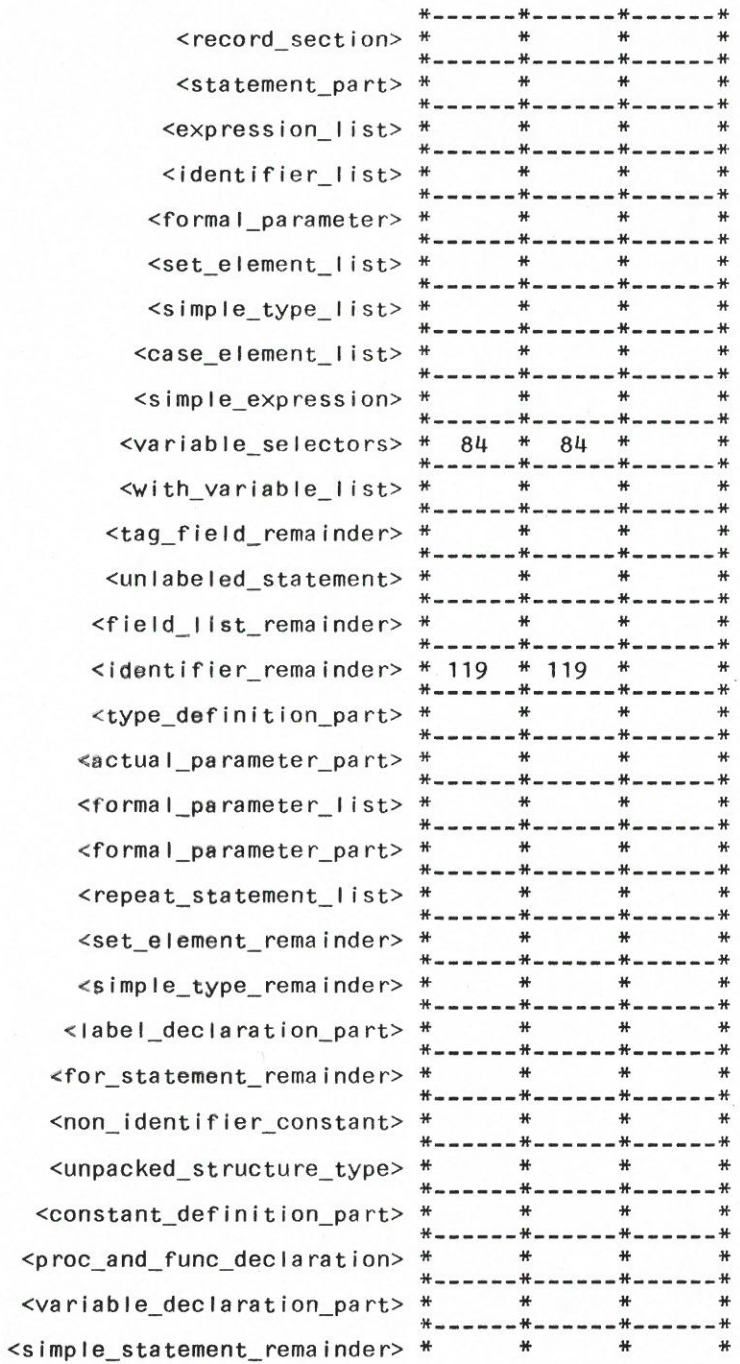
	PACKED	RECORD	REPEAT	STRING	POINTER	PROGRAM	FUNCTION	PLUSMINUS	PROCEDURE	IDENTIFIER	UNSIGNEDREAL	UNSIGNEDINTEGER
<term>				108						108	108	108
<type>	25	26		24	27			24		24	24	24
<block>							2		2			
<factor>				113						112	115	114
<program>						1						
<variant>				45				45		45	45	45
<constant>				11				11		12	11	11
<set_range>				121				121		121	121	121
<statement>			67							67		66
<term_list>								105				
<expression>				99				99		99	99	99
<field_list>										37		
<else_clause>												
<factor_list>								110				
<set_element>				123				123		123	123	123
<simple_type>				29				29		30	29	29
<case_element>				87				87		87	87	87
<variant_list>												
<constant_list>												

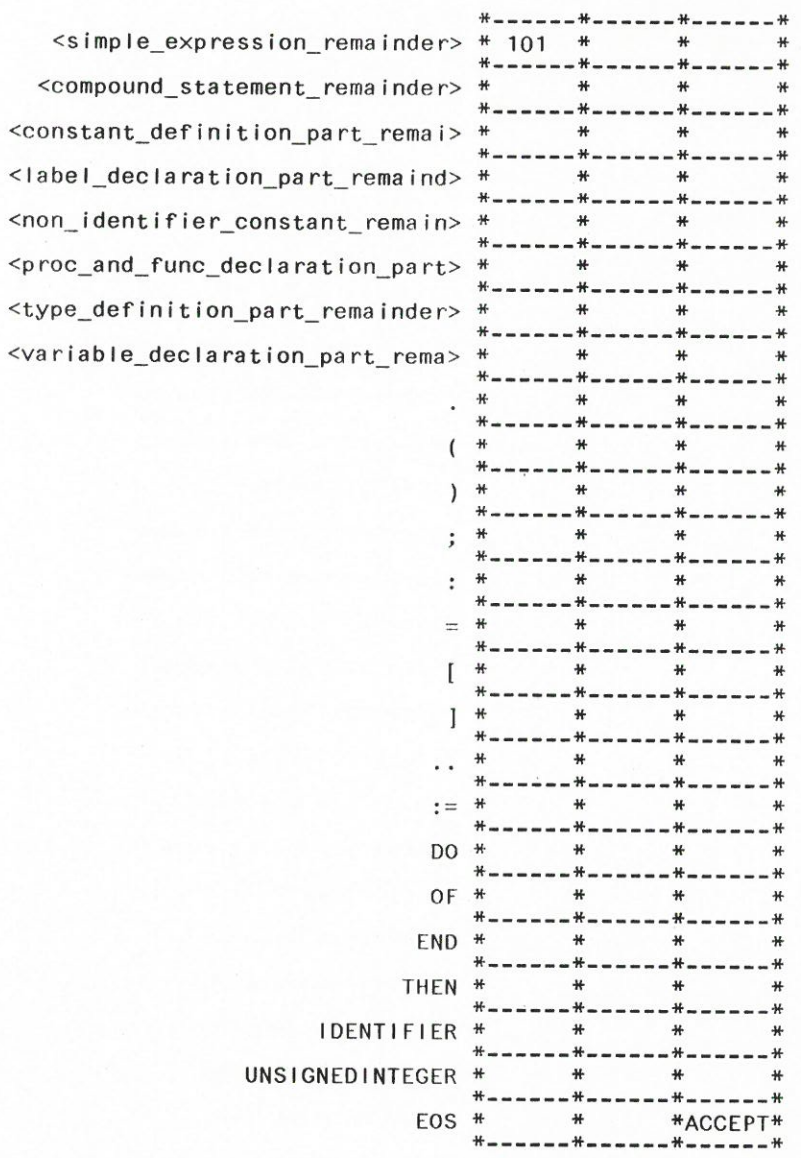

<record_section>
<statement_part>
<expression_list>
<identifier_list>
<formal_parameter>
<set_element_list>
<simple_type_list>
<case_element_list>
<simple_expression>
<variable_selectors>
<with_variable_list>
<tag_field_remainder>
<unlabeled_statement>
<field_list_remainder>
<identifier_remainder>
<type_definition_part>
<actual_parameter_part>
<formal_parameter_list>
<formal_parameter_part>
<repeat_statement_list>
<set_element_remainder>
<simple_type_remainder>
<label_declaration_part>
<for_statement_remainder>
<non_identifier_constant>
<unpacked_structure_type>
<constant_definition_part>
<proc_and_func_declaration>
<variable_declaration_part>
<simple_statement_remainder>
39
61
62
59
104
103
104
104
104
83
84
73
68
119
119
21
21
4
4
16
13
15
14
34
8
8
56
55
50
50
78


<simple_expression_remainder>
<compound_statement_remainder>
<constant_definition_part_remai>
<label_declaration_part_remaind>
<non_identifier_constant_remain>
<proc_and_func_declaration_part>
<type_definition_part_remainder>
<variable_declaration_part_rema>
.
(
)
;
:
=
[
]
..
:=
DO
OF
END
THEN
IDENTIFIER
UNSIGNEDINTEGER
EOS
10 10 9
53 53
23 23 22
52 52 51
17 19 18
POPADV
POPADV

TRANSITION TABLE

	RELATIONOPERATOR	MULTIPLYINGOPERATOR	END
<term>			
<type>			
<block>			
<factor>			
<program>			
<variant>			
<constant>			
<set_range>			
<statement>			
<term_list>	107		
<expression>			
<field_list>			
<else_clause>			
<factor_list>	110	109	
<set_element>			
<simple_type>			
<case_element>			
<variant_list>			
<constant_list>			


<record_section>
<statement_part>
<expression_list>
<identifier_list>
<formal_parameter>
<set_element_list>
<simple_type_list>
<case_element_list>
<simple_expression>
<variable_selectors>
84
84
<with_variable_list>
<tag_field_remainder>
<unlabeled_statement>
<field_list_remainder>
<identifier_remainder>
119
119
<type_definition_part>
<actual_parameter_part>
<formal_parameter_list>
<formal_parameter_part>
<repeat_statement_list>
<set_element_remainder>
<simple_type_remainder>
<label_declaration_part>
<for_statement_remainder>
<non_identifier_constant>
<unpacked_structure_type>
<constant_definition_part>
<proc_and_func_declaration>
<variable_declaration_part>
<simple_statement_remainder>


<simple_expression_remainder>
<compound_statement_remainder>
<constant_definition_part_remai>
<label_declaration_part_remaind>
<non_identifier_constant_remain>
<proc_and_func_declaration_part>
<type_definition_part_remainder>
<variable_declaration_part_rema>
.
(
)
;
:
=
[
]
..
:=
DO
OF
END
THEN
IDENTIFIER
UNSIGNEDINTEGER
EOS
101
ACCEPT