

## Context-free Grammar for Standard Pascal

In these productions, non-terminals are denoted in *italic typewriter* surrounded by angle brackets ( $\langle \rangle$ ). Literal terminals are in *typewriter*. Generic terminals are in **boldface**. The symbols ‘|’, ‘ $\emptyset$ ’ and ‘ $::=$ ’ are part of the BNF description language rather than *Pascal*. Items in roman are verbal descriptions.

A few features (bugs?) are omitted. These include the **set** data type, labels and the **goto** statement, the **with** statement, and variant records.

1.  $\langle Program \rangle ::= \langle ProgramHeader \rangle \langle Declarations \rangle \langle CompoundStatement \rangle .$
2.  $\langle ProgramHeader \rangle ::= \text{program id ;}$   
|  $\text{program id } ( \langle IdList \rangle ) ;$
3.  $\langle Declarations \rangle ::= \langle ConstantDefinitions \rangle \langle TypeDefinitions \rangle$   
|  $\langle VariableDeclarations \rangle \langle ProcedureDeclarations \rangle$
4.  $\langle ConstantDefinitions \rangle ::= \emptyset$   
|  $\text{const } \langle ConstantDefinitionList \rangle$
5.  $\langle ConstantDefinitionList \rangle ::= \langle ConstDef \rangle$   
|  $\langle ConstDef \rangle \langle ConstantDefinitionList \rangle$
6.  $\langle ConstDef \rangle ::= \text{id = constant ;}$
7.  $\langle TypeDefinitions \rangle ::= \emptyset$   
|  $\text{type } \langle TypeDefinitionList \rangle$
8.  $\langle TypeDefinitionList \rangle ::= \langle TypeDef \rangle$   
|  $\langle TypeDef \rangle \langle TypeDefinitionList \rangle$
9.  $\langle TypeDef \rangle ::= \text{id = } \langle TypeSpecifier \rangle ;$
10.  $\langle VariableDeclarations \rangle ::= \emptyset$   
|  $\text{var } \langle VariableDeclarationList \rangle$
11.  $\langle VariableDeclarationList \rangle ::= \langle VarDec \rangle$   
|  $\langle VarDec \rangle \langle VariableDeclarationList \rangle$
12.  $\langle VarDec \rangle ::= \langle IdList \rangle : \langle TypeSpecifier \rangle ;$
13.  $\langle ProcedureDeclarations \rangle ::= \emptyset$   
|  $\langle ProcDec \rangle \langle ProcedureDeclarations \rangle$
14.  $\langle ProcDec \rangle ::= \langle ProcHeader \rangle \text{forward ;}$   
|  $\langle ProcHeader \rangle \langle Declarations \rangle \langle CompoundStatement \rangle ;$   
|  $\langle FuncHeader \rangle \text{forward ;}$   
|  $\langle FuncHeader \rangle \langle Declarations \rangle \langle CompoundStatement \rangle ;$
15.  $\langle ProcHeader \rangle ::= \text{procedure id } \langle Arguments \rangle ;$
16.  $\langle FuncHeader \rangle ::= \text{function id } \langle Arguments \rangle : \langle TypeSpecifier \rangle ;$
17.  $\langle Arguments \rangle ::= \emptyset$   
|  $( \langle ArgumentList \rangle )$
18.  $\langle ArgumentList \rangle ::= \langle Arg \rangle$   
|  $\langle Arg \rangle ; \langle ArgumentList \rangle$

19.  $\langle Arg \rangle ::= \langle IdList \rangle : \langle TypeSpecifier \rangle$   
 $\quad | \text{ var } \langle IdList \rangle : \langle TypeSpecifier \rangle$
20.  $\langle CompoundStatement \rangle ::= \text{ begin } \langle StatementList \rangle \text{ end}$
21.  $\langle StatementList \rangle ::= \langle Statement \rangle$   
 $\quad | \langle Statement \rangle ; \langle StatementList \rangle$
22.  $\langle Statement \rangle ::= \emptyset$   
 $\quad | \langle CompoundStatement \rangle$   
 $\quad | \langle AssignmentStatement \rangle$   
 $\quad | \langle ProcedureCall \rangle$   
 $\quad | \langle ForStatement \rangle$   
 $\quad | \langle WhileStatement \rangle$   
 $\quad | \langle IfStatement \rangle$   
 $\quad | \langle CaseStatement \rangle$   
 $\quad | \langle RepeatStatement \rangle$
23.  $\langle AssignmentStatement \rangle ::= \langle Variable \rangle := \langle Expression \rangle$
24.  $\langle ProcedureCall \rangle ::= \text{ id } \langle Actuals \rangle$
25.  $\langle ForStatement \rangle ::= \text{ for id } := \langle Expression \rangle \text{ to } \langle Expression \rangle \text{ do } \langle Statement \rangle$   
 $\quad | \text{ for id } := \langle Expression \rangle \text{ downto } \langle Expression \rangle \text{ do } \langle Statement \rangle$
26.  $\langle WhileStatement \rangle ::= \text{ while } \langle Expression \rangle \text{ do } \langle Statement \rangle$
27.  $\langle IfStatement \rangle ::= \text{ if } \langle Expression \rangle \text{ then } \langle Statement \rangle$   
 $\quad | \text{ if } \langle Expression \rangle \text{ then } \langle Statement \rangle \text{ else } \langle Statement \rangle$
28.  $\langle RepeatStatement \rangle ::= \text{ repeat } \langle StatementList \rangle \text{ until } \langle Expression \rangle$
29.  $\langle CaseStatement \rangle ::= \text{ case } \langle Expression \rangle \text{ of } \langle CaseList \rangle \text{ end}$
30.  $\langle CaseList \rangle ::= \langle Case \rangle$   
 $\quad | \langle Case \rangle ; \langle CaseList \rangle$
31.  $\langle Case \rangle ::= \langle ConstantList \rangle : \langle Statement \rangle$
32.  $\langle ConstantList \rangle ::= \text{ constant}$   
 $\quad | \text{ constant } , \langle ConstantList \rangle$
33.  $\langle Expression \rangle ::= \langle SimpleExpression \rangle$   
 $\quad | \langle SimpleExpression \rangle = \langle SimpleExpression \rangle$   
 $\quad | \langle SimpleExpression \rangle <> \langle SimpleExpression \rangle$   
 $\quad | \langle SimpleExpression \rangle < \langle SimpleExpression \rangle$   
 $\quad | \langle SimpleExpression \rangle <= \langle SimpleExpression \rangle$   
 $\quad | \langle SimpleExpression \rangle > \langle SimpleExpression \rangle$   
 $\quad | \langle SimpleExpression \rangle >= \langle SimpleExpression \rangle$
34.  $\langle SimpleExpression \rangle ::= \langle Term \rangle$   
 $\quad | \langle SimpleExpression \rangle + \langle Term \rangle$   
 $\quad | \langle SimpleExpression \rangle - \langle Term \rangle$   
 $\quad | \langle SimpleExpression \rangle \text{ or } \langle Term \rangle$

35.  $\langle \text{Term} \rangle ::= \langle \text{Factor} \rangle$   
     |  $\langle \text{Term} \rangle * \langle \text{Factor} \rangle$   
     |  $\langle \text{Term} \rangle / \langle \text{Factor} \rangle$   
     |  $\langle \text{Term} \rangle \text{ div } \langle \text{Factor} \rangle$   
     |  $\langle \text{Term} \rangle \text{ mod } \langle \text{Factor} \rangle$   
     |  $\langle \text{Term} \rangle \text{ and } \langle \text{Factor} \rangle$
36.  $\langle \text{Factor} \rangle ::= ( \langle \text{Expression} \rangle )$   
     |  $+ \langle \text{Factor} \rangle$   
     |  $- \langle \text{Factor} \rangle$   
     | **not**  $\langle \text{Factor} \rangle$   
     |  $\langle \text{FunctionCall} \rangle$   
     | **constant**  
     |  $\langle \text{Variable} \rangle$
37.  $\langle \text{FunctionCall} \rangle ::= \text{id } \langle \text{Actuals} \rangle$
38.  $\langle \text{Actuals} \rangle ::= \emptyset$   
     |  $( \langle \text{ExpressionList} \rangle )$
39.  $\langle \text{ExpressionList} \rangle ::= \langle \text{Expression} \rangle$   
     |  $\langle \text{Expression} \rangle , \langle \text{ExpressionList} \rangle$
40.  $\langle \text{Variable} \rangle ::= \text{id}$   
     |  $\langle \text{Variable} \rangle . \text{id}$   
     |  $\langle \text{Variable} \rangle ^$   
     |  $\langle \text{Variable} \rangle [ \langle \text{ExpressionList} \rangle ]$
41.  $\langle \text{TypeSpecifier} \rangle ::= \text{id}$   
     |  $^ \langle \text{TypeSpecifier} \rangle$   
     |  $( \langle \text{IdList} \rangle )$   
     | **constant .. constant**  
     | **array**  $[ \langle \text{DimensionList} \rangle ]$  **of**  $\langle \text{TypeSpecifier} \rangle$   
     | **record**  $\langle \text{FieldList} \rangle$  **end**  
     | **file of**  $\langle \text{TypeSpecifier} \rangle$
42.  $\langle \text{DimensionList} \rangle ::= \langle \text{Dimension} \rangle$   
     |  $\langle \text{Dimension} \rangle , \langle \text{DimensionList} \rangle$
43.  $\langle \text{Dimension} \rangle ::= \text{constant .. constant}$   
     | **id**
44.  $\langle \text{FieldList} \rangle ::= \langle \text{Field} \rangle$   
     |  $\langle \text{Field} \rangle ; \langle \text{FieldList} \rangle$
45.  $\langle \text{Field} \rangle ::= \langle \text{IdList} \rangle : \langle \text{TypeSpecifier} \rangle$
46.  $\langle \text{IdList} \rangle ::= \text{id}$   
     |  $\text{id} , \langle \text{IdList} \rangle$