

program → PROGRAM\_KW IDENTIFIER declarations\_list procedure\_list MAIN\_KW block

declarations\_list → declarations | declarations\_list declarations

declarations → type\_specifiers declarator\_list ; | “empty”


type\_specifiers → INTEGER\_KW | REAL\_KW | CHAR\_KW | BOOLEAN\_KW

declarator\_list → declarator | declarator\_list , declarator

declarator → dec | dec := initializer

dec → IDENTIFIER | IDENTIFIER [range] | IDENTIFIER [INTEGER\_CONSTANT]

range → IDENTIFIER .. IDENTIFIER | INTEGER\_CONSTANT .. INTEGER\_CONSTANT  
| arithmetic\_expressions .. arithmetic\_expressions

initializer → Initializer\_list | { initializer\_list } 

Initializer\_list → constant\_expressions , initializer\_list | constant\_expressions

procedure\_list → procedure\_list procedure | “empty”


procedure → PROCEDURE\_KW IDENTIFIER parameters { declarations\_list block } ;

parameters → ( declarations\_list )

block → { statement\_list }


statement\_list → statement ; | statement\_list statement ;

statement → IDENTIFIER := expressions |

IF\_KW bool\_expressions THEN\_KW statement  |

IF\_KW bool\_expressions THEN\_KW statement  ELSE\_KW statement  |

DO\_KW statement  WHILE\_KW bool\_expressions |

FOR\_KW IDENTIFIER IN\_KW range REPEAT\_KW statement  |

SWITCH\_KW expressions caseelement default END\_KW |

IDENTIFIER (arguments) |

RETURN\_KW expressions |

EXIT\_KW WHEN\_KW (bool\_expressions) | block | “empty”

arguments → declarator\_list | “empty”

caseelement → CASE\_KW INTEGER\_CONSTANT : block ; | caseelement CASE\_KW  
INTEGER\_CONSTANT : block ;

default → DEFAULT\_KW : block ; | empty

expressions → constant\_expressions | bool\_expressions | arithmetic\_expressions |  
IDENTIFIER | (expressions)

constant\_expressions → INTEGER\_CONSTANT | REAL\_CONSTANT |  
CHAR\_CONSTANT | BOOLEAN\_CONSTANT

bool\_expressions → expressions < expressions | expressions <= expressions |  
expressions > expressions | expressions >= expressions |  
expressions = expressions | expressions <> expressions |  
expressions AND\_KW expressions | expressions OR\_KW expressions |  
expressions AND\_KW THEN\_KW expressions |  
expressions OR\_KW ELSE\_KW expressions |  
NOT\_KW expressions

arithmetic\_expressions → expressions + expressions | expressions - expressions |  
expressions \* expressions | expressions / expressions |  
expressions % expressions | - expressions