

program : PROGRAM identifier ';' block '.'

block : { var_list } { function_definition } * function_body

var_list : VAR data_definition { data_definition } *

data_definition : identifier_list ':' INTEGER ';'

identifier_list : identifier { ',' identifier } *

function_definition : function_heading ';' block ';'

function_heading : FUNCTION identifier { parameter_list } ':' INTEGER

parameter_list : '(' identifier_list ':' INTEGER ')'

function_body : BEGIN statement_sequence END

statement_sequence : statement { ';' statement } *

statement : { identifier ':=' expression }

| BEGIN statement_sequence END

| IF expression THEN statement { ELSE statement }

| WHILE expression DO statement

| FOR identifier ':=' expression [TO | DOWN TO] expression DO statement

expression : simple_expr | simple_expr relop simple_expr

relop : '=' | '<>' | '<' | '>' | '<=' | '>='

simple_expr : operand | sign operand | simple_exp addop operand

sign : '+' | '-'

addop : '+' | '-'

operand : factor | operand multop factor

multop : '*' | '/' | DIV | MOD

factor : identifier | constant | '(' expression ')' | identifier argument_list

argument_list : '(' expression { ',' expression } * ')'

1) identifier : letter { letter | digit } *

2) constant : digit { digit } *

از این دو قانون تولید در تشخیص token ها توسط Lex. Anal. استفاده می شود.

✓ در قوانین تولید فوق بایستی برای قوانین مربوط به دستورات بررسی لازم به منظور تجربه به روس LL(1) صورت گیرد و بررسی لازم جهت نوشتن زیر برنامه های مربوط صورت گیرد. (در صورت LL(1) بودن بایستی تغییرات لازم داده شود).