Main_Program ::= Main_Program'' Main_Program' ✅

Main_Program' ::= Main_Program Main_Program' | ε ✅

Main_Program'' ::= Declaration | Assignment | City ✅

Declaration ::= **const** Declaration' | Declaration' ✅

Declaration' ::= Type **variable equals** Expression ✅

Assignment ::= **variable equals** Expression ✅

Type ::= **type_number** | **type_string** | **type_coordinate** | **type_list lthen** Type **mthen** ✅

Expression ::= Additive ✅

Additive ::= Multiplicative Additive' ✅

Additive' ::= **plus** Multiplicative Additive' | **minus** Multiplicative Additive' | ε ✅

Multiplicative ::= Exponential Multiplicative' ✅

Multiplicative' ::= **times** Exponential Multiplicative' | **divide** Exponential Multiplicative' | **integer-divide** Exponential Multiplicative' | ε ✅

Exponential ::= Unary Exponential' ✅

Exponential' ::= **pow** Unary Exponential' | ε ✅

Unary ::= **plus** Primary | **minus** Primary | Primary ✅

Primary ::= **real** | **variable** | **lbracket** Additive **rbracket** | **lsq_bracket** Inner_List **rsq_bracket** ✅

Inner_List ::= Inner_List'' Inner_List' ✅

Inner_List' ::= **comma** Inner_List'' Inner_List | ε ✅

Inner_List'' ::= Expression ✅

City ::= **city variable block_start** City' **block_end** ✅

City' ::= City''' City'' ✅

City'' ::= City' City'' | ε ✅

City''' ::= City_Constructs | Declaration | Assignment | Print ✅

Print ::= **print lbracket** Expression **rbracket** ✅

City_Constructs ::= Restaurant | Road ✅

Restaurant ::= Name Shape Marker Routes ✅

Name ::= **name colon** String ✅

String ::= **variable | string** ✅

Shape ::= **shape colon block_start** Lines **block_end** ✅

Lines ::= Line Line Line Line' ✅

Line' ::= Line Line' | ε ✅

Line ::= **line lsq_bracket** Coord **comma** Coord **rsq_bracket** ✅

Coord ::= **variable** | **lbracket** Expression **comma** Expression **rbracket** ✅

Marker ::= Marker' | ε ✅

Marker' ::= **marker colon** Point ✅

Point ::= **point lsq_bracket** Coord **rsq_bracket** ✅

Routes ::= Routes' | ε ✅

Routes' ::= **routes colon lsq_bracket** Roads **rsq_bracket** ✅

Roads ::= Road Road' ✅

Road' ::= **comma** Road Roads' | ε ✅

Road ::= **road variable block_start** Name Road_Shapes **block_end** ✅

Road_Shapes ::= **shape colon block_start** Road_Shapes' Road_Shapes **block_end** ✅

Road_Shapes' ::= Road_Shape Road_Shapes' | ε ✅

Road_Shape ::= Line | Bend ✅

Bend ::= **bend lsq_bracket** Coord **comma** Coord **comma** Expression **rsq_bracket** ✅

For ::= **foreach variable in** Variable **block_start** Program **block_end** ✅

Variable ::= **variable** | Radius ✅

Radius ::= **lsq_bracket** Coord **comma** Expression **rsq_bracket** ✅

Program ::= Program'' Program' ✅

Program '::= Program Program' | ε ✅

Program ''::= Declaration | Assignment | Print | Highlight ✅

Highlight ::= **highlight lbracket variable rbracket** ✅ pazi da je variable coord!

Neterminali

const -> const
variable -> {A,…,Z,a,…,z}{A,…,Z,a,…,z,0,…,9}*
equals -> =
type_number -> num
type_string -> string
type_coordinate -> coord
type_list -> List
lthen -> <
mthen -> >
city -> city
block_start -> {
block_end -> }
name -> name
colon -> :
string -> “{A,…,Z,a,…,z,0,…,9}*” oz. ASCII
shape -> shape
line -> line
lsq_bracket -> [
comma -> ,
rsq_bracket ->]
lbracket -> (
rbracket ->)
marker -> marker
point-> point
routes-> routes
road-> road
bend-> bend
foreach-> foreach
highlight -> highlight