Liss - A Toy Language

Language of **I**ntegers, **S**equences and **S**ets

1 Grammar

```
/************************ Program

Liss                 --> "program" IDENTIFIER Body

Body                 --> '{'
                           "declarations" Declarations
                           "statements"   Statements
                         '}'

/************************ Declarations

Declarations         --> Declaration
                       | Declarations Declaration


Declaration          --> Variable_Declaration
                       | SubProgram_Definition

/************************ Variables

Variable_Declaration --> Vars "->" Type ';'

Vars                 --> Var
                       | Vars "," Var

Var                  --> IDENTIFIER Value_Var

Value_Var            -->
                        | "=" Inic_Var


Type                 --> "integer"
                       | "boolean"
                       | "set"
                       | "sequence"
                       | "array" "size" Dimension

Dimension            --> NUMBER
                       | Dimension "," NUMBER
```

```
Inic_Var             --> Constant
                       | Array_Definition
                       | Set_Definition
                       | Sequence_Definition

Constant             --> Sign NUMBER
                       | "true"
                       | "false"

Sign                 -->
                       | "+"
                       | "-"



/*********************** Array_Definition


Array_Definition     --> '[' Array_Initialization ']'

Array_Initialization --> '[' Elem ']'
                       | Array_Initialization ',' Elem

Elem                 --> NUMBER
                       | Array_Definition



/*********************** Sequence_Definition


Sequence_Definition     --> "<<" Sequence_Initialization ">>"

Sequence_Initialization -->
                          | Values


/*********************** Set Definition

Set_Definition          --> '{'
                              Set_Initialization
                            '}'


Set_Initialization      -->
                          | IDENTIFIER '|' Expression


/*********************** SubProgram_Definition

SubProgram_Definition  --> "subProgram" IDENTIFIER
                           '(' FormalArgs ')' Return_Type FBody

FBody                  --> '{'
```

```
                              "declarations" Declarations
                              "statements" Statements
                              Return
                          '}'

/********* FormalArgs

FormalArgs              -->
                          | FArgs

FArgs                   --> Formal_Arg
                          | FArgs ';' Formal_Arg

Formal_Arg              --> IDENTIFIER "->" Type


/******** Return Type

Return_Type             -->
                          | "->" Type


/******** Return

Return                 -->
                         | "return" Expression


/************************ Statements

Statements            --> Statement
                        | Statements Statement


Statement             --> Assignment
                        | Write_Statement
                        | Read_Statement
                        | Conditional_Statement
                        | Iterative_Statement
                        | Function_Call
                        | SucOrPrecc
                        | CopyStatement


/************************ Assignment

Assignment            --> Designator "=" Expression


/******** Designator

Designator            --> IDENTIFIER Array_Acess

Array_Acess           -->
```

```
                      | '[' Elem ']'

Elem                --> Expression
                      | Elem ',' Expression


/************************ Function Call

Function_Call       --> IDENTIFIER '(' SubPrg_Args ')'

SubPrg_Args         -->
                      | Args

Args                --> Expression
                      | Args ',' Expression


/*********************** Expression

Expression          --> Single_Expression
                      | Expression Rel_Oper Single_Expression

/******** Single_Expression

Single_Expression   --> Term
                      | Single_Expression Add_Op Term

/******** Term

Term                --> Factor
                      | Term Mul_Op Factor


/******** Factor

Factor              --> Constant
                      | Designator
                      | Function_Call
                      | '!' One_Factor
                      | Tail
                      | Head
                      | Cons
                      | Member
                      | IsEmpty
                      | Length
                      | Delete


/******** Add_Op, Mul_Op, Rel_Op

Add_Op              --> "+"
                      | "-"
                      | "||"
                      | "++"
```

```
Mul_Op              --> "*"
                      | "/"
                      | "&&"
                      | "**"

Rel_Op              --> "=="
                      | "!="
                      | "<"
                      | ">"
                      | "<="
                      | ">="
                      | "in"


/********************* Write_Statement

Write_Statement    --> Write_Expr '(' Print_What ')'

Write_Expr         --> "write"
                     | "writeln"

Print_What         -->
                     | Expression


/********************* Read_Statement

Read_Statement     --> "input" IDENTIFIER



/********************* Conditional & Iterative

Conditional_Statement --> IfThenElse_Stat

Iterative_Statement   --> For_Stat
                        | While_Stat


/******** IfThenElse_Stat

IfThenElse_Stat       --> "if" Expression
                          "then" '{' Statements '}'
                          Else_Expression

Else_Expression       -->
                        | "else" '{' Statements '}'


/******** For_Stat

For_Stat             --> "for" '(' Interval ')' Step Satisfy
                         '{' Statements '}'
```

```
Interval             --> IDENTIFIER Type_Interval

Type_Interval        --> "in"         Range
                       | "inArray"    IDENTIFIER
                       | "inFunction" IDENTIFIER

Range                --> Minimum ".." Maximum

Minimum              --> NUMBER
                       | IDENTIFIER

Maximum              --> NUMBER
                       | IDENTIFIER

Step                 -->
                       | UpDown NUMBER

UpDown               --> "stepUp"
                       | "stepDown"

Satisfy              -->
                       | "satisfying" Expression  -- boolean expression


/******** While_Stat

While_Stat           --> "while" '(' Expression ')'
                         '{' Statements '}'


/********************** SuccOrPredd

SuccOrPred          --> SuccPred IDENTIFIER

SuccPred            --> "succ"
                      | "pred"


/********************** SequenceOper

Tail               --> "tail" '(' Expression ')'

Head               --> "head" '(' Expression ')'

Cons               --> "cons" '(' Expression ',' IDENTIFIER ')'

Delete             --> "del"  '(' Expression ',' IDENTIFIER ')'

Copy               --> "copy" '(' IDENTIFIER ',' IDENTIFIER ')'

IsEmpty            --> "isEmpty" '(' Expression ')'

Length             --> "length" '(' Expression ')'
```

```
/********************* SetOper

Member              --> "isMember" '(' Expression, ',' IDENTIFIER ')'
```