**Tiny Language CFG**

# Basic Terminals

Letter -> \_ | [a-z] | [A-Z]

Digit -> [0-9]

Boolean\_Operator -> && | \|\|

Stats\_set -> Assign\_Stat | Write\_Stat | Read\_Stat | Return\_Stat | Declaration\_Stat | If\_Stat | ElseIF\_Stat | Else\_Stat

# Terminals\NonTerminals

1. Constant

Constant -> ConstantDash. ConstantDash | ConstantDash

ConstantDash -> Digit ConstantDash | Digit

1. String:

String -> “String’”

String’ -> String’’ String’ | String’’

String’’ -> Letter | Digit | A\_Operator | Ɛ

1. Comment:
   1. Comment -> \*Comment’\*
   2. Comment’ -> String
2. Function Call:
   1. FunctionCall -> id FunctionCallDash;
   2. FunctionCallDash -> (Params)
   3. Id\_Arg ->, Id
3. Term:

Term -> Constant | Id | FunctionCall

1. Arithmetic Operator:

AritmeticOperator -> +|-|/|\*

1. Equation:

Equation-> Term | Term ArithmeticOperator Equation | (Equation) EquationDash

EquationDash -> ArithmeticOperator Equation

1. Expression:
   1. Expression -> String | Term | Equation
2. Assignment Statement:
   1. AssignmentStatement -> Id := Expression ;
3. Datatypes:
   1. Type -> int | float | string
4. Declaration Statement:

VariableDecls -> Type VariableDecl VariableDeclsDash

VariableDecl -> Id Definition;

VariableDeclsDash -> , VariableDecl VariableDeclsDash | ;

Definition-> := Exp | eps

1. Write Statement:

WriteStatement -> write WriteDash;

WriteDash -> Expression | endl

1. Read Statement:

ReadStatement -> read Id ;

1. Return Statement:

Return -> return Expression;

1. Condition Operator:

ConditionOperator -> < | > | = | <>

1. Condition:

Condition -> Id ConditionOperator Term

1. Statement:

Statement -> Block | VariableDecls | AssignmentStatement | Return| IfStatement | WriteStatement | ReadStatement

1. -> Block | VariableDecl | AssignmentStatement | ReturnStatement | IFStatment| WriteStatement | ReadStatement
2. Condition Statement:

ConditionStatement -> Condition ConditionStatementDash

ConditionStatementDash -> BooleanOperator Condition ConditionStatement | Ɛ

1. If Statement:
   1. IfStatement -> if ConditionStatement then StatementsDash ElseIFStatementDash ElseStatementsDash end
   2. StatementDash -> Statments StatmentsDash | Ɛ
   3. ElseIFStatementDash -> ElseifStatement ElseIFStaementDash | Ɛ
   4. ElseStatementDash -> ElseStatement | Ɛ
2. Elseif Statement:

ElseifStatement -> elseif Condition then StatementsDash ElseIFStatementDash ElseStatementDash end

1. Else Statement:

ElseStatement -> else StatementsDash end

1. Repeat Statement:

RepeatStatement -> repeat StatementsDash until ConditionStatement

1. Function Name:

FunctionName -> Id

1. Function Declaration:

Function Declaration -> Type FunctionName (Parameters)

26- Parameters:

Parameters -> Parameter ParameterDash | Ɛ

ParameterDash -> ',' Parameter ParameterDash | Ɛ

1. Function Body:

Block -> {StatementsDash Return}

1. Function Statement:

FunctionStatement -> FunctionDeclation Block

1. Main Function:

Main -> Type main () Block

1. Program:

Program -> FunctionStatementDash Main

FunctionStatementDash -> FunctionStatement FunctionStatementDash | Ɛ