Tiny Language CFG & Parser implementation

Compiler's Theory

MS2

MS2(CFG&Parser Implementaion(

Team : G3_56

TEAM MEMBERS:

NAME	ID	SECTION
كريم السيد عبدالقوي	20201700600	6
محمد هشام زين العابدين	20201700754	8
عبدالله مصطفي عبدالسلام	20201700493	5

TINY LANGUAGE

Terminals & Non Terminals

- **Terminals**

Number	String
Identifier	Comment_Statement
Reserved_Words	

- **Non-Terminals**

FunctionCall	Argument	Functions
ArgumentList	Arguments	Program
MainFunction	If_Statement	Function
Else_If_Statement	Else_Statement	FunctionBody
Declaration_Statement	Declaration	FunctionDecl
Declarations	Datatype	FunctionName
ConditionOp	PlusOrMinus	ParameterList
MultOrDivide	BooleanOp	Parameter
Statement	ReturnStatement	Parameters
Statements	Declaration_Statement	RepeatStatement
Assignment_Statement	Read_Statement	Write_Statement
ConditionStatement		

CFG for Tiny Language

- *Program* -> FunctionBlock MainBlock
- *FunctionBLock* -> Function FunctionBlock | ε
- *MainBlock* -> Datatype main() FunctionBody
- *Function* -> FunctionDecl FunctionBody
- *FunctionDecl* -> Datatype identifier ParameterList
- *ParameterList* -> (Parameter)
- *Parameter* -> Datatype identifier Parameters | ε
- *Parameters* -> ,Datatype identifier Parameters| ε
- *FunctionBody* -> { Statements ReturnStatement }
- *Statements* -> Statement Statements | ε
- *Statement* -> Read_Statement| Write_Statement| Assignment_Statement | If_Statement| FunctionCall | RepeatStatement| Declaration_Statement| ε
- *ReturnStatement* -> return Expression ; | ε
- *RepeatStatement* -> repeat Statements until Condition
- *Assignment_Statement* -> identifier := Expression;
- *Read_Statement* -> read identifier ;
- *Write_Statement* -> write Expression | *endl* ;
- *Condition* -> Term ConditionOp Term
- *Expression* -> string | Term | Equation
- *Term* -> number | identifier | FunctionCall
- *Equation* -> TermAndOp | (Equation) SubEquation
- *TermAndOp* -> Term SubEquation
- *SubEquation* -> PlusOrMinus Equation | MultOrDivide Equation
- *FunctionCall* -> identifier ArgumentList;
- *ArgumentList* -> (Argument) | ()
- *Argument* -> Expression Arguments | ε
- *Arguments*-> , Argument | ε
- *If_Statement* -> if ConditionStatement then Statements *ReturnStatement* Else_If_Statement | Else_Statement

- *Else_If_Statement*->elseif ConditionStatement thenStatementsWithRet Else_If_Statement | Else_Statement
- *Else_Statement* -> else Statements *ReturnStatement*
- *Declaration_Statement* -> Datatype Declaration ;
- *Declaration* -> identifier Declarations | Assignment_Statement Declarations
- *Declarations*->,identifier Declarations | ,Assignment_Statement Declarations|ε
- *Datatype* -> integer | float | string
- *ConditionOp* -> = | > | < | <>
- *PlusOrMinus* -> + | -
- *MultOrDivide* -> * | /
- *BooleanOp* -> && | ||