COMPILER THEORY

Tiny Compiler || MS2

Document containing Two Sections:

- Section One : Terminals
- Section Two : CFG

Ain Shams University
FCIS

Terminals

Symbols and Operators

- { } (function body delimiters)
- ( ) (for grouping and function parameters)
- ; (statement terminator)
- , (separator)
- := (assignment operator)
- +, - (addition operators)
- *, / (multiplication operators)
- <, >, =, <> (relational operators)
- ss, || (Boolean operators)

Keywords

- int, float, string (datatypes)
- main (for main function)
- if, then, else, elseif, end (conditional statements)
- repeat, until (loop statements)
- read, write (I/O statements)
- return (function return)
- endl (end line)

Special Sequences

- " (string delimiter)

Tokens

- identifier (user-defined names)
- number (numeric literals)
- letter (for identifiers)
- digit (for numbers)
- symbol (for comments)

CFG

Program Structure:

Program → Functions Main_Function

Functions → Function_Statement Functions | ε

Main_Function → int main () Function_Body

Function Declarations:

Function_Statement → Function_Declaration Function_Body

Function_Declaration → Datatype FunctionName (Parameters)

Parameters → Parameter, Parameters | Parameter | ε

Parameter → Datatype identifier

FunctionName → identifier

Function Body:

Function_Body → {Statements Return_Statement }

Statements:

Statements → Statement Statements | ε

Statement → Declaration_Statement

| Assignment_Statement

| Write_Statement

| Read_Statement

| If_Statement

| Repeat_Statement

| Function_Call ;

Characters → Character Characters | ε

Character → letter | digit | symbol

Declaration Statements:

Declaration_Statement → Datatype DeclarationList ;

DeclarationList → DeclItem DeclarationList_Tail

DeclarationList_Tail → , DeclarationList | ε

DeclItem → identifier DeclItem_Tail

DeclItem_Tail → := Expression | ε

Datatype → int | float | string

Assignment Statements:

Assignment_Statement → identifier := Expression ;

Input/Output Statements:

Write_Statement → write Write_Tail

Write_Tail → Expression ; | endl ;

Read_Statement → read identifier ;

Conditional Statements:

If_Statement → if Condition_Statement then Statements Else_Statement end

Else_Statement → elseif Condition_Statement then Statements Else_Statement | else Statements | ε

Loops:

Repeat_Statement → repeat Statements until Condition_Statement

Function Calls

Function_Call → identifier (ArgumentList)

ArgumentList→ ExpressionList| ε

ExpressionList → Expression ExpressionList_Tail

ExpressionList_Tail → , ExpressionList | ε

Expressions:

Expression → String | Equation | Term

String → " Characters "

Equation → Term Equation_Tail

Equation_Tail → AddOp Equation | ε

Term → Factor Term_Tail

Term_Tail → MultOp Term | ε

Factor → number | identifier | Function_Call | (Equation)

Conditions:

Condition_Statement → Condition BooleanClause

BooleanClause → Boolean_Operator Condition_Statement | ε

Condition → Expression RelOp Expression

Operators:

AddOp → + | -

MultOp → * | /

RelOp → < | > | = | <>

Boolean_Operator → && | ||

Return Statement:

Return_Statement → return Expression ;