Faculty of Computer & Information Sciences
Ain Shams University
Subject: CSC360
Compilers' Theory

Examiners: Prof. Zaki Taha
Dr. Sally Saad
Academic year: 1st term 2021-2022
Year: 3ed undergraduate

Compilers' Theory
Milestone -1-

1. The "TINY" Language Regular Expressions:

1) Number:

Digit := [0-9]

Num_Un_Signed := (Digit)+

Num_Signed := (+|-)? Num_Un_Signed

Num_float := Num_Signed (\.Num_Un_Signed)?

2) String:

Letter = [a-z]|[A|Z]

Str := ^\".*\"$

3) Reserved_Keywords:

R_Keywords := int|float|string|read|write|repeat|until|if|elseif|else|then|return|end

4) Comment_Statement:

L_Comment := ^/*.**/$

5) Identifiers:

identifier := Letter(Letter|Digit)*

6) Function_Call:

Fun_call := identifier \(((identifier)(, identifier)*)? \)

7) Term:

Term := (Num_float | identifier | Fun_call)

8) Arithmatic_Operator:

Arth_op = (+ | - | * | /)

9) Equation:

E_unit = (Term+ Arth_op)*(Term+)$

Equ = E_unit | (Term Arth_op)* \(E_unit \)(Arth_op Term)*

10) Expression:

Exp := Term|Str|Equ

11) Assignment_Statement:

Ass_st:= (identifier := Exp)

12) Datatype:

Datatype := (int|float|string)

13) Declaration_Statement:

Dec_st := ^Datatype identifier (,identifier|,Ass_st)*;$

14) Write_Statement:

Write_st:= ^ write (EXp|\n) ;$

15) Read_Statement:

Read_st:= ^ read identifier ;$

16) Return_Statement:

Return_st:= ^ return Exp ;$

17) Condition_Operator:

Con_op:= (<|>|=|<>)

18) Condition:

Con:= (identifier Con_op term)

19) Boolean_Operator:

Boolean_Op:= (&& | ||)

20) Condition_Statement:

Condition (Boolean_Operator Condition)*

21) Set_of_Statements

Set_of_Statements := (Assignment_Statement | Declaration_Statement | Write_Statement | Read_Statement | (Return_Statement)? | Function_Call)

22) If_Statement:

If_Statement := “if” Condition_Statement “then” Set_of_Statements (Else_If_Statement | Else_Statement | end)

23) Else_If_Statement:

Else_If_Statement := “elseif” Condition_Statement then Set_of_Statements (Else_If_Statement | Else_Statement | “end”)

24) Else_Statement:

Else_Statement := ”elseif” Condition_Statement “then” Set_of_Statements (Else_If_Statement | Else_Statement | “end”)

25) Repeat_Statement:

Repeat_Statement := “repeat” Set_of_Statements “until” Condition_Statement

26) FunctionName:

FunctionName := Identifier

27) Parameter:

Parameter := Datatype Identifier

28) Function_Declaration:

Function_Declaration :=

Datatype FunctionName \((Parameter(,Parameter)*)? \)

29) Function_Body:

Function_Body := { Set_of_Statements (Return_Statement) }

30) Function_Statement:

Function_Statement:= Function_Declaration Function_Body

31) Main_Function:

Main_Function := Datatype “main“ \(\ ) Function_Body

32) Program:

Program := (Function_Statment)* Main_Function