

ASSIGNMENT DOCUMENTATION

COMPILER CONSTRUCTION

Hafiz Ahmad Hassan 15L-4359

Maryam Nadeem 15L-4107

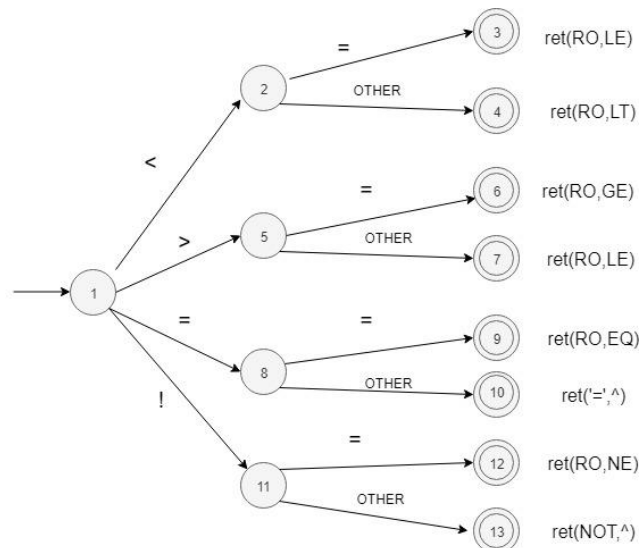
LEXICAL ANALYZER

LANGUAGE TOKENS

- **Data types:** there are two data types recognized by the lexical analyzer int and char
 1. (int , kw)
 2. (char , kw)
- **Keywords:** there are six keywords recognized by the lexical analyzer and those are if, else, while, ret, in, and out
 1. (if , kw)
 2. (else , kw)
 3. (while , kw)
 4. (ret , kw)
 5. (in , kw)
 6. (out , kw)
- **Arithmetic operators:** there are four arithmetic operators recognized by the lexical analyzer + , - , * , /
 1. ('+' , ^)
 2. ('-' , ^)
 3. ('*' , ^)
 4. ('/' , ^)
- **Relational operators:** there are four relational operators in the lexical analyzer < , <= , > , >= , == , !=
 1. (RO , <)
 2. (RO , <=)
 3. (RO , >)
 4. (RO , >=)
 5. (RO , ==)
 6. (RO , !=)

Regular definition: < | <= | > | >= | == | !=

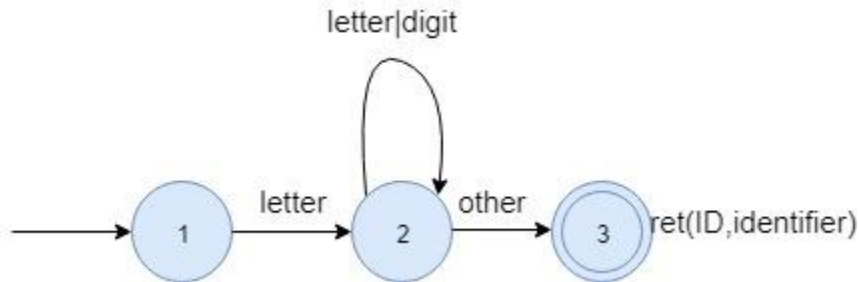
Transition Diagram:



- **Line comments:** ('# ' , ^)
- **Multi line comments:** (' ## ' , ^)
- **Identifier:** (ID, identifier)

Regular definition: letter: a|b|...|z|A|B|...|Z| digit: 0|1|2|...|9 identifier: letter (letter|digit)*

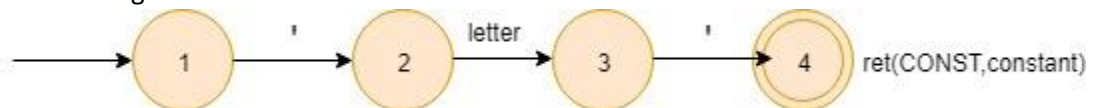
Transition Diagram:



- **Numeric constant:** (NUM, numeric constant)
- **Constants:** (CONST, constant)

Regular definition: letter: a|b|...|z|A|B|...|Z| Constant: 'letter'

Transition Diagram:



- **Strings:** (STR, string)
- **Parenthesis:** (' (' , ^) (') ' , ^)
- **Braces:** (' (' , ^) (' (' , ^)
- **Square brackets:** (' [' , ^) ('] ' , ^)
- **Assignment operator:** (' := ' , ^)
- **Semi colon:** (' ; ' , ^)
- **Comma:** (' , ' , ^)

PARSER

Declaration -> Function Declaration

Function -> FunctType id (Parameter) { Statements }

FunctType -> Type | ^

Parameter -> One More | ^

One -> Type id

More -> , Type id More | ^

DeclareVariable -> Type VarId ;

VarId -> id MoreVar

MoreVar-> , id MoreVar | ^

InitializeVariable -> id AssignOP IniVar ;

IniVar -> id | Integer | LiteralConstant | E

OUT -> out OutType ;

OutType -> String | id | integer | LiteralConstant

RET -> ret RetType ;

RetType -> id | integer | LiteralConstant

IN -> in id ;

E -> T R

R -> + T R | - T R

T -> F R'

R' -> * F R' | / F R'

F -> id | integer | (E)

WHILE -> while (Con) ConStatement

IF -> if (Con) ConStatement

Con -> P R O P

P -> id | integer | LiteralConstant | (Con)

ConStatement -> S | ManyStatements

S -> DeclareVariable | Initialize Variable | E | OUT | RET | IN

ManyStatements -> { Statements }

Statements -> Statement Statements

Statement -> DeclareVariable | Initialize Variable | E | OUT | RET | IN | WHILE | IF

Type -> int | char