Asjad Iftikhar → 18L-0951

Tayyab Waseem → 18L-1017

Compiler Construction Fall' 2021

Assignment #1

Phase I: "Lexical Analyzer Description"

Task 2.1 a) Description

1. a, b) Data Types

Token.name	Lexeme	Description
INT	int	Keyword
CHAR	char	Keyword

2. a, b) Control Structures

IF	if	Keyword
ELIF	elif	Keyword
ELSE	else	Keyword
WHILE	while	Keyword

2. c, d) I/O

INPUT	input	Keyword
ARROW	→	Operator
PRINT	print	Keyword
PRINTLN	println	Keyword

3. Arithmetic and compound Operators

ADD	+	Operator
SUB	-	Operator
MUL	*	Operator
DIV	/	Operator
INCR	++	Operator

4. Relation Operators

RO	<, <=, >, >=, ==, ~=	Operator
ASSIGN	=, :	Operator

5. Comments

COMMENT	\\	Operator
MCOMMENT	*, */	Operator

6. Identifiers

ID	RE	Identifier

7. Literals

NUM	RE	Literal
CHAR	RE	Literal
STRING	RE	Literal

*REs given in next section of this document.

8. Separators

OCB	{	Separator
CCB	}	Separator
OSB	[	Separator
OCB	]	Separator
OB	(	Separator
CB	)	Separator
SEMICOLON	;	Separator/Punctuation
COMMA	,	Separator/Punctuation

Task 2.1 b) Regular Expressions

1. Data Types

L = (int | char)

2. Control Structures & I/O

L = i(f | nput) | el(if | se) | print(ln | ^) | while | →

3. Arithmetic and compound Operators

L = + (+ | ^) | - | * | /

4. Relation Operators

L = (< | > | =) (= | ^) | ~= | :

5. Comments

L = (\\ (C)* /n) | (/* (C)* */)
C = any characters

6. Identifiers

ID = letter(letter | digit | _)*
letter = a | b| …… | z | A | B | … | Z
digit = 0 | 1 |……. | 9

7. Literals

NUM = digit(digit)*
digit = 0 | 1 | ……. | 9

CHAR = ‘C | S’
C = a | b | …. | z
S = A | B | …. | Z

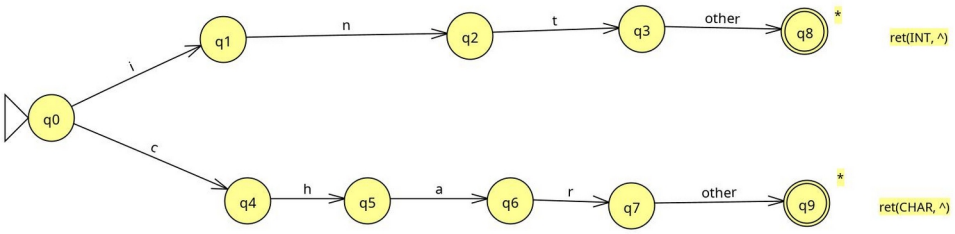
STRING = “(A)*”
A = characters | white spaces

8. Separators

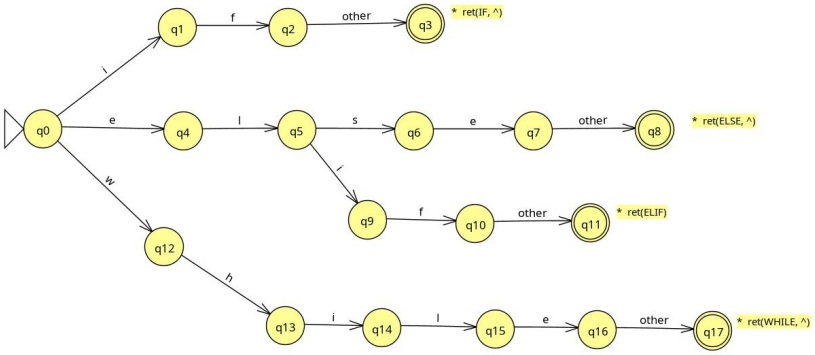
L = (|) | { | } | [|] | ; | ,

Task 2.1 C) DFAs

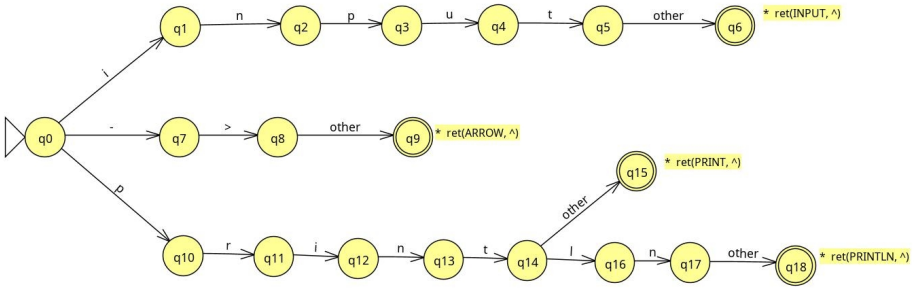
Data Types:



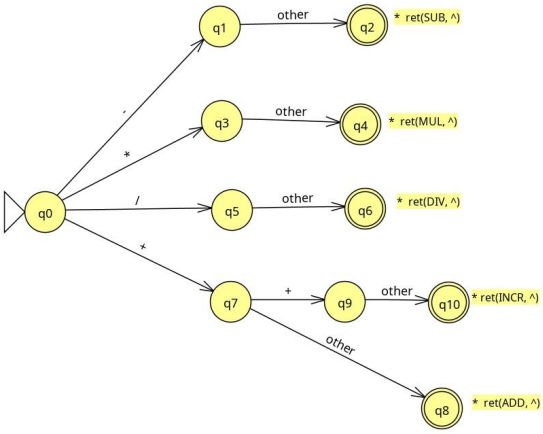
Control Statements:



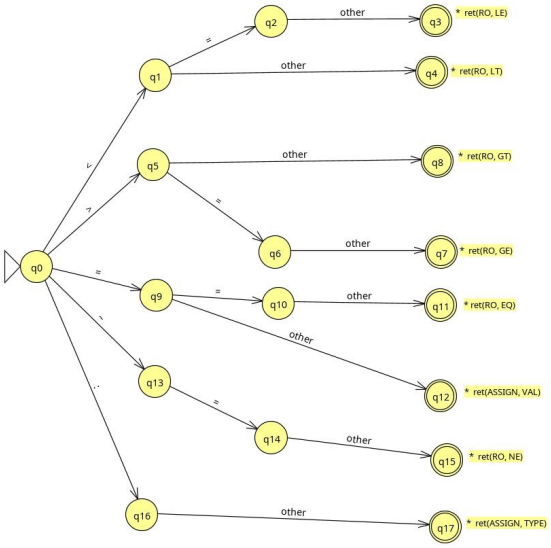
I/O:



Arithematic Operators:



Relation Operators:


q0
q1
q2
q3
* ret(RO, LE)
q4
* ret(RO, LT)
q5
q6
q7
* ret(RO, GE)
q8
* ret(RO, GT)
q9
q10
q11
* ret(RO, EQ)
q12
* ret(ASSIGN, VAL)
q13
q14
q15
* ret(RO, NE)
q16
q17
* ret(ASSIGN, TYPE)
other
=
<
>

Assignment #3 Design CFGs of the Language

* NUM, CHAR, STRING and RO are defined in previous section of this document

Task 3: CFGs

GLOBAL:

SUBSCRIPT → id [NUM | id]

LITERAL → NUM | STRING | CHAR

1. Data Types (Variable declaration and Initialization)

VDI → I | IA | C | CA

//integer

I → int : id IOP;

IOP → = NUM IC | SUBSCRIPT IC | ^

IC → , id IOP | ^

//char

C → char : id COP;

COP → = CHAR CC | SUBSCRIPT CC | ^

CC → , id COP | ^

//integer array

IA → int [] : id IAOP;

IAOP → = { AN } IAC | ^

AN → NUM , AN | NUM

IAC → , id IAOP | ^

//char array

CA → char [] : id CAOP;

CAOP → = { AC } CAC | ^

AC → CHAR , AC | CHAR

CAC → , id CAOP | ^

2. I/O Statements

PS → P (POP) | INP ;

P → print | println

POP → id | LITERAL | SUBSCRIPT

INP → input → id

3. Control Structures Statement

CS → CK EXP: { ST }

CK → while | if

EXP → EV RO EV

EV → LITERAL | id | SUBSCRIPT

ST → VDI ST | PS ST | AO ST | AS ST | ^

3. Arithmetic and compound Statements

AO → T E’

E’ → + T E’ | - T E’ | ^

T → F T’

T’ → * F T’ | / F T’ | ^

F → Final P’

P’ → ++ Final P’ | ^

Final → id | NUM | SUBSCRIPT | (AO)

4. Assignment Statement

AS → id = LITERAL ; | AO ;