Compiler Design

- 18CS41604

Name: Mohnish Devaraj
Reg No: 39110636
Section: C1

Assignment-1

PART-A

① Lexical Analysis

② Analysis phase

③ c) Retarget code

④ b) Token generator

⑤ b) It works on any computers with similar process and OS

PART-B

① Interpreter

→ Interpreter translates just one statement of the program at a time into machine code.

→ An interpreter takes very less time to analyze the source code. However, the overall time to execute the process is much slower.

→ An interpreter does not generate an intermediary code

Hence, an interpreter is highly efficient in terms of its memory.

Compiler

→ Compiler scans the entire program and translates the whole of it into machine code at once.

→ A Compiler takes a lot of time to analyze the source code. However, the overall time taken to execute the process is much faster.

→ A Compiler always generates an intermediary object code. It will need further linking. Hence more memory is needed.

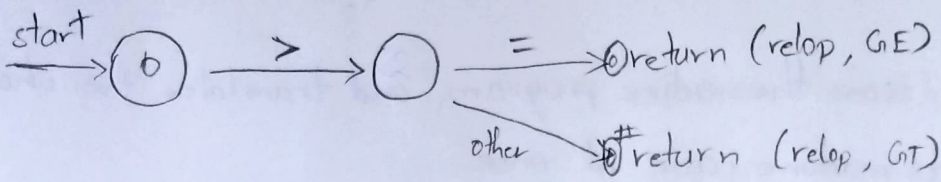
② There are 4 error recovery modes in Lexical Phase of the compiler

→ Deleting an extraneous character
→ Inserting a missing character
→ Replacing an incorrect character by a correct character
→ Transposing two adjacent characters.

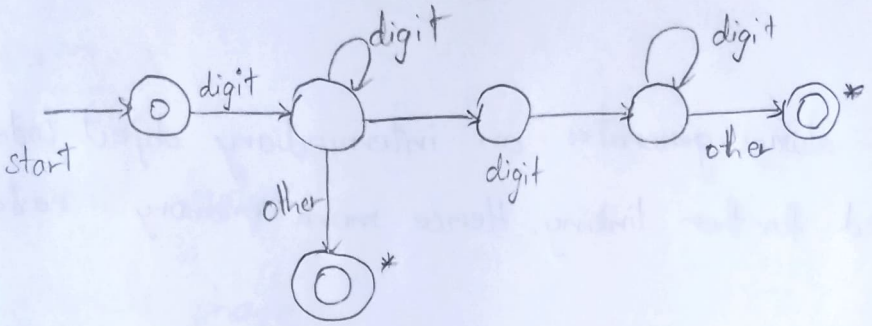
③

Lexeme	Tokens
int	keyword
radius	Identifier
=	operator
5	keyword
;	Operator
float	keyword
area	identifier
if	keyword
>	Operator
0	keyword
{	Operator
3.14	keyword
*	operator

④ a) relop >=



b) floating point number - (eg: 3.14)



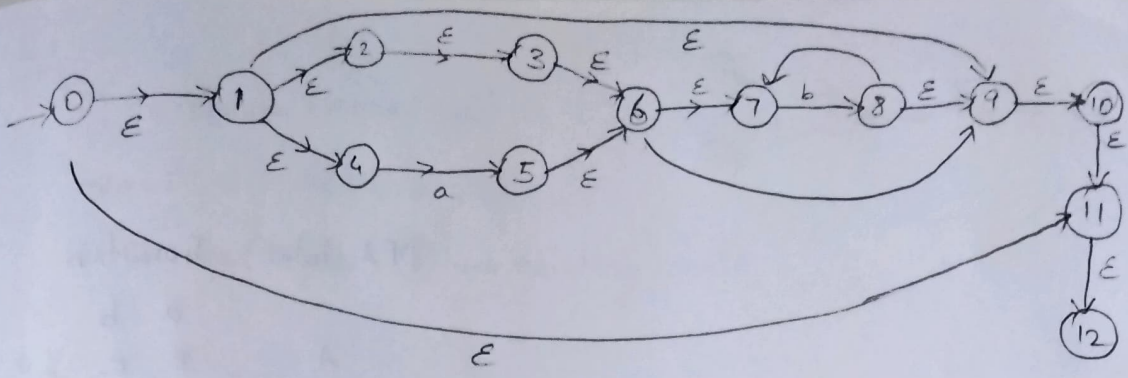
⑤ Input Buffering

A two-buffer input scheme that is useful when look ahead on the input is necessary to identify tokens is discussed. Later, other techniques for speeding up the LA, such as the use of "sentinels" to mark the buffer end is also discussed.

PART-C

① ② Given expression

$$((\varepsilon|a)b^*)^*$$



step 2:-

ε-closure (start state)

ε-closure (0) = {0, 1, 10, 2, 4, 3, 6, 7, 9}

= {0, 1, 2, 3, 4, 6, 7, 9, 10}

→ (A) state

2.1:-

ε-closure (move (A, a))

move (A, a) = δ(A, a)

= {5}

ε-closure (δ(A, a)) = ε-closure {5}

= {5, 6, 7, 9, 10, 1, 2, 4, 3, 6}

= {1, 2, 3, 4, 5, 6, 7, 9, 10}

→ (B) new state

2.2:-

ε-closure (move (A, b))

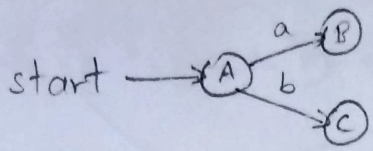
move (A, b) = δ(A, b)

= {8}

ε-closure (δ(A, b)) = ε-closure {8}

= {8, 9, 1, 10, 2, 4, 3, 6, 7}

= {1, 2, 3, 4, 6, 7, 8, 9, 10} → (C) new state



DFA states	Transition	
	a	b
A	B	C
B		
C		

step 3:

$B = \{1, 2, 3, 4, 5, 6, 7, 9, 10\}$

3.1 :-

ε-closure (move (B, a))

move (B, a) = $\delta(B, a) = \{5\}$

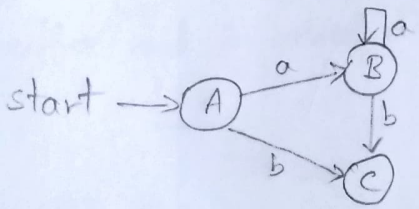
ε-closure ($\delta(B, a)$) = (B) → existing state

3.2 :-

ε-closure (move (B, b))

move (B, b) = $\delta(B, b) = \{8\}$

ε-closure ($\delta(B, b)$) = (C) → existing state



DFA states	Transition	
	a	b
A	B	C
B	B	C
C		

step 4:

$C = \{1, 2, 3, 4, 6, 7, 8, 9, 10\}$

4.1:-

ε-closure (move (C, a))

move (C, a) = δ(C, a) = {5}

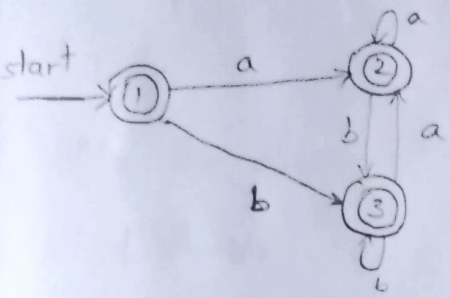
ε-closure (δ(C, a)) = (B) → existing state

4.2:

ε-closure (move (C, b))

move (C, b) = δ(C, b) = {8}

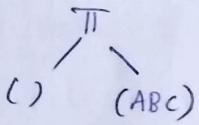
ε-closure (δ(C, b)) = (C) → existing state



DFA states	Transition Table a	b
A	B	C
B	B	C
C	B	C

DFA Subset of NFA :-

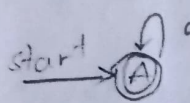
DFA states	Transition a	b
A	B	C
B	B	C
C	B	C



Since for each state the transition of inputs a, b are same so

A = B = C

The final minimized DFA is



DFA states	Transition a	b
A	A	A