

1

Name: Preeti Rautela
 ID : 53831
 Sub : Compiler Design
 Code : TIT31
 Date : 29/6/2021

(Signature)

Ques 1

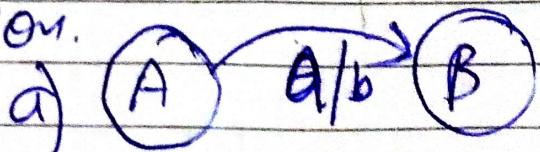
Ans - The procedure of converting Regular exp to NFA is as follows:

Regular EXP \rightarrow E-NFA \rightarrow NFA

(I) Regular EXP \rightarrow E-NFA

Rules:

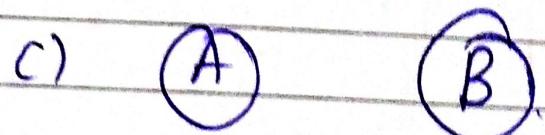
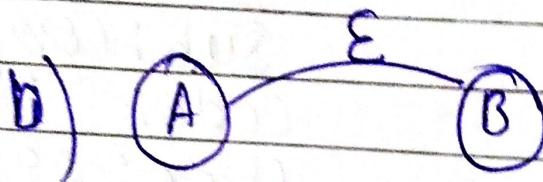
- If operand is a character c then our FA has two states, a start and end state, & a transition.
- If the operand is epsilon then our FA has two states start & end & an epsilon transition.
- If operand is null. then two states only no transition.



(2)

ANMOL
P. No.
Dt.

Name: Preeti Routela
ID : 53831



ϵ -NFA to NFA

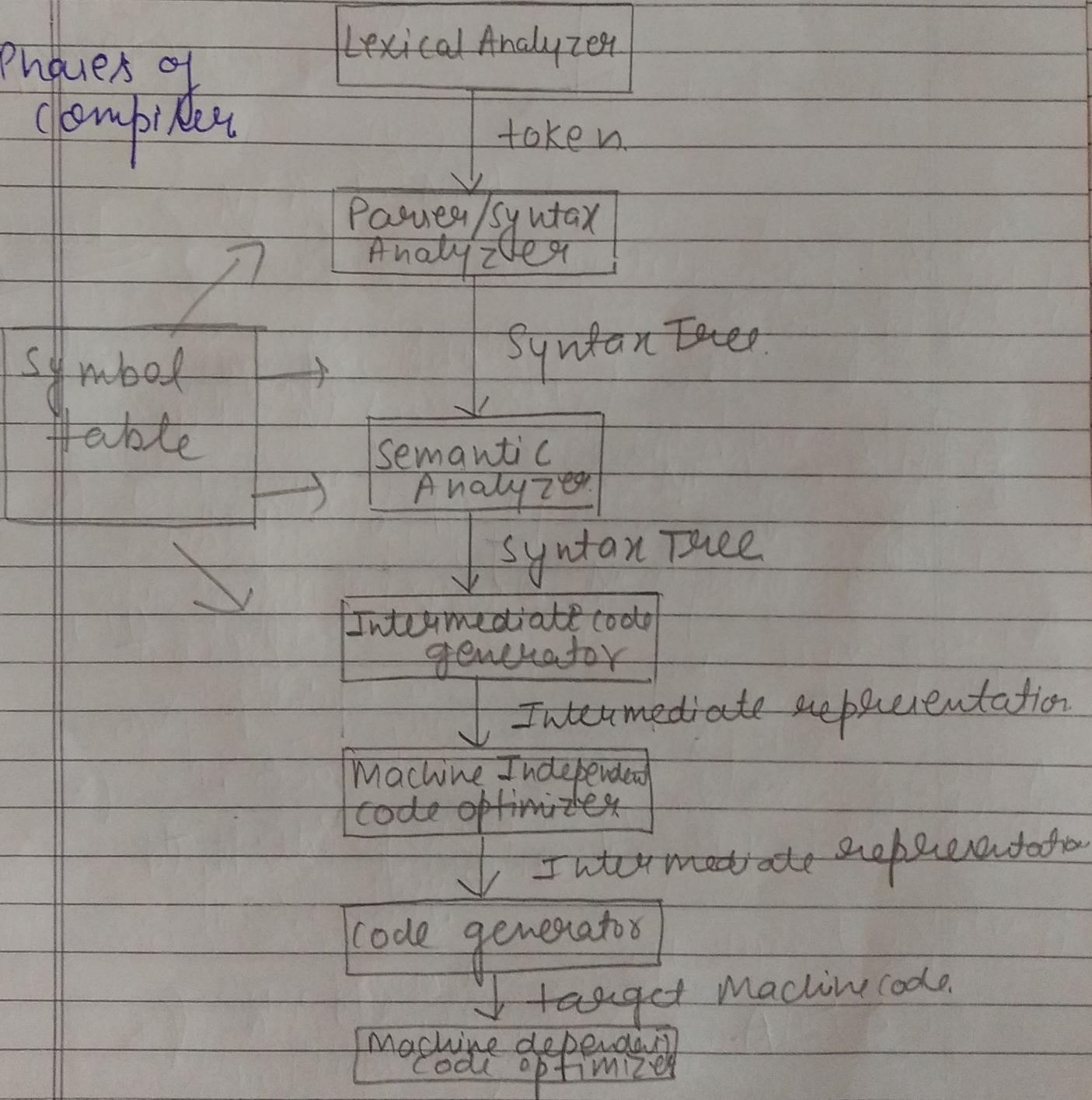
- 1) The states F_2 are all states from F_1 that have an outgoing transition labeled by symbol other than ϵ .
- 2) Determine for each state which states are reachable by ϵ -transition. If a state of F_1 can reach final state by ϵ -transition, then that state is also final state.
- 3) If two states have ϵ -transition, if f_1 is start state then F_2 with ϵ -transition from f_1 will also be a start state.
- 4) All the moves that are from F_1 will be duplicated for F_2 .

3

ANMOL
P. No.
Dt.

Ques 2 Describe - - - - compilation.
↓ characters

a) Phases of compiler



The source program after passing through preprocessor, is passed onto lexical analysis. The analysis part of compiler breaks into constituent pieces and imposes grammatical structure. The analysis part checks whether the code is syntactically incorrect. And if it is incorrect, then it makes changes. Then it passes it to synthesis part. The synthesis part constructs the target program from intermediate representation.

The phases of compiler are:

Lexical analysis - It reads stream of characters and groups them into meaningful sequences called lexemes. For each lexeme a token is generated

Syntax analysis - It uses the tokens produced to create a tree like representation.

The representation is ~~a~~ syntax tree in which each interior node represents an operation & children represent the arguments.

Semantic analysis - It uses syntax tree and the info in symbol table to check the source program for semantic consistency within language definition.

Intermediate code generation - A compiler may generate one or more intermediate representations.

Code optimization - It attempts to improve internal code so that better target will result.

Code generator - It takes intermediate representation of source code and maps it to target address.

Ques 3

The syntactic errors generally include:

- I) Error in structure
- II) Missing operator.
- III) Misspelled keyword
- IV) Unbalanced parentheses

Ex - ~~switck(ch)~~
 { -- .

The keyword switch is written as switck.

Error Recovery can be done using

- a) Panic Mode Recovery

Successive characters are removed one by one until set of undesignated sets synchronizing token is found.

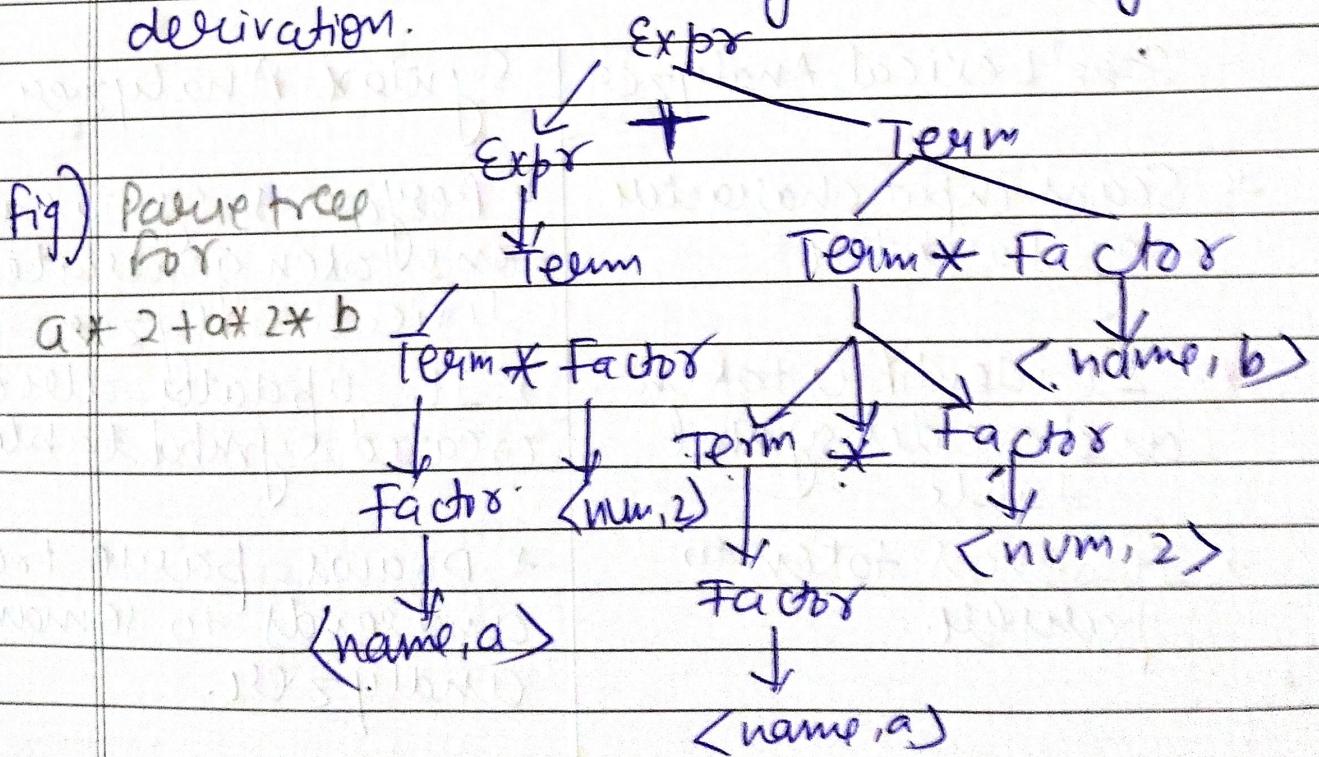
- b) Statement Mode Recovery
 When parser encounters error, it performs necessary correction remaining input so that rest of statement is parsed.
- c) If user has knowledge of common errors that can be encountered then, these errors can be incorporated by augmenting the grammar with error production.
- d) Global correction - The parser examines the whole program and tries to find closest match which is error free.

Ques 4 Difference - - - - -

Scm Lexical Analyzer	Syntax Analyzer
* Scans input character by character	* Performs syntax analysis on tokens generated by lexical analyzer.
* It identifies token and creates symbol table	* It updates already created symbol table.
* It sends token to parser.	* Draws parse tree and sends to semantic analyzer.

- (A)
- | | |
|----------|--|
| Page No. | |
| Date | |
- * It is first phase of compilation
 - * It is second phase of compilation
 - * New complex approach are used
 - * It requires a more complex approach
 - * It may not be portable
 - * Always portable.

Ques 6 Parse tree is a graphical representation for the derivation or parse that corresponds to input program. The parse tree is large relative to the source text because it represents the complete derivation with a node for each grammar symbol in derivation.



8

Page No. _____

Date _____

- * Parse trees are used primarily in discussion of parsing and in attribute grammars.
- * It helps in making syntax analysis by reflecting the syntax of the input language.
- * It uses an in-memory representation of the input with a structure that conforms to the grammar.
- * They will make multiple passes over info without having to reparse input.