1. Regular Language to Regular Expression (RE):

Ex: There will be a regular language like this-

{w| w has at least 3 symbol, and the third symbol is 1}

You have to convert that language to regular expression.

1. Regular Expression to string generation:

Ex: There will be a regular expression like this-

a\*b\*+b\*

you have write string for that RE.

1. Regular Expression to ɛ-NFA:

Ex: There will be a regular expression like this-

a\*b\*+b\*

you have to convert that RE to ɛ-NFA by using state elimination method.

* Union
* Concatenation
* \* operation

1. DFA to Regular Expression:

Ex: There will be a DFA diagram, you have to convert that diagram into RE by using state elimination method. For that you have to follow the following steps—

1. Convert the DFA to GNFA(Generalized Non-deterministic Finite Automata)

* Add a start state with start state of that DFA with a incoming transition
* Add a final state with the all final states of that DFA with a outgoing transition
* Then convert all symbols to RE like write a+b instead of a,b or 0+1 instead of 0,1

1. Then remove all state one by one according to your choice without new start state and new final states.
2. Context Free Grammar (CFG)

* CFG Construction

Ex: There can be a Regular language like {w|w starts with a} etc. You have to construct a CFG for that language

Or There can be a Context free language like anbn  ; n>= 1

* Formal Definition: You have define a CFG formally which is identify Variable(v), Terminals(T), Production (P), Start symbol (S).
* Derivation (Leftmost and Rightmost)
* Parse tree : Draw parse for particular provided string from a grammar(CFG)
* Ambiguity: You have to prove the ambiguity for provided grammar. You have to generate a string from that grammar which shows two different parse for that string. If this happens then the grammar is ambiguous if grammar shows unique parse tree all string then the grammar is not ambiguous.

1. Push down automata (PDA):

* PDA Construction: There will be Context free language like anbn  ; n>= 1 etc, you have to draw a PDA for that language
* Formal Definition: You have define a PDA formally which is identify Set of all states (Q), PDA input symbol (∑), start state(q0), set of all final states (F), transition function (ᵟ), stack symbol (z0), stack input (τ).
* Transition function(ᵟ) of PDA
* Push down automata theory:

3 components of PDA:

1. Input tape
2. Stack
3. control unit

