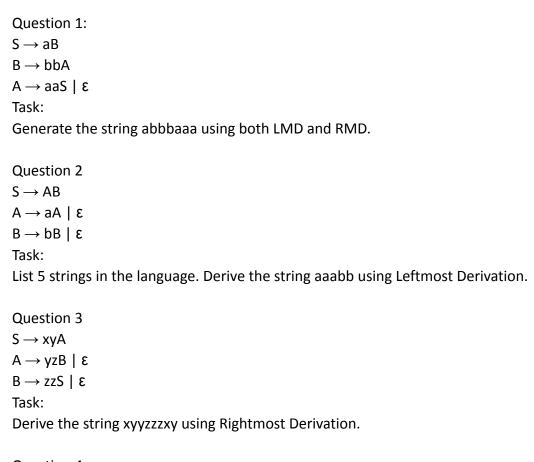
Practice Questions

Context Free Grammar

For the questions given below your task is to create Total Language Tree, Left Most Derivation (LMD), Right Most Derivation (RMD) to find the given string can be generated or not. Then from this grammar generate the FA.



Question 4

 $S \rightarrow abA$

 $A \rightarrow cdB \mid \epsilon$

 $B \rightarrow efS \mid \epsilon$

Task:

Give all derivation steps for generating abcdefab.

Question 5

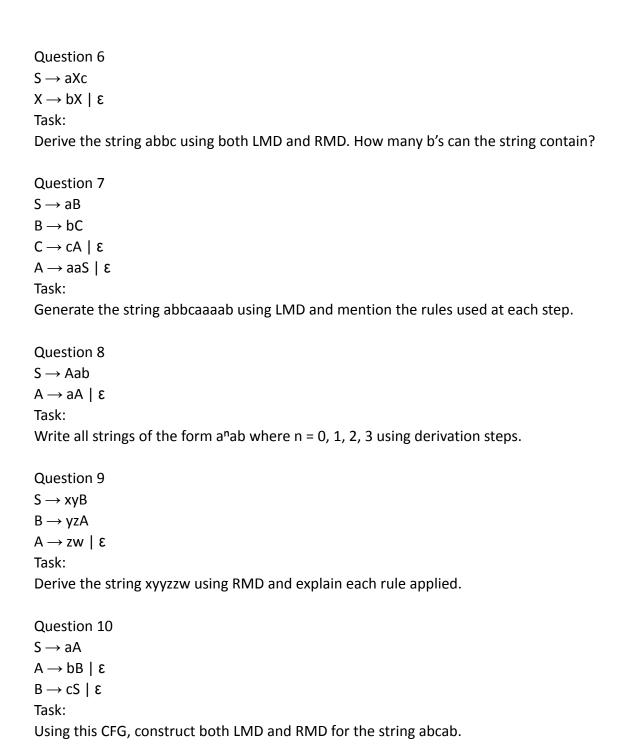
 $S \rightarrow aA \mid bB$

 $A \rightarrow bS \mid \epsilon$

 $B \rightarrow aS \mid \epsilon$

Task:

Construct the parse tree for the string aba.



Grammar

```
Phrase Structure Rules
Noun Phrase (NP)
NP → Art AdjList N
Adjective List
AdjList → Adj AdjList | E
Verb Phrase (VP)
VP \rightarrow V NP
VP \rightarrow AdvP V NP
VP \rightarrow V NP AdvP
VP \rightarrow V
Adverb Phrase (AdvP)
AdvP \rightarrow Adv AdvP \mid Adv
Lexicon Rules (Terminal Symbols)
Articles
Art \rightarrow the | a
Adjectives
Adj → itchy | jumpy | lazy | wild | sleepy | clever | huge
Nouns
N \rightarrow bear \mid dog \mid cat \mid tiger \mid fox \mid bird \mid worm \mid elephant \mid mouse \mid man \mid horse \mid ball \mid boys
girls
Verbs
V → hugs | chases | bites | sees | sleeps | runs | eats | finds | feeds | calls | kicks | grabs |
plays | sings
```

Now, use the following sentences and show either they can be generated or not using the above grammar.

- 1. the lazy dog bites the fox
- 2. the dog quickly eats the worm
- 3. the bear sees

Adverbs

4. a huge elephant slowly feeds the mouse

 $Adv \rightarrow quickly \mid slowly \mid silently \mid happily \mid angrily$

- 5. lazy dog eats
- 6. the bird the worm eats
- 7. a clever cat sees a huge dog
- 8. the fox happily
- 9. the boys run quickly
- 10. a jumpy dog silently sleeps
- 11. the girls plays the ball
- 12. the man kicks the ball slowly
- 13. elephant sleeps
- 14. a wild tiger feeds the worm happily
- 15. the the cat jumps

Push Down Automata

- 1. Design EvenPalindrome [s reverse (s), where s is in (a+b)*]. Construct the table having State, Stack and tape positions.
- 2. Design OddPalindrome [a b aaa aba bab bbb]. Construct the table having State, Stack and tape positions.
- 3. Explain the nondeterministic PDA or NPDA. Write five cases where NPDA will be design to solve the problem.
- 4. Design a deterministic PDA to accept the language [a^n b^n+1] . Assume n is positive number. Construct the table having State, Stack and tape positions.
- Design the logic and construct the PDA for the language L = {d^n and e^m where n= >1, m> n+2}. Construct the table having State, Stack and tape positions.
- 6. Design the logic and construct NPDA for accepting the language $L = \{a^m b^n c^m+n for m, n >= 1\}$. Construct the table having State, Stack and tape positions.
- 7. Design the logics and the NPDA for accepting the language L = {a^m b^ 2m+1, where m >= 1}. Construct the table having State, Stack and tape positions.
- 8. Design the logic and NPDA for accepting the languages $L = \{a^m b^n c^p d^q where m + n = p+q \}$. Construct the table having State, Stack and tape positions.
- Design the logic and the NPDA for the language L = {a^2m b^3m, where m>= 1}.
 Construct the table having State, Stack and tape positions.
- 10. Design the logic and NPDA for the language L = {0^I 1^j 2^k where i == j , or j==k ; I, j, k >=1}. Construct the table having State, Stack and tape positions.

Turing Machine

Show all the possible transactions and the position of Tape, Movement of Tape head, write the algorithm how it will work. Then make construct the TM.

1. Construct a TM for the language L = {0^n 1^n 2^n where n >=1}

- 2. Construct a TM for checking the palindrome of the string of even length for (0+1)*.
- 3. Construct a TM for checking the palindrome of the string of odd length for (0+1)*.
- 4. Construct a TM for adding two binary numbers.
- 5. Construct a TM that accepts binary numbers which are multiples of 3.
- 6. Construct a TM that doubles the given unary number on Tape drive.
- 7. Construct a TM to shift all symbols left by one cell.
- 8. Construct a TM to delete all b's in a string given on Tape drive.
- 9. Construct a TM to replace all the '01' with '10' on Tape drive.
- 10. Construct a TM to simulate the Binary Counter such as if 101 is given on tape drive it should write 110.
- 11. Construct a TM for Language L = {ww where w belongs (a+b)*}
- 12. Construct a TM for Language $L = \{ a^i b^i c^i \}$ where i and $i \ge 0 \}$

Mealy Moore Machine

Use multiple logical gates in the production rules and then convert the Mealy Machine to Moore Machine.

Use basic and extended gates

Generate the rules with delay 1 and delay 2 then make different machine then convert the machine into other machine.