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 1:
S → aB
B → bbA
A → aaS | ε
Task:
Generate the string abbbaaa using both LMD and RMD.

Question 2
S → AB
A → aA | ε
B → bB | ε
Task:
List 5 strings in the language. Derive the string aaabb using Leftmost Derivation.

Question 3
S → xyA
A → yzB | ε
B → zzS | ε
Task:
Derive the string xyyzzzxy using Rightmost Derivation.

Question 4
S → abA
A → cdB | ε
B → efS | ε
Task:
Give all derivation steps for generating abcdefab.

Question 5
S → aA | bB
A → bS | ε
B → aS | ε
Task:
Construct the parse tree for the string aba.

Question 6

$S \rightarrow aXc$

$X \rightarrow bX \mid \varepsilon$

Task:

Derive the string abbc using both LMD and RMD. How many b's can the string contain?

Question 7

$S \rightarrow aB$

$B \rightarrow bC$

$C \rightarrow cA \mid \varepsilon$

$A \rightarrow aaS \mid \varepsilon$

Task:

Generate the string abbcaaaab using LMD and mention the rules used at each step.

Question 8

$S \rightarrow Aab$

$A \rightarrow aA \mid \varepsilon$

Task:

Write all strings of the form $a^n ab$ where n = 0, 1, 2, 3 using derivation steps.

Question 9

$S \rightarrow xyB$

$B \rightarrow yzA$

$A \rightarrow zw \mid \varepsilon$

Task:

Derive the string xyyzzw using RMD and explain each rule applied.

Question 10

$S \rightarrow aA$

$A \rightarrow bB \mid \varepsilon$

$B \rightarrow cS \mid \varepsilon$

Task:

Using this CFG, construct both LMD and RMD for the string abcab.

Grammar

Phrase Structure Rules
Noun Phrase (NP)
NP ⟶ Art AdjList N

Adjective List
AdjList ⟶ Adj AdjList | ε

Verb Phrase (VP)
VP ⟶ V NP
VP ⟶ AdvP V NP
VP ⟶ V NP AdvP
VP ⟶ V

Adverb Phrase (AdvP)
AdvP ⟶ Adv AdvP | Adv

Lexicon Rules (Terminal Symbols)
Articles
Art ⟶ the | a
Adjectives
Adj ⟶ itchy | jumpy | lazy | wild | sleepy | clever | huge

Nouns
N ⟶ bear | dog | cat | tiger | fox | bird | worm | elephant | mouse | man | horse | ball | boys | girls

Verbs
V ⟶ hugs | chases | bites | sees | sleeps | runs | eats | finds | feeds | calls | kicks | grabs | plays | sings

Adverbs
Adv ⟶ quickly | slowly | silently | happily | angrily

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
4. a huge elephant slowly feeds the mouse

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.
5. 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.
9. 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*j where i and j >= 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.