CS1356 Introduction to Information Engineering

Homework 14

1. Which of the following matches the regular expression `a(ab)*a`
   (a) abababa   (b) aaba   (c) aabbaa   (d) aba   (e) aababababa

Ans: b , e

2. Which of the following matches the regular expression `ab+c?`
   (a) abc   (b) ac   (c) abbb   (d) bbc

Ans: a , c

3. Which of the following matches the regular expression `a(b|c)+`
   (a) abc   (b) abbbbbbb   (c) abcbcbcbc   (d) ac   (e) accbbbbcbcccc
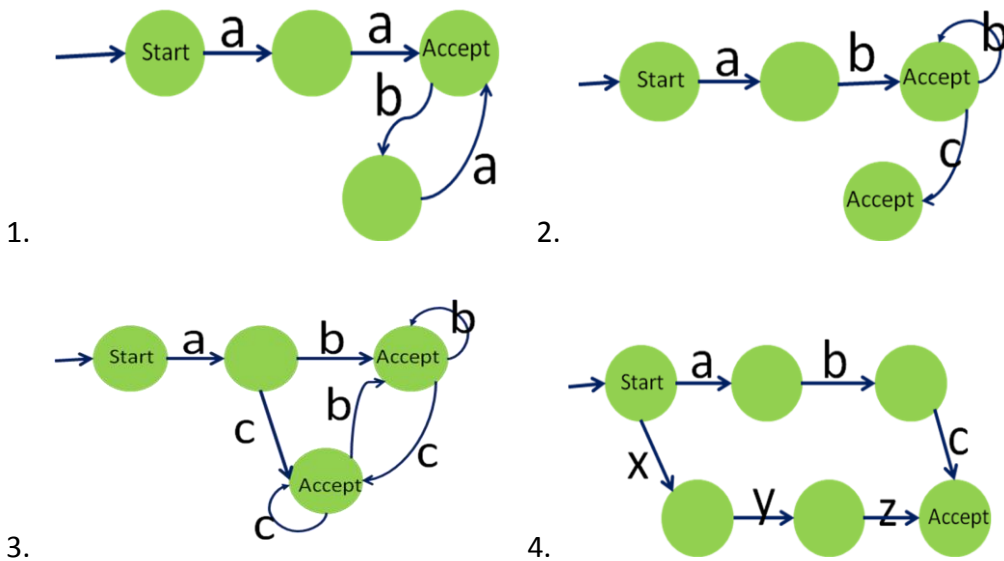
Ans: a , b , c , d , e

4. Which of the following matches the regular expression `abc|xyz`
   (a) abc   (b) xyz   (c) abc|xyz
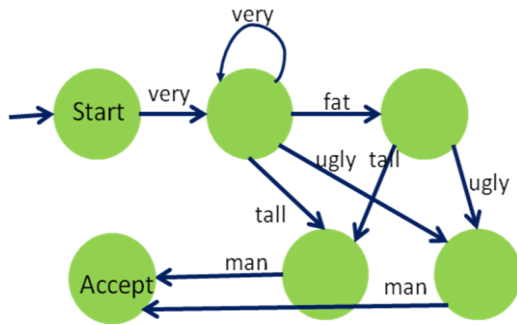
Ans: a , b

5. Which of the following matches the regular expression
   `(very )+(fat )?(tall|ugly) man`
   (a) very fat man   (b) fat tall man   (c) very very fat ugly man
   (d) very very very tall ugly man

Ans: c

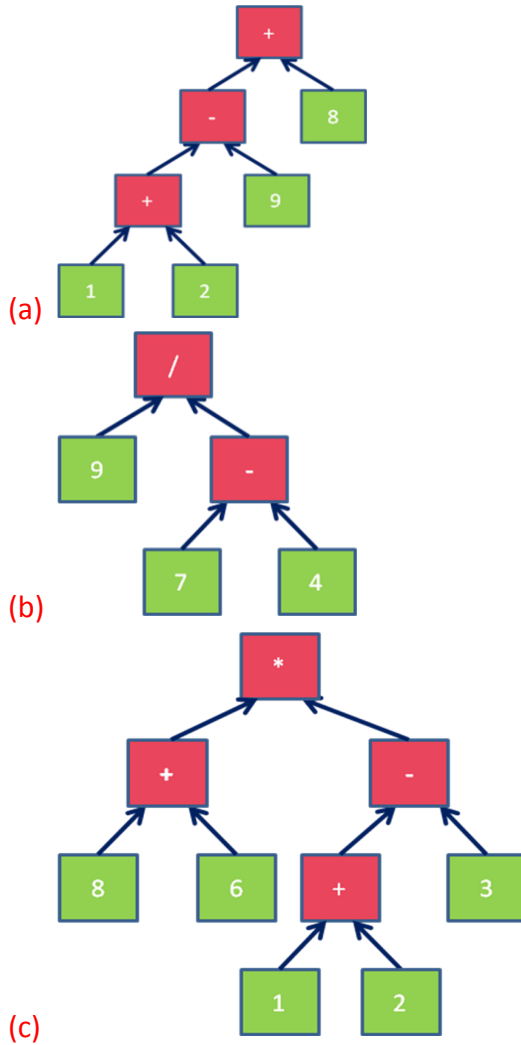6. Draw finite state machines for the regular expression in question 1 to question 5.

1.

2.

3.

4.

5.

7. Evaluate the following prefix notations

(a) + - + 1 2 9 8   (b) / 9 - 7 4   (c) * + 8 6 - + 1 2 3

Ans: (a)   2   (b)   3   (c)   0

8. Change the prefix notations in question 7 to their corresponding postfix notations, and draw their parsing tree
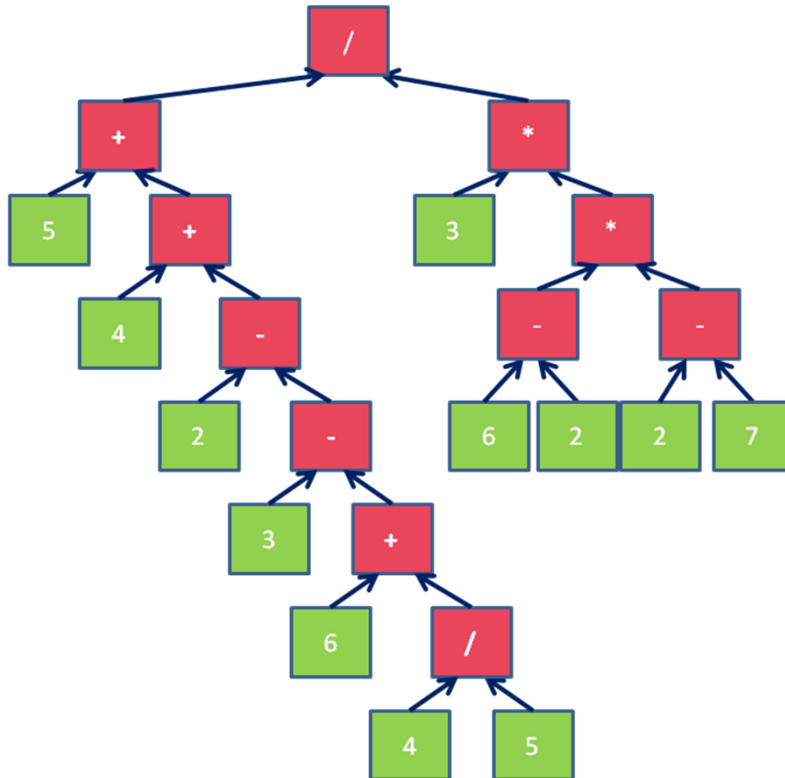
Ans: (a) 1 2 + 9 – 8 +   (b)   9 7 4 - /   (c) 8 6 + 1 2 + 3 - *

(a)

(b)

(c)

9. Convert the expression $\dfrac{5 + 4 + \left(2 - \left(3 - \left(6 + \frac{4}{5}\right)\right)\right)}{3\,(6-2)\,(2-7)}$ to its prefix notation and its postfix notation.

Ans:

(a) / + 5 + 4 − 2 − 3 + 6 / 4 5 * 3 * − 6 2 − 2 7

(b) 5 4 2 3 6 4 5 / + − − + + 3 6 2 − 2 7 − * * /

10. Think an algorithm to check the balance of the parentheses (括號) of an expression.   For example, when the input is "((3+(2)))*(3-(2)", it should return false, and when the input is "((3+(2)))*(3-(2))", it should return true.

(Hint: using stack.)

Ans:

Step1:

Check character in an expression. When meeting a character which is "(" , we push "(" into the stack. And when meeting a character which is ")" , we pop out an element "(" from the top of the stack. Other character we will skip it.

Step2:

When there is no element can be poped out from the stack , we return false.

When there is no character in expression , but still some elements in the stack , we return false.

When there is no character in expression and no elements in the stack , we return ture.