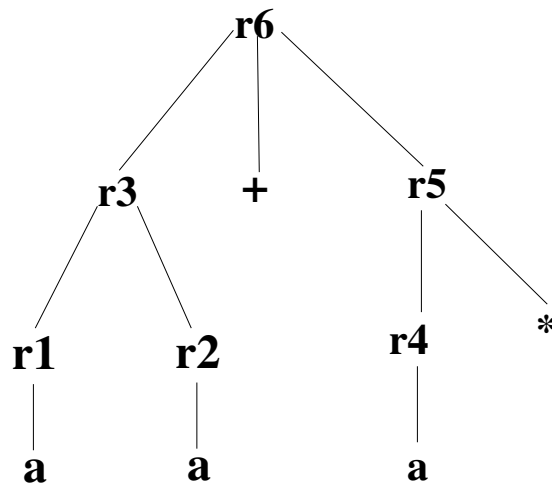


1) Mostre a árvore de derivação para a expressão regular: aa+a*



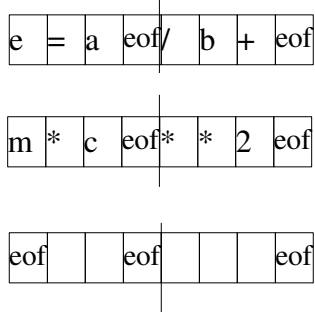
2) Defina os tokens para o trecho de código a seguir

```
float raiz_parcial (int k, float x){
    /*retorna raiz x parcial */
    return (x<= -10.0 || x>= 10.0)?k*x:x*x;
}
```

Tokens: <float> <ID, raiz_parcial>, <(>, <int>, <ID, k>, <,>, <float>, <ID, x>, <)>, <{>, <return>, <(>, <ID, x>, <=>, <REAL, -10.0>, <||>, <ID, x>, <=>, <REAL, 10.00>, <)>, <?>, <ID, k>, <*>, <ID, x>, <:>, <ID, x>, <*>, <ID,x><:>

3) Considere um par de buffers onde cada buffer armazena, no máximo, 4 bytes. Considerando que sentinelas (eof) são utilizados para indicar fim-de-buffer e fim-de-arquivo. Quantos pares de buffers são necessários para armazenar o seguinte código? Mostre o conteúdo de cada buffer.

e=a/b+m*c**2



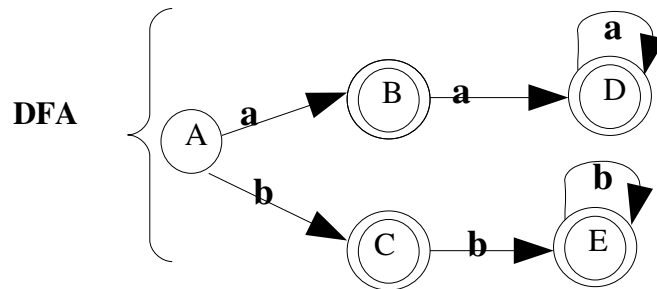
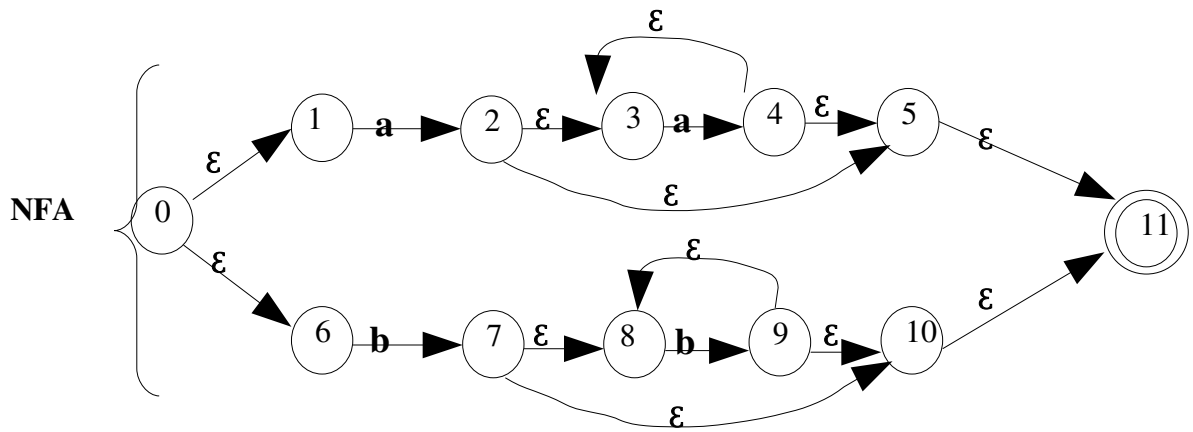
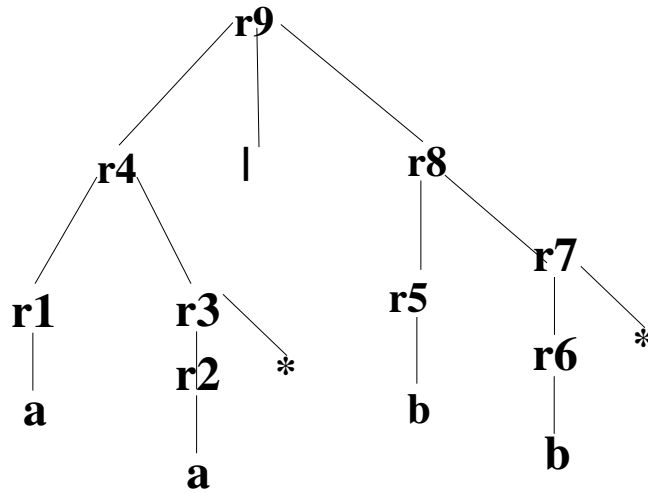
4) Descreva a linguagem gerada pelas seguintes expressões

a) a(alb)*a= Linguagem formada por cadeias que iniciam com o símbolo a, seguido por seqüências de 0 ou mais a e b, terminando com o símbolo a

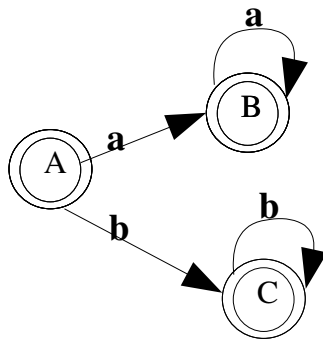
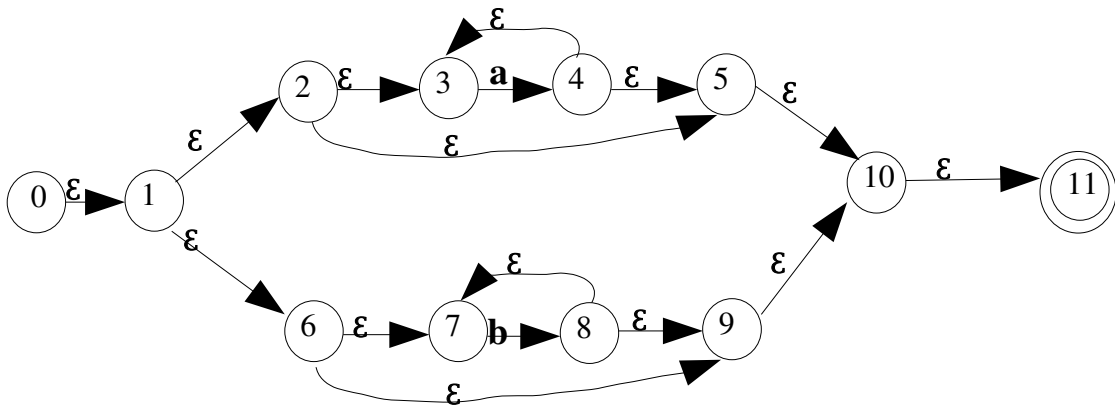
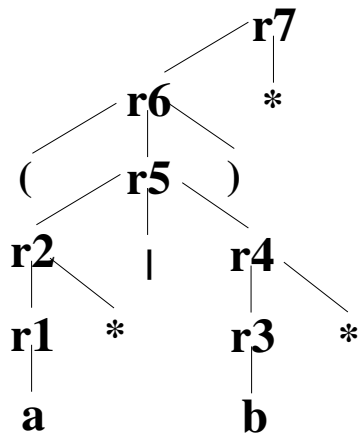
b) $a^*ba^*ba^*ba^*$ = Linguagem formada por cadeias que iniciam e terminam com 0 ou mais símbolos a, possuem, no máximo, 3 símbolos b separados entre si por 0 ou mais símbolos a.

5) Converter as seguintes expressões regulares para DFA:

a) $aa^* | bb^*$



b) $(a^*lb^*)^*$



6) Transforme o NFA a seguir para um DFA

