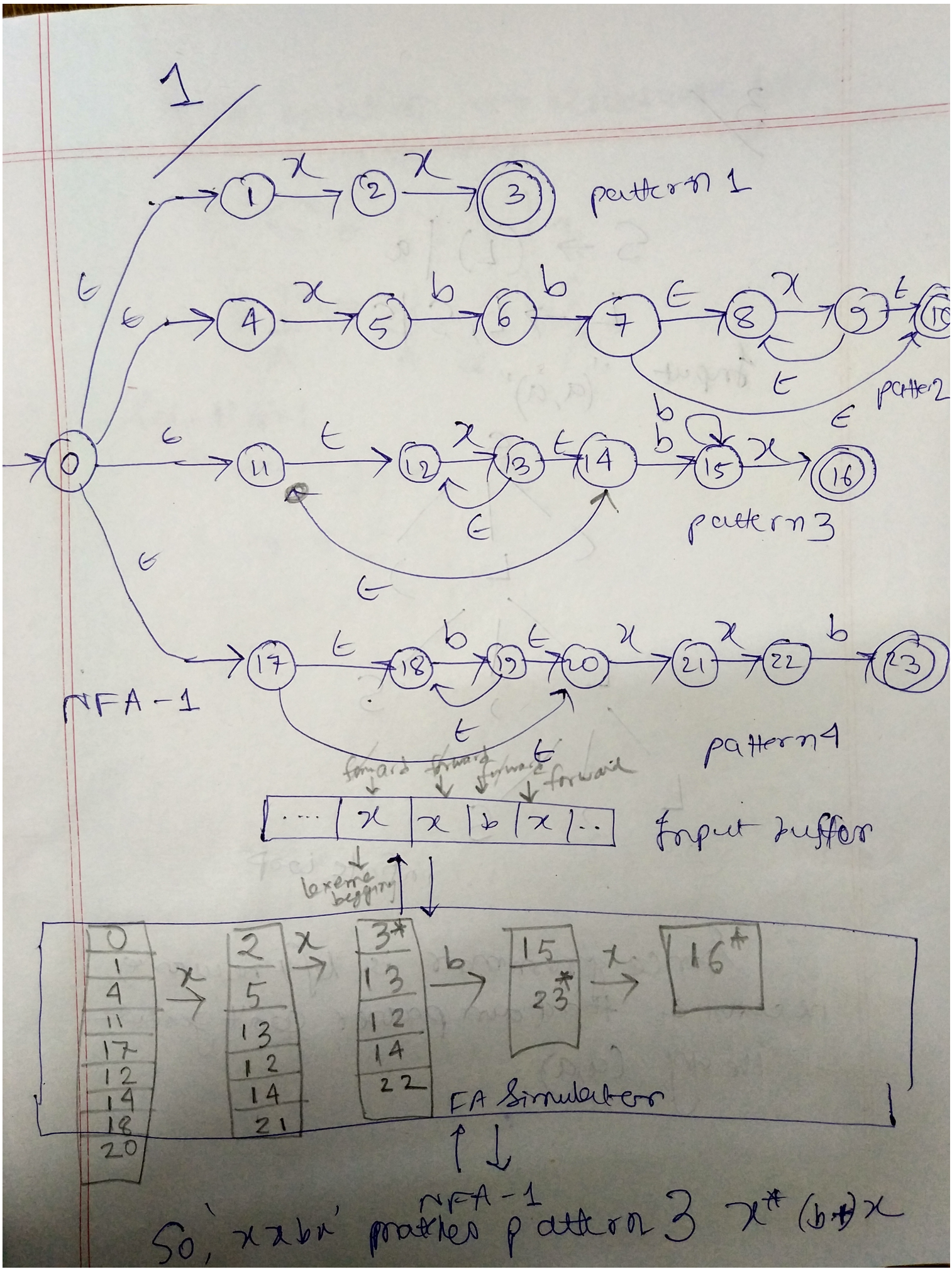

1
pattern 1
pattern 2
pattern 3
pattern 4
NFA - 1
forward
forward
forward
forward
input buffer
lexeme beginning
FA Simulator
NFA - 1
So, 'xxbx' matches pattern 3 x* (b+)x

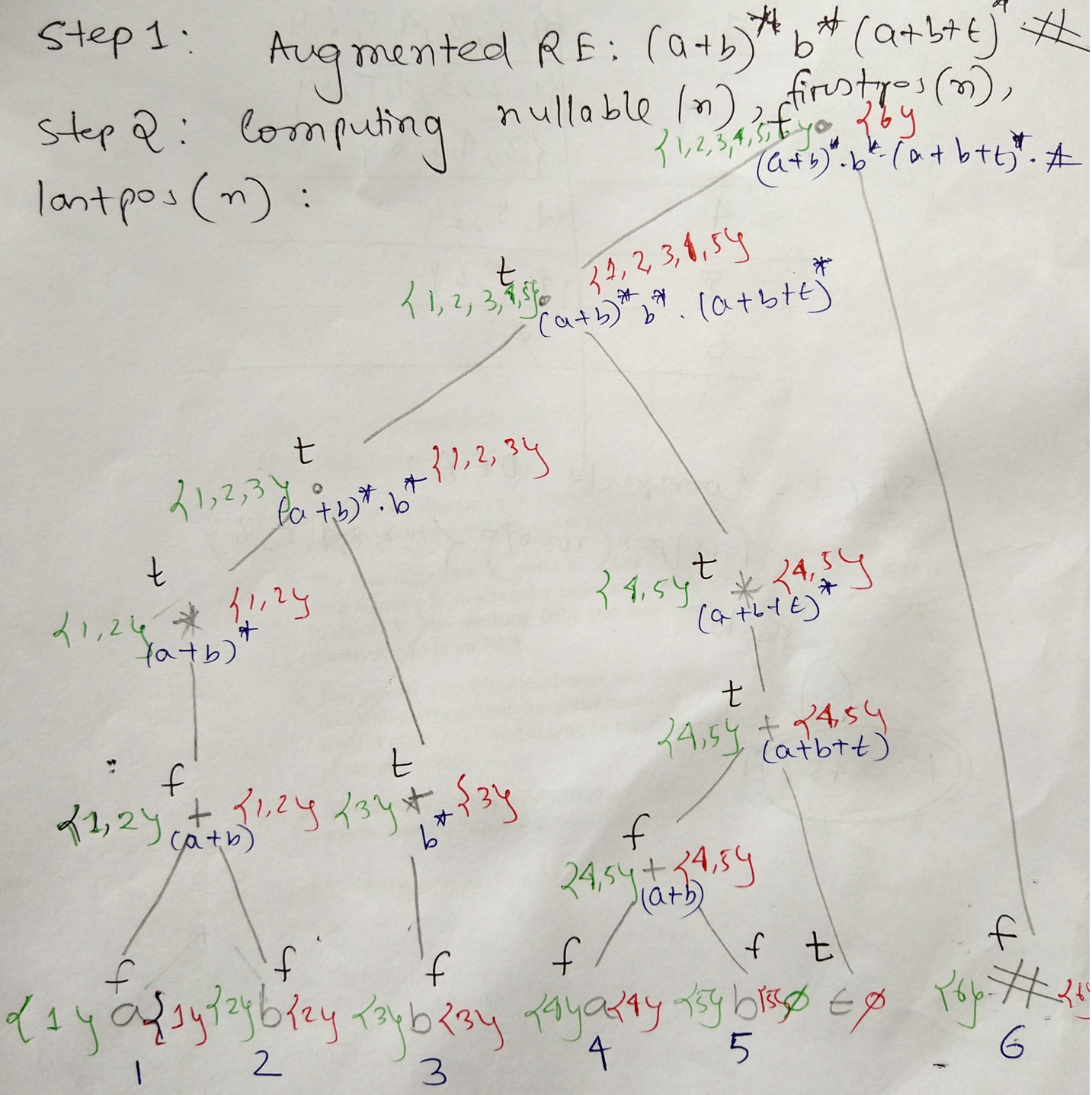
Convert the RE directly to DFA:

$(a+b)^* b^* (a+b+t)^*$

Solution:-

Step 1: Augmented RE: $(a+b)^* b^* (a+b+t)^* \#$

Step 2: Computing nullable(n), firstpos(n), lastpos(n):



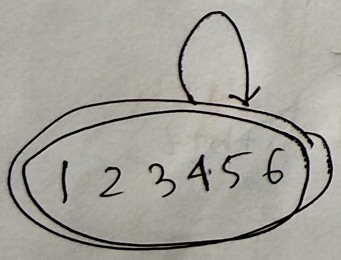
Step 3: Compute followpos(n):

node (n)	followpos (n)
1	{1, 2, 3, 4, 5, 6}
2	{1, 2, 3, 4, 5, 6}
3	{3, 4, 5, 6}
4	{4, 5, 6}
5	{4, 5, 6}
6	∅

Step 4: Compute DFA:

firstpos(root) = {1, 2, 3, 4, 5, 6}

a, b

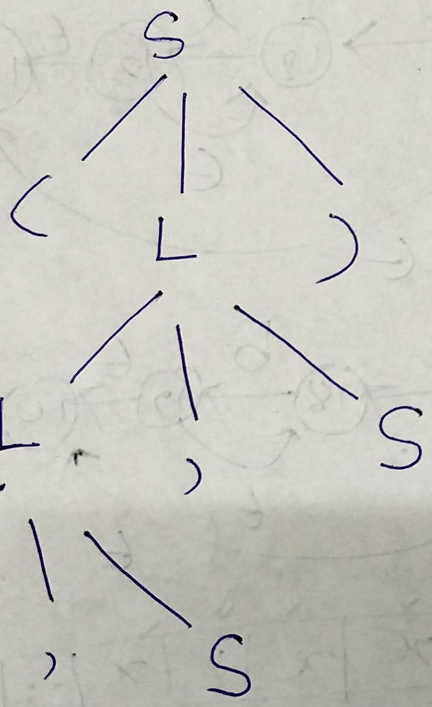


3/

$S \rightarrow (L) \mid a$

$L \rightarrow L, S \mid S$

Input "(a,a)"



infinite loop

Since grammar is left recursive, recursive top down parser can't generate the string (a,a)

The solution is to eliminate left recursion.

$$S \rightarrow (L) \mid a$$

$$\underset{A}{L} \rightarrow \underset{A}{L}\underbrace{,S}_{\alpha} \mid \underset{\beta}{S}$$

Solution:

$$S \rightarrow (L) \mid a$$

$$L \rightarrow SL'$$

$$L' \rightarrow ,SL' \mid \epsilon$$