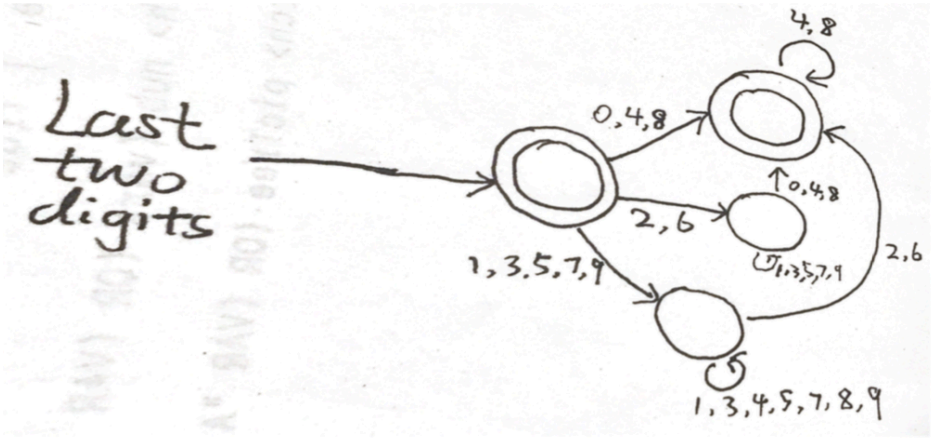
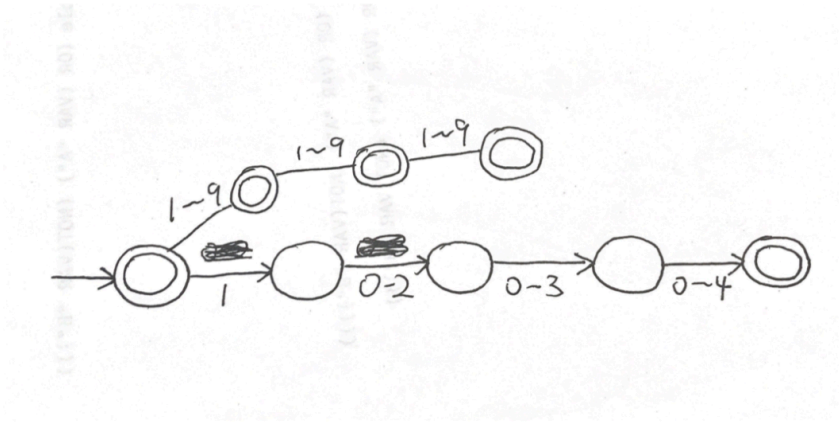
QUESTION 1:

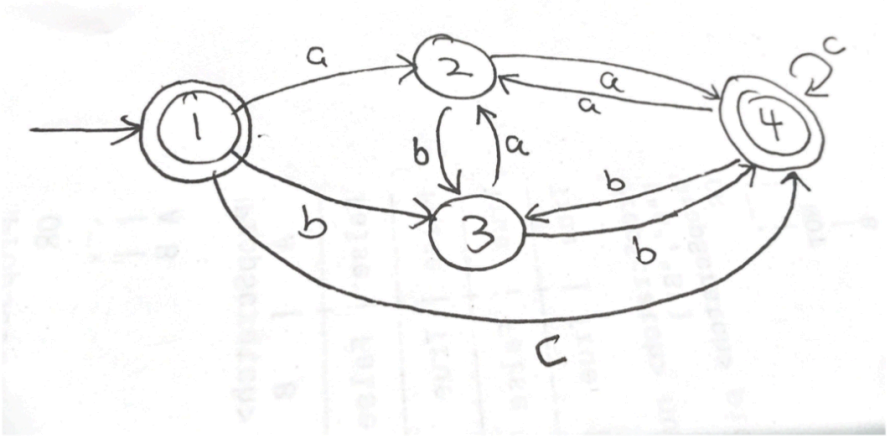
a)



b)



c)



QUESTION 2:

a)

```
GrammarScratch> m
"M -> x ; M -> M , x"
GrammarScratch> langSentences 5 m
1) x
2) x , x
3) x , x , x
4) x , x , x , x
5) x , x , x , x , x

GrammarScratch> ambigExamples 100 m
No ambiguity examples found

GrammarScratch> langDerivations 5 m
1) M  ==>  x
2) M  ==>  M , x  ==>  x , x
3) M  ==>  M , x  ==>  M , x , x  ==>  x , x , x
4) M  ==>  M , x  ==>  M , x , x  ==>  M , x , x , x  ==>  x , x , x , x
5) M  ==>  M , x  ==>  M , x , x  ==>  M , x , x , x  ==>  M , x , x , x , x  ==>  x , x , x , x , x

GrammarScratch> terminals (readGrammar m)
["x",","]
GrammarScratch>
```

b)

```
GrammarScratch> m1
" P -> [ ]; P -> [ M ] ; M -> x; M -> x , M ;"
GrammarScratch> langSentences 5 m1
1) [ ]
2) [ x ]
3) [ x , x ]
4) [ x , x , x ]
5) [ x , x , x , x ]

GrammarScratch> ambigExamples 100 m1
No ambiguity examples found

GrammarScratch> langDerivations 5 m1
1) P  ==>  [ ]
2) P  ==>  [ M ]  ==>  [ x ]
3) P  ==>  [ M ]  ==>  [ x , M ]  ==>  [ x , x ]
4) P  ==>  [ M ]  ==>  [ x , M ]  ==>  [ x , x , M ]  ==>  [ x , x , x ]
5) P  ==>  [ M ]  ==>  [ x , M ]  ==>  [ x , x , M ]  ==>  [ x , x , x , M ]  ==>  [ x , x , x , x ]

GrammarScratch> terminals (readGrammar m1)
["[","]","x",","]
GrammarScratch>
```

c)

```
GrammarScratch> m2
"M -> x . M ; M->"
GrammarScratch> langSentences 5 m2
1)
2) x .
3) x . x .
4) x . x . x .
5) x . x . x . x .

GrammarScratch> ambigExamples 100 m2
No ambiguity examples found

GrammarScratch> langDerivations 5 m2
1) M  ==>
2) M  ==>  x . M  ==>  x .
3) M  ==>  x . M  ==>  x . x . M  ==>  x . x .
4) M  ==>  x . M  ==>  x . x . M  ==>  x . x . x . M  ==>  x . x . x .
5) M  ==>  x . M  ==>  x . x . M  ==>  x . x . x . M  ==>  x . x . x . x . M  ==>  x . x . x . x .

GrammarScratch> terminals (readGrammar m2)
["x","."]
GrammarScratch>
```

d)

```
GrammarScratch> m3
"M -> a M a ; M -> b M b; M -> c M c ; M -> ; M -> a ; M -> b ; M -> c"
GrammarScratch> langSentences 20 m3
1)
2) a
3) b
4) c
5) a a
6) a a a
7) a b a
8) a c a
9) b b
10) b a b
11) b b b
12) b c b
13) c c
14) c a c
15) c b c
16) c c c
17) a a a a
18) a a a a a
19) a a b a a
20) a a c a a

GrammarScratch> ambigExamples 100 m3
No ambiguity examples found

GrammarScratch> langDerivations 10 m3
1) M  ==>
2) M  ==>  a
3) M  ==>  b
4) M  ==>  c
5) M  ==>  a M a  ==>  a a
6) M  ==>  a M a  ==>  a a a
7) M  ==>  a M a  ==>  a b a
8) M  ==>  a M a  ==>  a c a
9) M  ==>  b M b  ==>  b b
10) M  ==>  b M b  ==>  b a b

GrammarScratch> terminals (readGrammar m3)
["a","b","c"]
GrammarScratch>
```

e)

```
GrammarScratch> m4
"M -> ; M -> 0 ; M -> M 3 ; M -> M 6 ; M -> M 9 ; M -> M 1 2 ; M -> M 2 1 ;M -> M 5 1 ; M -> M 1 5 ; M -> M
4 2 ; M-> M 2 4 ; M -> M 5 4 ; M -> M 4 5 ; M -> M 2 7 ; M -> M 7 2 ; M -> M 1 8 ; M -> M 8 1 ; M -> M 7 8 ;
 M -> M 8 7 "
GrammarScratch> langSentences 20 m4
1)
2) 0
3) 3
4) 0 3
5) 6
6) 0 6
7) 9
8) 0 9
9) 1 2
10) 0 1 2
11) 2 1
12) 0 2 1
13) 5 1
14) 0 5 1
15) 1 5
16) 0 1 5
17) 4 2
18) 0 4 2
19) 2 4
20) 0 2 4

GrammarScratch> ambigExamples 100 m4
No ambiguity examples found

GrammarScratch> langDerivations 10 m4
1) M  ==>
2) M  ==>  0
3) M  ==>  M 3  ==>  3
4) M  ==>  M 3  ==>  0 3
5) M  ==>  M 6  ==>  6
6) M  ==>  M 6  ==>  0 6
7) M  ==>  M 9  ==>  9
8) M  ==>  M 9  ==>  0 9
9) M  ==>  M 1 2  ==>  1 2
10) M  ==>  M 1 2  ==>  0 1 2

GrammarScratch> terminals (readGrammar m4)
["0","3","6","9","1","2","5","4","7","8"]
GrammarScratch>
```

f)

```
[GrammarScratch> m6                                                                                                    ]
"P -> M ; P -> P | M ; M -> n => Q ; Q -> ; Q -> Q n; Q -> Q t"
[GrammarScratch> langSentences 16 m6                                                                                   ]
1) n =>
2) n => n
3) n => t
4) n => n n
5) n => t n
6) n => n t
7) n => t t
8) n => n n n
9) n => t n n
10) n => n t n
11) n => t t n
12) n => n n t
13) n => t n t
14) n => n t t
15) n => t t t
16) n => | n =>

[GrammarScratch> ambigExamples 100 m6                                                                                  ]
No ambiguity examples found

[GrammarScratch> langDerivations 10 m6                                                                                 ]
1) P  ==>  M  ==>  n => Q  ==>  n =>
2) P  ==>  M  ==>  n => Q  ==>  n => Q n  ==>  n => n
3) P  ==>  M  ==>  n => Q  ==>  n => Q t  ==>  n => t
4) P  ==>  M  ==>  n => Q  ==>  n => Q n  ==>  n => Q n n  ==>  n => n n
5) P  ==>  M  ==>  n => Q  ==>  n => Q n  ==>  n => Q t n  ==>  n => t n
6) P  ==>  M  ==>  n => Q  ==>  n => Q t  ==>  n => Q n t  ==>  n => n t
7) P  ==>  M  ==>  n => Q  ==>  n => Q t  ==>  n => Q t t  ==>  n => t t
8) P  ==>  M  ==>  n => Q  ==>  n => Q n  ==>  n => Q n n  ==>  n => Q n n n  ==>  n => n n n
9) P  ==>  M  ==>  n => Q  ==>  n => Q n  ==>  n => Q n n  ==>  n => Q t n n  ==>  n => t n n
10) P  ==>  M  ==>  n => Q  ==>  n => Q n  ==>  n => Q t n  ==>  n => Q n t n  ==>  n => n t
n

[GrammarScratch> terminals (readGrammar m6)                                                                            ]
["|","n","=>","t"]
GrammarScratch>
```

QUESTION 3:

a) > a|bc have the same result with a|(bc), but not same with (a|b)c
> ab*c have the same result with a(b*c), but not same with (ab)*c
> a*bc is not same with a*(bc)
> a*(b|c) is an example of the alternatives group to the right
> (ab)*c is an example of sequencing groups to the left

b) " R -> (M) ; R -> R * E ; M -> a | b ; E -> a b"

E -> a b is sequence which means after a, b will execute.
R -> R * E is repetition which means R repeat again with E.
E -> a | b is alternative which means E goes to a or E goes to b.
R -> (M) is group expressive.
Alternative groups to the right and sequencing groups to the left.

```
GrammarScratch> un
" R -> ( M ) ; R -> R * E ; M -> a | b ; E -> a b"
GrammarScratch> sentences 10 un
1) R
2) ( M )
3) R * E
4) ( a | b )
5) ( M ) * E
6) R * E * E
7) R * a b
8) ( a | b ) * E
9) ( M ) * a b
10) ( M ) * E * E

GrammarScratch> ambigExamples 100 un
No ambiguity examples found

GrammarScratch> terminals (readGrammar un)
["(",")","*","a","|","b"]
```

c)

```
GrammarScratch> parseTree 2 un
   R
   |
.---.
| | |
( M )

GrammarScratch> treeLangDerivation 2 un
R   ==>     R     ==>         R      ==>          R       ==>
            |                 |                   |
         .---.           .-----.             .-------.
         | | |           |   | |             |     | |
         R * E           R   * E             R     * E
                         |                   |
                       .---.               .-------.
                       | | |               |   |   |
                       ( M )               (   M   )
                                           |
                                         .---.
                                         | | |
                                         a | b


                             R
                             |
                          .--------.
                          |     |  |
                          R     *  E
                          |        |
                       .-------.   .-.
                       |   |   |   | |
                       (   M   )   a b
                           |
                         .---.
                         | | |
                         a | b

GrammarScratch>
```