Midterm 342 : Formal languages and Automata Theory
50 minutes, 40 points

Name: Julian Garcia
ID: 900334702

Definitions (6 points 6 minutes)

1. (1 point 1 minute) What is an alphabet? Give an example

   An alphabet is a non-empty, finite set. Denoted by Σ
   Example: Σ ={a,b,c}

2. (1 point 1 minute) What is a language over the alphabet A? Give an example

   A set of strings over an alphabet which can be empty, finite, or infinite.
   Example:
   Σ ={a,b,c}
   L ={'abc', 'acb', 'aba'}

3. (1 point 1 minute) What is the configuration of a DFA M?
   The Configuration represents the current state of the machine M on input X and the remaining contents of X yet to be read

   The configuration of a DFA M consists of:
   An alphabet Σ
   Its set of states Q
   The start state S which is in Q
   The accepting state A which is in Q
   The transitions δ, which is a function over Q and Σ to Q

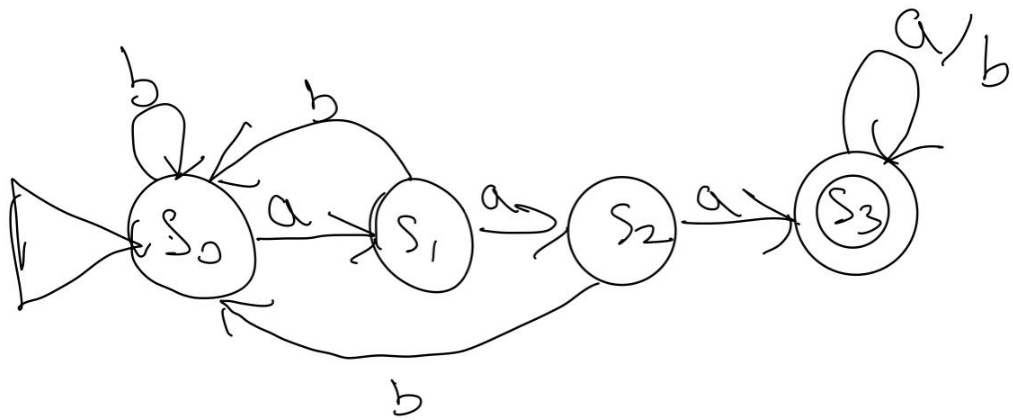
4. (2 point 2 minute) What is the trace of a DFA M on input w? How can you use the trace to determine if w is accepted by M or not.
   The trace is a sequence of states the DFA M goes through while processing the input w. You can use the trace to determine if w is accepted by determining if the final state of the trace is an accepting state in the DFA.

5. (1 point 1 minute) Define a non-regular language – be accurate in your definition. (I will not give you the benefit of doubt for unclear quantification).

   A non-regular language is a language for which no DFA exists.

Designing DFA/REX (8 points 8 minutes)

6. (5 points 5 minutes)
   a. Draw the DFA for the following language over {a,b}. L = {w | w contains the substring aaa}.



   b. Give the trace on the string w = "abaaaba"

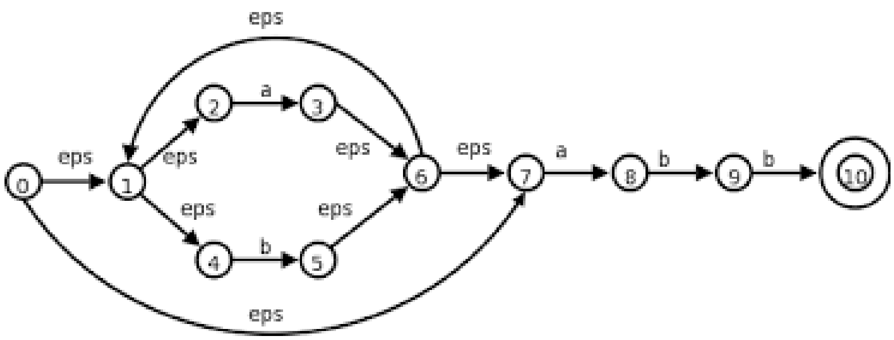
$\{S_0\} \xrightarrow{a} \{S_1\} \xrightarrow{b} \{S_0\} \xrightarrow{a} \{S_1\} \xrightarrow{a} \{S_2\} \xrightarrow{a} \{S_3\} \xrightarrow{b} \{S_3\} \xrightarrow{a} \{S_3\}$

7. (3 points 3 minutes) Give the regular expression over {a,b} for the language containing even-lengthed strings.

   (aa|bb|ab|ba)*

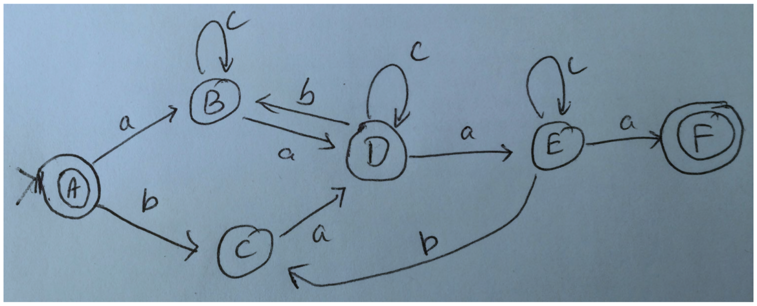
Algorithms (10 points 10 minutes)

8. (5 points 5 minutes) NFA to DFA: Find the eps-closure of state 0. Then say what the next state would be on input “a”.

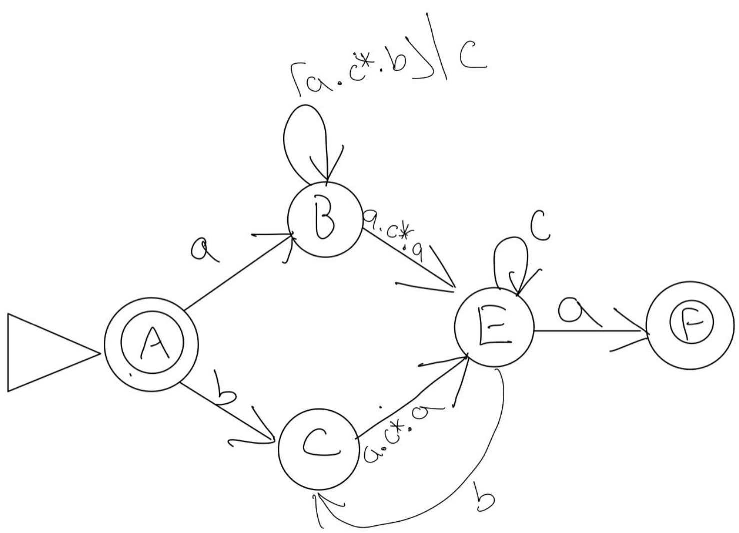


ϵ closure of state 0 = {0,1,2,4,7}
next state on input “a” = {3,8}

9. (5 points, 5 minutes) Eliminate state D and draw the DFA without D. Assume D is the first state you are eliminating



Path from B to E: a.c*.a
Path from B to B: a.c*.b
Path from C to E: a.c*.a
Resultant DFA:

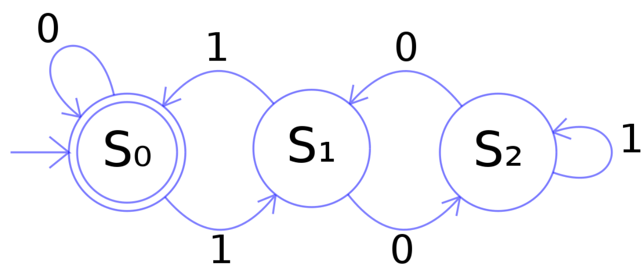


Pumping Lemma (12 points 12 minutes)

10. (3 points 3 minutes) State the Pumping Lemma for Regular Languages

$\forall$ L, if L is regular then
$\exists$ p ϵ N &
$\forall$ strings s. If L(s) & |s|$\geq$p then
$\exists$ x,y,z such that
  s = x.y.z &
  |y| > 0 &
  |x.y| $\leq$ p
  $\forall$ i ϵ N. L($x.y^i.z$)

11. (4 points 4 minutes) Verify pumping lemma for the following DFA



p = 3
s is accepted & |s| ≥ 3
s = 101...01
x= 10
y=1…
z=01

Since s is accepted, the pumping lemma is verified

12. (5 points 5 minutes) Prove using pumping lemma that
L = {w |w= $a^n b^{2n} c^{3n}$ for some natural number n} is not regular.
Strings in L:
abbccc
aabbbbcccccc

Strings NOT in L:
abc
abbbcc

Proof By contradiction:
By pumping lemma,
Assume L is Regular,
So there is a DFA M and p is the number of states in the DFA

Consider a string $s = a^p b^{2p} c^{3p}$ such that L(s) & $|s| \geq p$

Splits into any xyz
s = x.y.z
$|y| > 0$
$|xy| \leq p$

Loop → y is all b's
Pump loop through b^{2p}

$s' = xyyz = a^p b^{(2p + y)} c^{3p}$

s' is accepted by the DFA but not in the language,
Therefore L is not a regular language

Misc (4 points 10 minutes)

13. (1 point 1 minute) Give languages L1 and L2 such that L1 is a subset of L2, L2 is regular but L1 is not.
    $L1 = \{a^{2n} b^n c^{2n}\}$

    $L2 = \{a^2 b^1 c^2\}$

14. (3 points 3 minutes) What is wrong with the following proof?
    (Extra credit 4 points) Fix it

    Proof that the following L is not regular using pumping lemma:
    L = {x | there exists u, v. x = u.v and #a(u) = #a(v)}, i.e., x can be split into u and v such that the number of a's in u is the same as the number of a's in v.

    Proof by contradiction
    1. Assume L is regular. So it has a DFA with say q states. Then its pumping length is q.
    2. Consider the string w = a^q.b.a^q
        a. |w| >q
        b. L(w). This is because we can take u = a^q and v = b.a^q.
    3. There is some PLS such that w = PLS and |PL|<q. So, a^q has a loop. This loop L is made of only a's. So, when we pump more a's will be added.
    4. So, in PLLS, u has more a's than v. So, PLLS is not in L
    5. Since the conclusion of the pumping lemma is violated, the premise has to be false. So L is not regular.

    The problem comes in the string defined, as the language states that a's are in the language but b's are not. Also the only way to ensure that u > v or vice versa is to compare them on a basis of if the string is even or not, as the language says that THERE EXISTS a way to split u and v such that they're equal, and this would require that the total length of the string be even. Therefore to prove this you can show that the DFA accepts a string of odd length.

Instead we should consider the string w=a^2q. a^2q

Corrected Proof:
Proof by contradiction

6. Assume L is regular. So it has a DFA with say q states. Then its pumping length is q.
7. Consider the string w = a^2q.a^2q
    a. |w| >q
    b. L(w). This is because we can take u = a^2q and v = b.a^2q.
8. There is some PLS such that w = PLS and |PL|<q. So, a^2q has a loop. This loop L is made of only a's. So, when we pump more a's will be added.
9. So, in PLLS we would have PLLS = $a^{2q + y}c^{2q}$
10. Since the conclusion of the pumping lemma is violated, the premise has to be false. So L is not regular.