Automata and Formal Languages
Final

Name: Julian Garcia
900334702___________

Total: 75 points

Finals is to be done individually
You may use notes from class, lectures and other resources that are posted and the textbook.
You may not google for answers, or clarifications for questions

If you have any question whatsoever , please email me directly.

1. *(1 point 1 min)* What is an alphabet

   A non-empty, finite set

2. *(1 point 1 min)*What is a language

   A set of strings over an alphabet which can be empty, finite, or infinite.

3. *(1 point 1 min)*Give the inductive definition of regular expressions over {a, b} (i.e., what constitutes a regular expression)
   Recursively defined over alphabet {a, b}
   Base cases: {}, ϵ, a, b
   Recursively closed under:
   Disjunction (|)
   Kleene-star (*)
   Concatenation (.)

4. *(1 point 1 min)*What is the configuration of a machine? DFA? PDA?
   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 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

   The configuration of a PDA consists of:
   An alphabet Σ
   A stack alphabet Γ
   A finite set of states Q
   The start state S which is in Q
   A set of accepting states A which is in Q
   The transitions δ, which is a function over Q and Σ to Q

5. *(1 point 1 min)*Give 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
$\exists$ u,w,x,y,z such that
s = u.w.x.y.z &
|wy| > 0 &
|wxy| $\leq$ p
$\forall$ i ϵ N. L(u.w^i.x.y^i.z)

6. *(1 point 1 min)*Give the pumping lemma for context free languages
$\forall$ L, if L is Context Free 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)

7. *(1 point 1 min)*State the halting problem

The halting problem is a decision problem that tries to determine if a machine will finish running.

8. *(1 point 1 min)*Can a Context free language be regular? Can a context free language be non-regular?
   A context free language can be regular, and a context free language can be non-regular.

9. *(1 point 1 min)*Can a regular language be context free? Can a regular language be non-context free?
   All regular languages are context free, as regular languages are subsets of context free languages, therefore there cannot be a regular language that is not context free.

10. *(1 point 1 min)* Can a regular language be non-context free? Can a non-regular language be context free?
    Again, a regular language cannot be non-context free.
    A non-regular language however can be context free and can be non-context free. Since regular languages are a subset of context free languages, that implies that non-regular languages can be non-context free.

11. *(10 point 10 min)*
    (a) Consider a language of strings over {1} such that the string read as a unary number is a power of 2. Is this language regular or not? If regular, give the DFA. If non-regular, show that using pumping lemma.

    Strings Accepted would look like:
    "1", "11", "1111", "11111111" …

    This language is not regular

    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 $s = 1^{2p}$ such that L(s) & $|s| \geq p$

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

Loop → y is 1’s at beginning of string at length less than P

Pump loop through list of 1’s less than p

s’ = xyyz = $1^{2p}1^{lengthOfLoop} \leq 1^{2p}1^{p}$

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

(b)Consider a language of strings over {0,1} such that the string read as a binary number is a power of 2. Is this language regular or non-regular. If regular, give the DFA. If non-regular, show that using pumping lemma

Strings accepted here would look like:
“0”, “1”, “00”, “11”, “0000”, “0011”, “01010101”, …
This language also isn’t regular

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 = $0^{2p}1^{2p}$ such that L(s) & |s| ≥ p

Splits into any xyz
s = x.y.z
|y| > 0
|xy| ≤ p

Loop → y is in 0’s

Pump loop through 0^{2p}

s’ = xyyz = $0^{(2p + y)}1^{2p}$

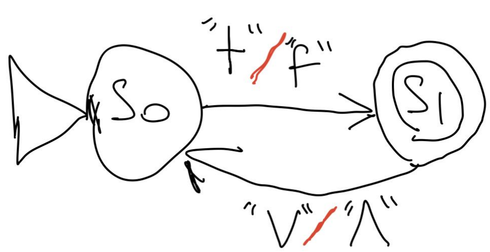
s’ is accepted by the DFA but not in the language,
Therefore L is not a regular language

12. *(5 point 5 min)*Consider the following grammar that defines Boolean expressions over Boolean constants t and f. Nonterminals = {E}; Terminals = {t, f, ∧, ∨ }. Production rules are: “E □ t | f | E ∧ E | E ∨ E“. Give all possible parse trees for “t ∧ t ∨ f”.

$E \to t \mid f \mid E \wedge E \mid E \vee E$

13. *(3 point 3 min)* Is the language generated by the grammar in Q12 regular? If yes, give a rex/dfa; if no prove using pumping lemma

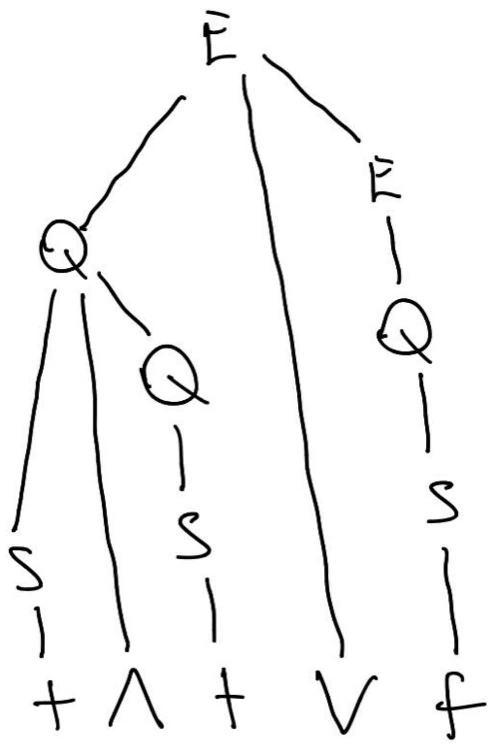
Strings accepted by language:
"t", "f", "t ∧ f", "t ∨ f", "t ∧ t ∨ f"
This language is regular
DFA:



14. *(7 point 7 min)* Assuming $\wedge$ has precedence over $\vee$, and both operators associate to the right, give an unambiguous grammar for Boolean expressions over t, f, $\wedge$ and $\vee$,

Nonterminals = {E, Q, S};
Terminals = {t, f, $\wedge$, $\vee$ }.
Production rules are:
E → Q
E → Q $\vee$ E
Q → S
Q → S $\wedge$ Q
F → t|f

15. *(3 point 3 min)* For the grammar you gave in Q14, give all possible parse trees for "t $\wedge$ t $\vee$ f"

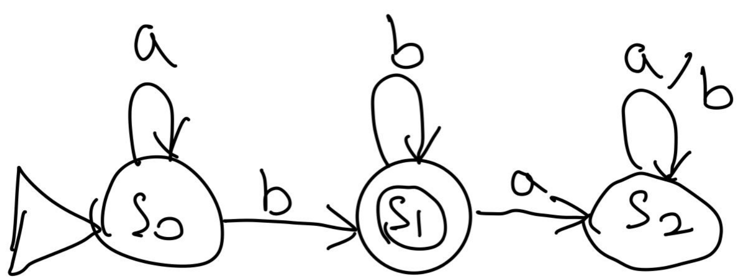


16. *(2 point 2 min)* How would you modify your answer for Q13 for the language generated by the grammar you gave in Q14?
    I have to extend the DFA to account for precedence.

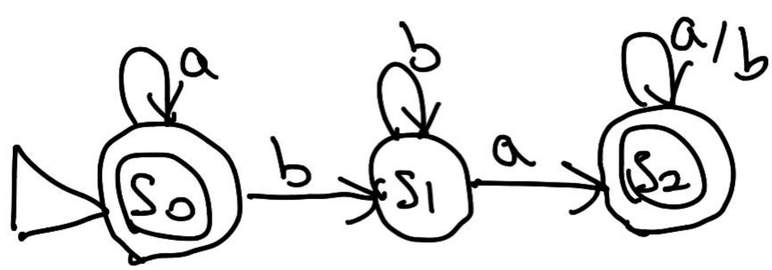
17. *(3 point 3 min)* Given a DFA that accepts a language L, how will you modify it to accept ~L? Why can the same method not be used with a PDA?

    To accept ~L you'd simply have to change the non-accepting states to accept states and the accepting states to non-accepting states.
    This wouldn't work with a PDA because a PDA implements a stack.

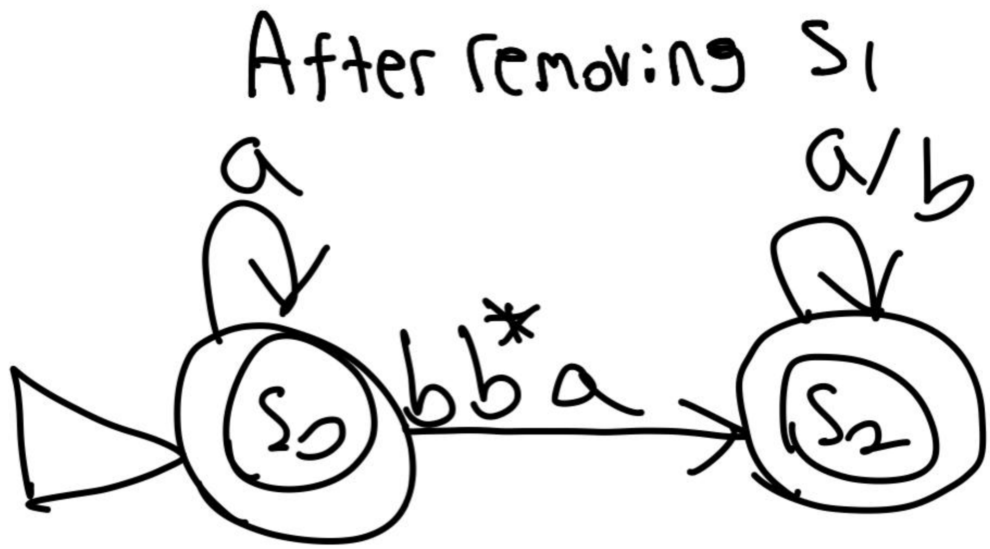
18. *(7 point 7 min)*
    (a)Construct a DFA for a*b*.



(b) Then use the method you gave in Q17 to get a DFA that accepts the complement of this language.



(c) Find the REX corresponding to this DFA using state elimination.



REX = a*| a*bb*aa*b*a*

19. *(10 point 10 min)* Run the CYK algorithm for the following grammar using input "aaabbb"
S □ AT | AU | epsilon
T □ UB | b
U □ AT | UT
A □ a
B □ b

S,U
(1,6)

S	S,T,U
(1,5)	(2,6)

-	S,U	T,U
(1,4)	(2,5)	(3,6)

-	S	T,U	-
(1,3)	(2,4)	(3,5)	(4,6)

-	-	S,U	-	-
(1,2)	(2,3)	(3,4)	(4,5)	(5,6)

A	A	A	B,T	B,T	B,T
a	a	a	b	b	b

20. *(5 point 5 min)* Write a Turing machine that gets a binary number as an input, and if the number is even adds one to it, and if the number is odd, subtracts one from it. You would need just a one cursor, one tape, one way tape machine; give the full machine, not just the algorithm

    It's easy to see if a binary number is even or odd just by looking at its last digit, our algorithm will simply have to read through all the digits in the binary number and stop when it finds the last digit. If the digit is zero we'd have an even number, if the digit is one we'd have an odd number. Subtraction and addition would come simply from changing that last number.

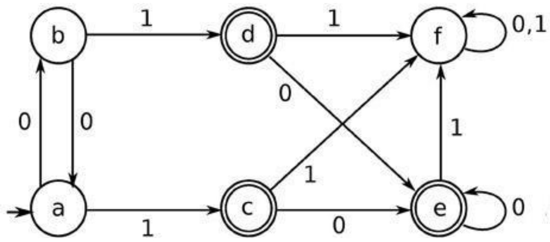
    Algorithm:
    At the initial state, if 0 is placed don't change statem hold on to 0 and go to the next number.
    At the initial state, if 1 is placed don't change state and keep 1 then go to the next number.
    At the initial state, if an empty symbol comes (representing the end of the binary number) change the state and go back to that final number.
    At the second state, if 1 is placed go to accept state and change 1 to 0
    At the second state, if 0 is placed change the state to accept state and change 0 to 1.

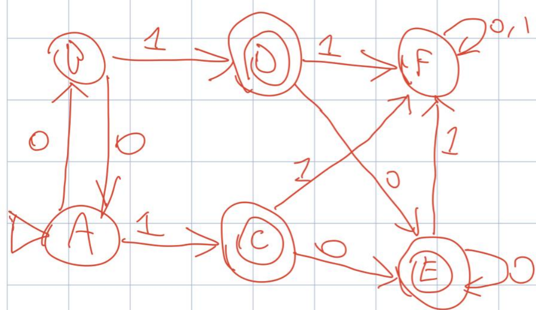
21. (extra credit) *(5 point 5 min)* If the answer to this question (Q21) were to be randomly guessed, what is the probability that the correct answer will be guessed?
    a. 1/3
    b. 1/3
    c. 2/3

    Explain your answer

    The probability would be ½ on the first guess, as there are as many evens as there are odds

22. *(10 minutes 10 points)* Minimize the following DFA





			acc	acc	acc	
	A	B	C	D	E	F
A			X	X	X	x(1)
B			X	X	X	x(1)
accept C	X	X				X
accept D	x	X				X
accept E	x	X				X
F	x'	x'	X	X	X	

Transition 1
First

$C \xleftarrow{1} A \Rightarrow B$

$D \xleftarrow{1} B \Rightarrow A$

$A \xrightarrow{1} C$ different

$F \xrightarrow{1} F$