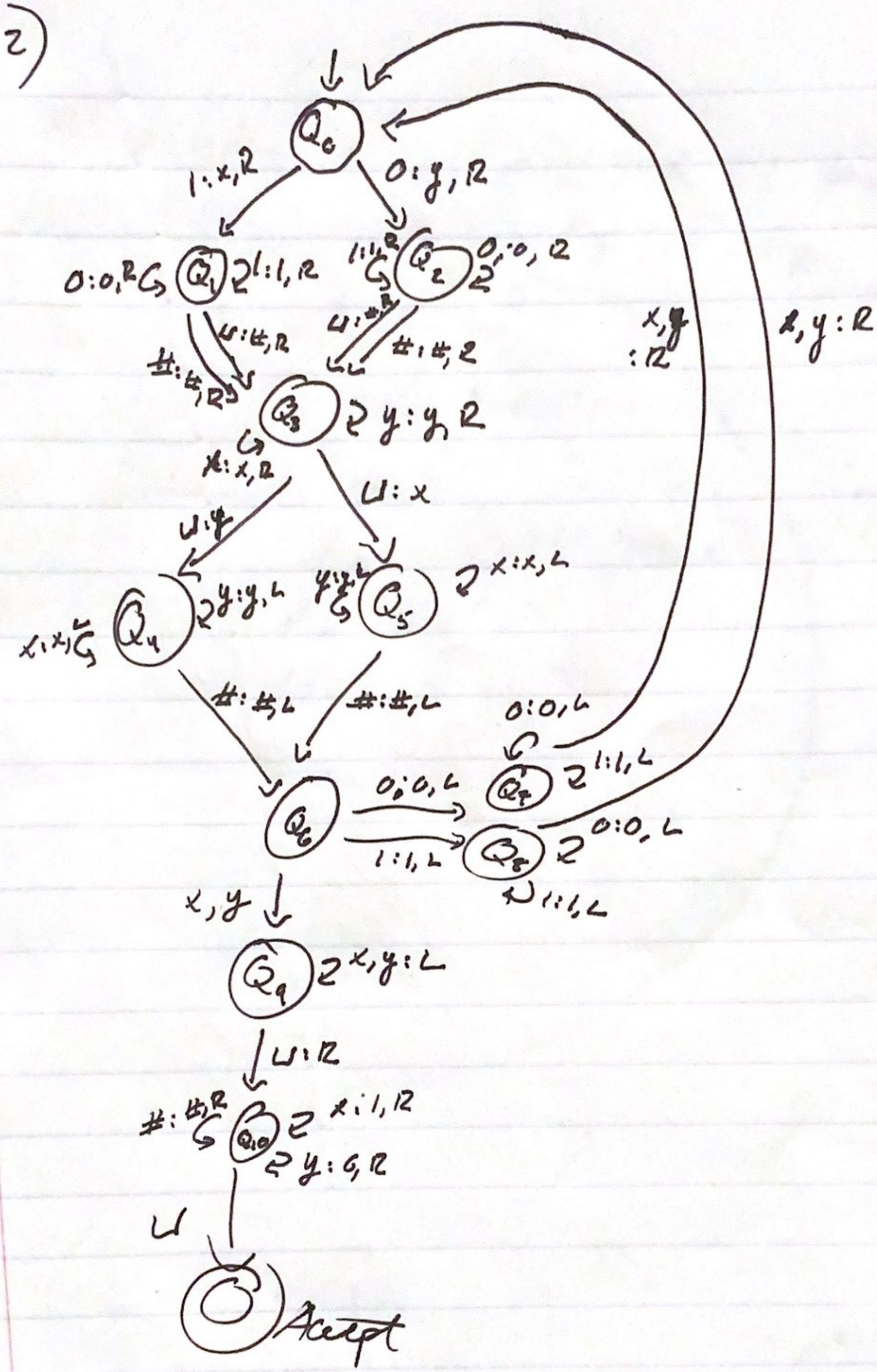
1) Regular language L recognized by $^{\text{de}}$finite state automata.

$$X_{DFA} = \{\langle M, w\rangle \mid M \text{ DFA that recognizes } w\}$$

- This DFA is ~~recognized by~~ recognizes L, which has set of strings w. By proof in class $\langle M, w\rangle$ is decidable & if X_{DFA} is decidable from this proof, all w is decidable too in L. So L is decidable.

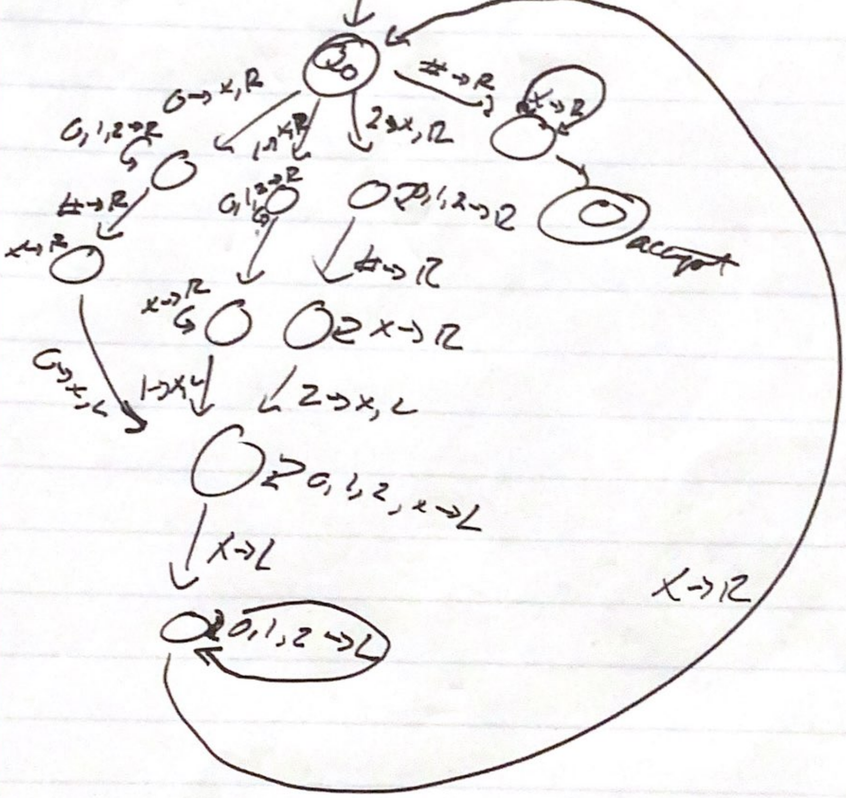
(Q2)



Q0
1:x,R
0:y,R
Q1
0:0,R
1:1,R
Q2
1:1,R
0:0,R
⊔:#,R
#:#,R
⊔:#,R
#:#,R
Q3
y:y,R
x:x,R
⊔:y
⊔:x
Q4
x:x,L
y:y,L
Q5
y:y,L
x:x,L
#:#,L
#:#,L
Q6
0:0,L
1:1,L
Q7
0:0,L
1:1,L
Q8
0:0,L
1:1,L
x,y:R
x,y:R
x,y
Q9
x,y:L
⊔:R
Q10
#:#,R
x:1,R
y:0,R
⊔
Accept

Q3)

Used ideas from slide 24, basically
Same concept w/ 2 added set of string



4) A set of elements/language/strings are not countable if it cannot be mapped to the set of all natural numbers.

Proof by Contradiction

If set of countable binary strings is ~~countable~~ infinite & countable, denoted by S, then we can create a mapping from countable set $A_\infty \{S_0, S_1, S_2 \text{---} S_n\}$ to natural #.
Where A = # on diagonal.

S_0	$\boxed{1}$	0	1	1	0	...
S_1	0	$\boxed{1}$	1	0	1	...
S_2	1	0	$\boxed{0}$	0	0	...
~~S_3~~	⋮					

For this to be countable, A must be in the list of binary strings as well as A^c. K^{th} bit in A should be K^{th} bit of string @ K. But, A^c can't be in A_∞ b/c A^c will never show up b/c it goes against K^{th} bit of string. Therefore B_∞ cannot be countable.

5) let C be a turing machine that holds DFA A & regex R
$\Rightarrow C = \{ \langle A, R \rangle \mid A = R \}$. Since regex can be represented as DFA & we Compare 2 DFA in the turing machine, from class, we know that C is decidable. Let L_R denote language represented by R & if L_R is regular, then exists a DFA A that is also regular & accepts L_R. If we convert R to a DFA D_R & compare $L(D_R)$ to $L(A)$ in a turing machine, then you should be able to prove that A = R = decidable.

~~$C = \langle L(D_R), L(A) \rangle$~~

$C = \{ \langle D_R, A \rangle \mid L(D_R) = L(A) \}$