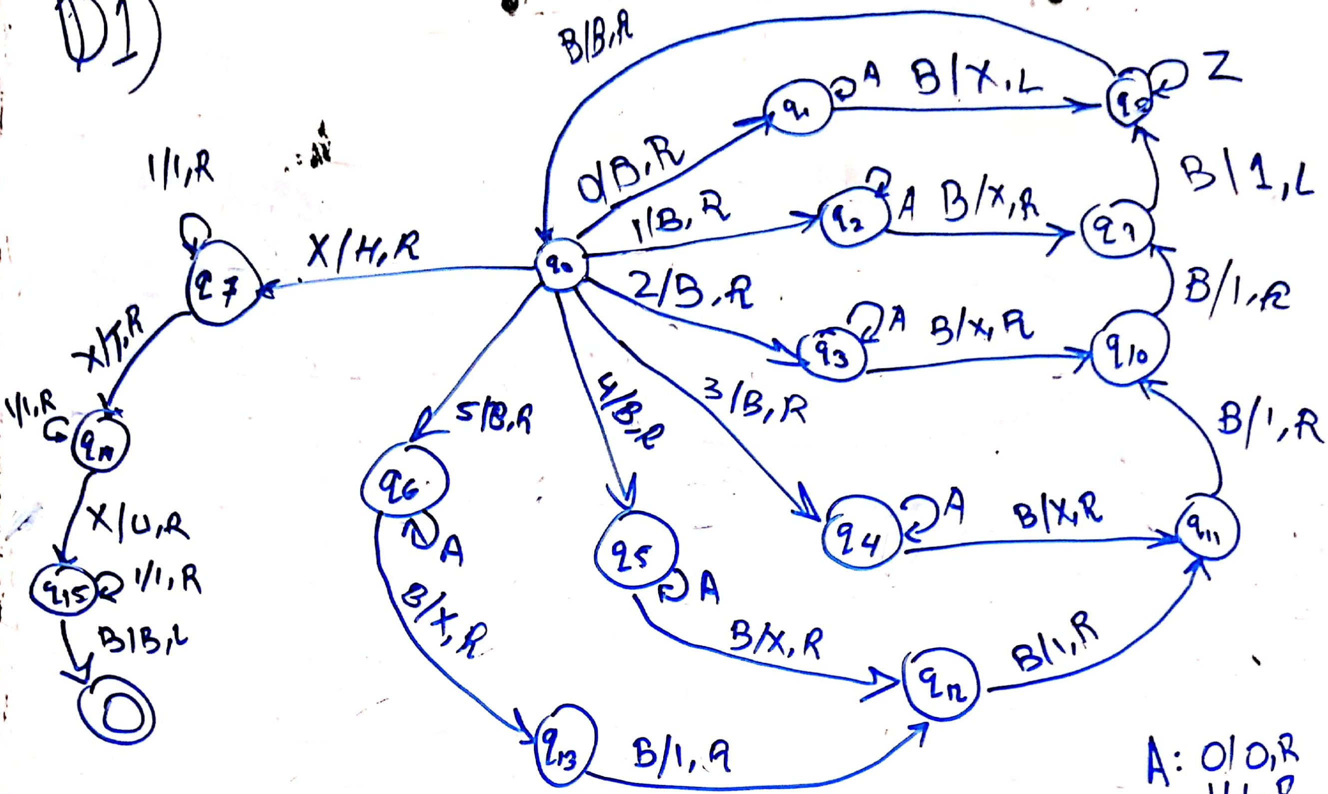
D1)



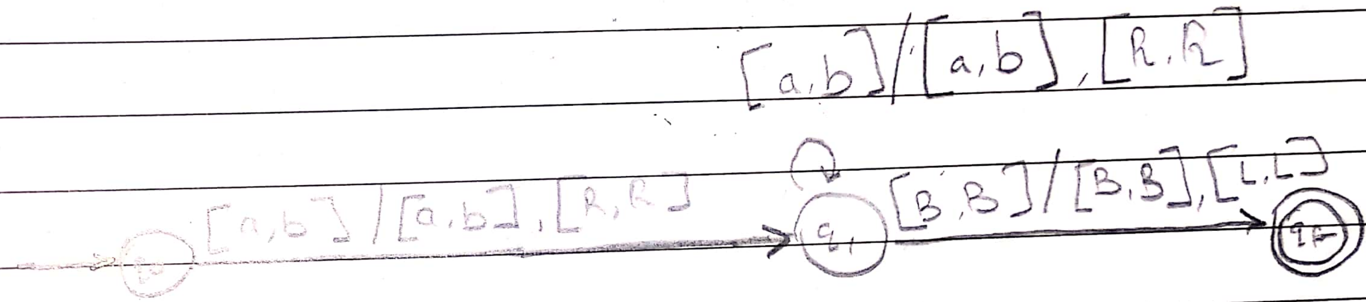
A: 0/0,R
1/1,R
2/2,R
3/3,R
4/4,R
5/5,R
X/X,R

Z: 0/0,L
1/1,L
2/2,L
3/3,L
4/4,L
5/5,L
X/X,L

Q2) variation of turing machine

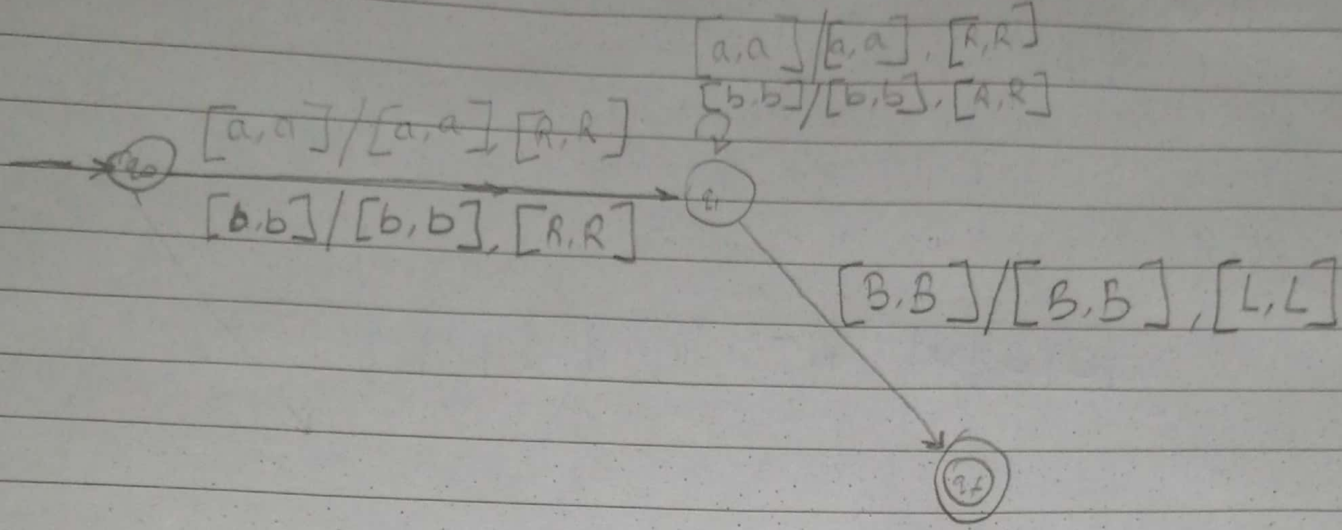
1) $a^n b^n \quad n \geq 1$ Ex string aaabbb

a	B	a	a	a	B
b	B	b	b	b	B



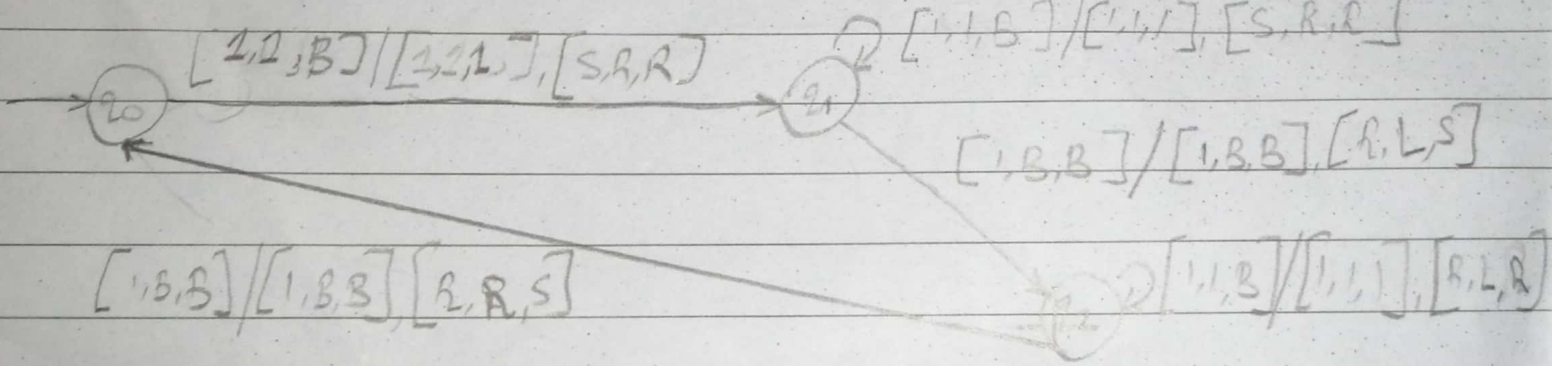
2) wcw $\{a,b\}$ Ex string abbacabba

w	B	a	b	b	a	B
w	B	a	b	b	a	B



3) m^2 $m >= 1$ Ex input $m = 3$

m	B	1	1	1	B	B			
m	B	1	1	1	B	B			
m^2	B	B	B	B	B	B	B	B	B



Scanned with CamScanner

Q3) Recursive and Recursive enumerable properties

★ Recursive

•) ~~Concatenation~~

↳ Suppose that L1 & L2 are recursive languages. Then there is a ~~t~~ machine m1 which accepts any input which is part of L1 and ~~a machine m2~~ rejects others, same for L2, a machine m2 exists. Input for new language is w.

↳ Concatenation: ~~To~~ We first non deterministically guess where to divide our input w into two parts, $w_1 w_2$. Then we run M_1 on w_1 and M_2 on w_2. We accept ~~and reject~~ if both accept or reject otherwise.
∴ Close under Concatenation

↳ Kleene's Closure: If we want to determine if $w \in L_1^*$. If w is empty, we accept, otherwise we guess in how many parts we need to break our input. Then we guess where to put all our ~~input~~ breaks in our input, and run M_1 on each part. We accept if M_1 accepts all parts
∴ Close under Kleene's Closure

↳ Homomorphism: Recursive languages are not closed under homomorphism. Proof:- We will show rL L and homomorphism h such that h(L) is undecideable

•) let $L = \{ xy \mid x \in \{0,1\}^*, y \in \{a,b\}^*, x = \langle H, w \rangle$ and y encodes an integer n such that the TM M ~~will halt~~ on input w will halt in n steps$\}$

•) L is decideable : Can simulate M on input w for n steps

•) Consider homomorphism $h : h(0) = 0, h(1) = 1, h(a) = h(b) = e$

•) h(L) = Halt which is undecidable

Date ______

Thm ↳ Inverse homomorphism: RL are closed under Inverse homomorphism
Proof:- Given TM M_1 that decides L_1, a TM to decide $h^{-1}(L_1)$ is on input x, compute $h(1)$ and run M_1 on $h(x)$ accepts if and only if M_1 accepts

★ Recursively enumerable:-
Suppose M_1 & M_2 accept L_1 & L_2 respectively w is our new language

↳ Concatenation: If $w \in L_1 L_2$, we guess where to break into $w_1 w_2$ and run on M_1 & M_2 respectively. If w is in the language then both M_1 & M_2 will accept. RE is closed under Concatenation.

↳ Kleene's Closure: To determine if $w \in L^*$, we guess in how many parts we need to break our input and where to put our breaks. Then we run on each part. If w is the language then each of these will accept. Thus RE is closed under Kleene's Closure

↳ Union and Intersection:- RE language are closed under both union & Intersection.
Proof:- given TMs M_1 & M_2 that recognize L_1 & L_2
•) A TM that recognizes $L_1 \cup L_2$: on input x, run M_1 & M_2 on x in parallel, and accept if and only if either accepts. (Similarly for intersection but no need for parallel simulation.)