Hw6

1) In a Boolean algebra, ~~every element x has~~

Suppose x and y are two inverse elements of x such that

$x+x=1$ and $x \cdot x = 0$

and $x+y=1$ and $x \cdot y = 0$

To prove prove $x=y$

consider $y = y \cdot 1$ — Identity law given

$= y \cdot (x+x)$ — $1 = x+x$

$= yox + y \cdot x$ — Districtive law given $y \cdot x = 0$

$= 0 + yox$ — Identity law given $x \cdot x = 0$

$= yox$ — Commutative law

$= yox + 0$ — Districtive law

$= y \cdot x + x \cdot x$ — given, $x+y=1$

$= x \cdot y + x \cdot x$

$= x \cdot (y+x)$

$= x \cdot 1$

$= x$

$\Rightarrow y = x$ Inverse element is unique.

2) Not Gate Using NAND Gat

(x) out

(x)

(y) out

AND Gate using N

(x)

(y)

OR

3) $\frac{\bar{w}\bar{x}\bar{y}\bar{z}}{1} + \frac{\bar{w}\bar{x}\bar{y}z}{3} + \frac{\bar{w}\bar{x}y\bar{z}}{4} +$

$+ \frac{\bar{w}xy\bar{z}}{4} + \frac{\bar{w}\bar{x}\bar{y}z}{12}$

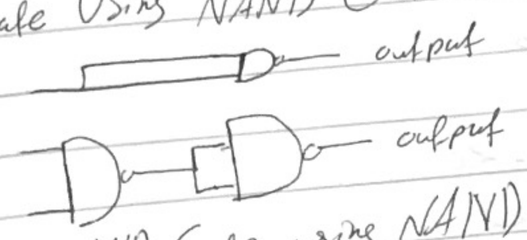
here $F(w,x,y,z) = \Sigma m(1,3,$

$F = \bar{x}_2 + \bar{x}_4$

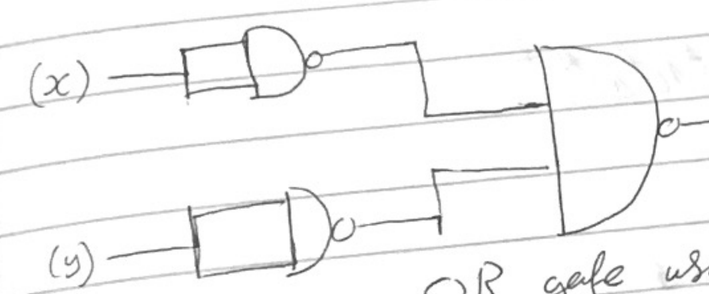
elements of
Identity law given
1. $x+x$
Distributive law given $y \cdot x = 0$
Identity law given $x \cdot x = 0$
Commutative law
Distributive law
given, $x+y=1$

element is unique.

2) Not Gate Using NAND Gate



AND Gate using NAND gate



OR gate using NAND gate

3) $\frac{\bar{w}\bar{x}\bar{y}z}{1} + \frac{\bar{w}x\bar{y}\bar{z}}{3} + \frac{\bar{w}xy\bar{z}}{4} + \frac{\bar{w}xyz}{6} + \frac{w\bar{x}\bar{y}\bar{z}}{12} + \frac{w\bar{x}\bar{y}z}{14} + \frac{w\bar{x}y\bar{z}}{9} + \frac{w\bar{x}yz}{11}$

here $F(w,x,y,z) = \sum m(1,3,4,6,9,11,12,14)$

$F = \bar{x}_2 + \bar{x}_1$

4) $V^-WX^-Y^-Z = 00100_2 = 4$

$V^-WXY^-Z = 00110_2 = 6$

$VWX^-YZ = 01101_2 = 13$

$V^-WX^-YZ = 10100_2 = 20$

$V^-WXY^-Z = 10110_2 = 22$

$VWX^-YZ = 11101_2 = 29$

5) $R.E = 0^*10^*(0^*10^*10^*)^*$

Let's take

1000000000100000001

11000101110010000000 and all such strings that contain odd number of 1's in the string.

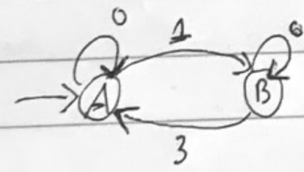
Regular Expression for all length 4's defined over $\{a,b\}$

$RE \Rightarrow b + a(a\ b^*a)^*$

valid strings: a, b, aaa, aaaaa, abaa and many more similar strings

6) All Binary String containing an odd number of 1's

$L = \{1, 01, 10, 111, 1011, 1101\}$

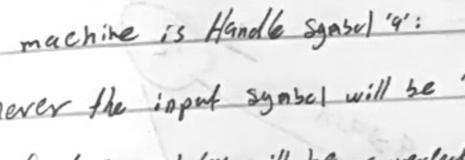


RE:

$0^*10^*(0^*10^*10^*)^*$

6) DFA: all binary strings divisible

$L = \{0, 11, 110, 1001, 1100$



7) $S = 1m \Rightarrow SS^* \Rightarrow SS+S^* \Rightarrow a$

$S = rm \Rightarrow SS^* \Rightarrow Sa^* \Rightarrow SS+$

It is Unambiguous

The set of all postfix expressions multiplication.

8) The given function for which we have t

where $w \in \{a,b\}^*$ and w^R is the reverse

Turing machine is Handle symbol 'q':

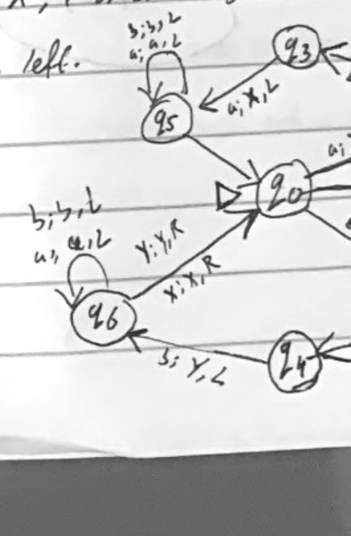
Whenever the input symbol will be

In it first symbol 'q' will be converted

and we will go to state q1. Now,

right until X, Y or blank symbol come

move to the left.



ch strings that contain

defined over {a, s}

a and many more similis
strings

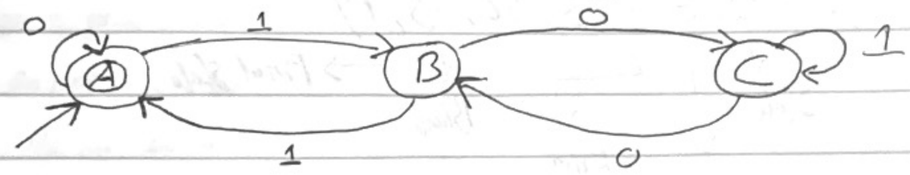
mber of 1's
= {1, 01, 10, 111, 1011, 1101}

:

$0^*10^*(0^*10^*10^*)^*$

6) DFA: all binary strings divisible by 3

$L = \{0, 11, 110, 1001, 1100 \ldots\ldots\}$



7) $S = 1m \Rightarrow SS^* \Rightarrow SS + S^* \Rightarrow aS + S^* \Rightarrow aa + S^* \Rightarrow aa + a^*$

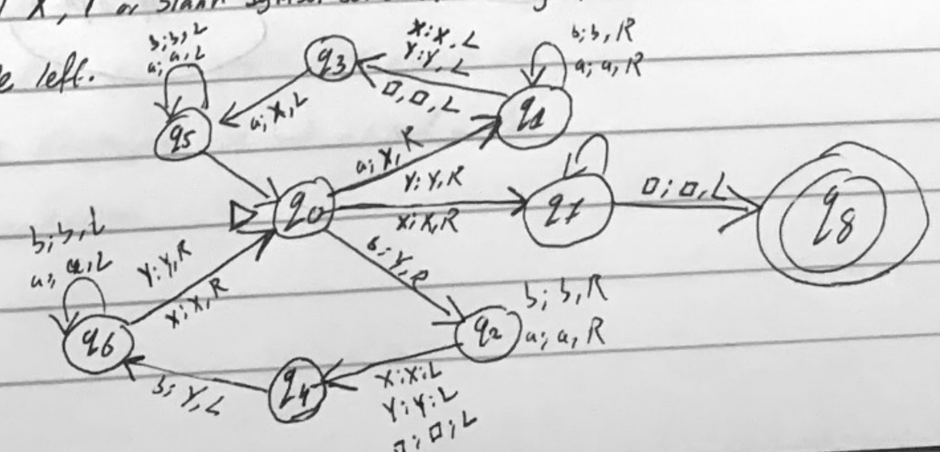
$S = 1m \Rightarrow SS^* \Rightarrow Sa^* \Rightarrow SS + a^* \Rightarrow Sa + a^* \Rightarrow Sa + a^* \Rightarrow aa + a^*$

It is Unambiguous

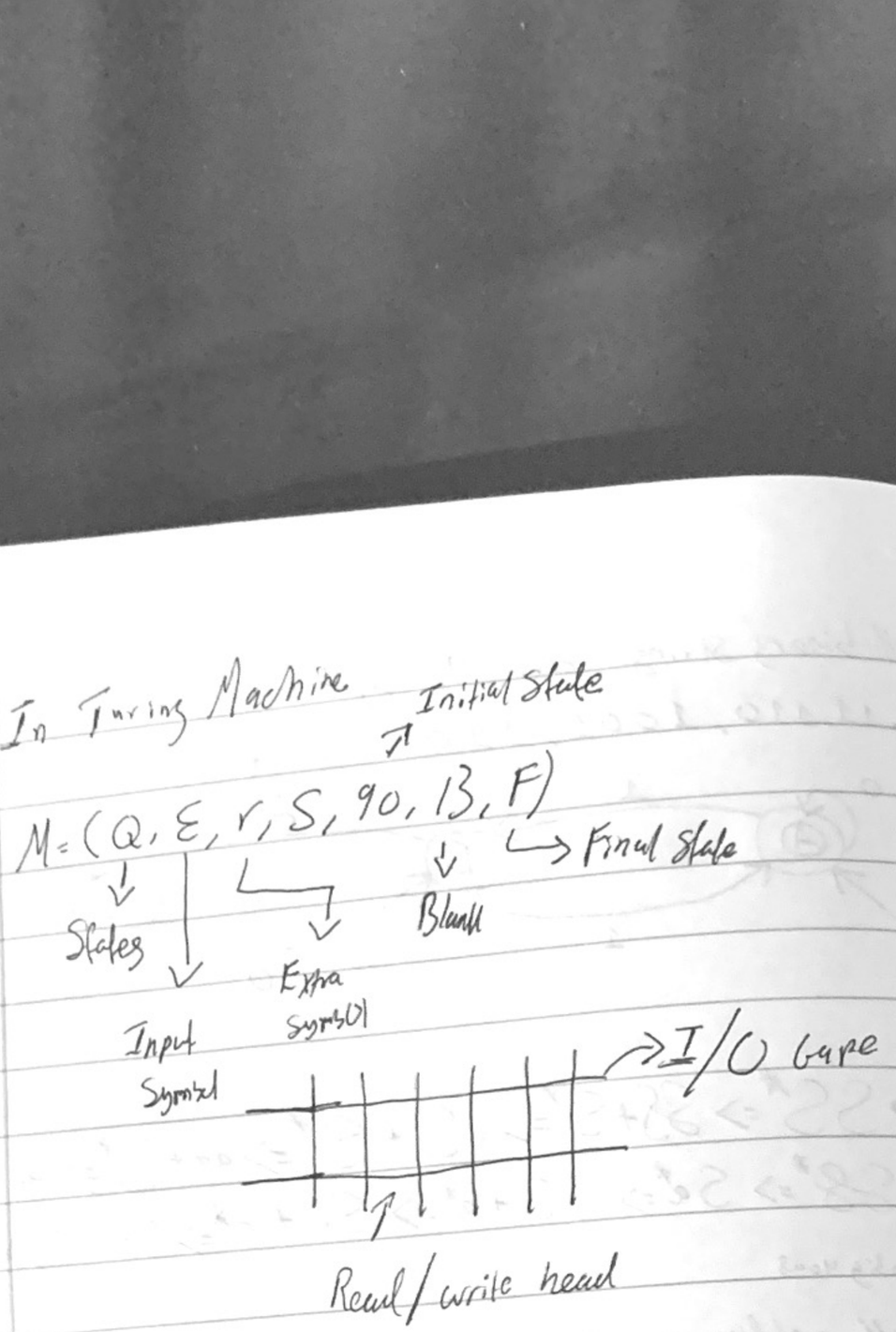
The set of all postfix expressions consist of addition and multiplication.

8) The given function for which we have to design the turing machine if $f(w) = ww^R$, where $w \in \{a,b\}^*$ and w^R is the reverse of w. Steps to design the required Turing machine is Handle symbol 'a':

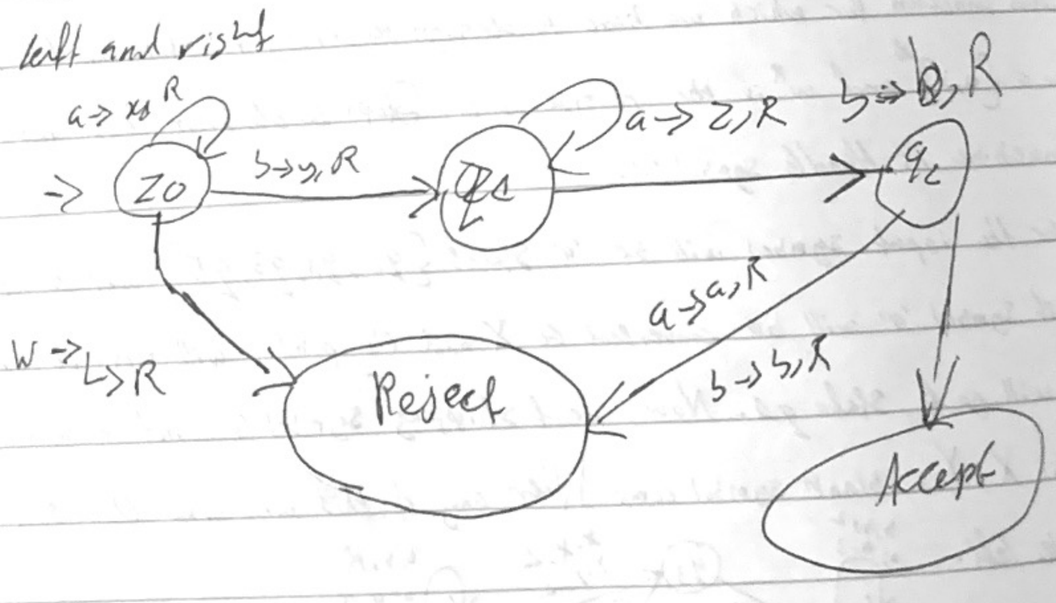
Whenever the input symbol will be 'a' states $\{q0, q1, q3, q5\}$ will handle it. In it first symbol 'a' will be converted to X and the pointer will move to right and we will go to state q1. Now, start skipping symbol 'a' and 'b' and move right until X, Y or blank symbol come. When any of this will come then skip it and move to the left.



9) In Turing Machine



-> Read and write both are possible and read/write head moves in both possible and read/write head moves in both directions left and right



10) Prove that Recursive Languages are cl
-> Let M_0 = TM for L_1, $U L_2$
-> Let M_U construction
1) Make 2-tapes and copy input W on
2) Simulate M_1 on tape 1
3) Simulate M_2 on tape 2
4) If either M_1 or M_2 accepts, then M_U
5) Otherwise M_U rejects

Prove that recursive language
-> Let M_n = TM for $L, n/2$
-> Let M_n Construction
1) Make 2-tapes and copy input W on
2) Simulate M_1 on tape 1
3) Simulate M_2 on tape 2
4) If M_1 and M_2 accepts, then M_n
5) Otherwise M_n rejects.

Recursive languages are also clos
- Concalention
- Kleen closure (Star operator)
- Homomophism and inverse Homom

Recursive Languages are not clo
- Complementation.

Initial State

B, F)

↳ Final State

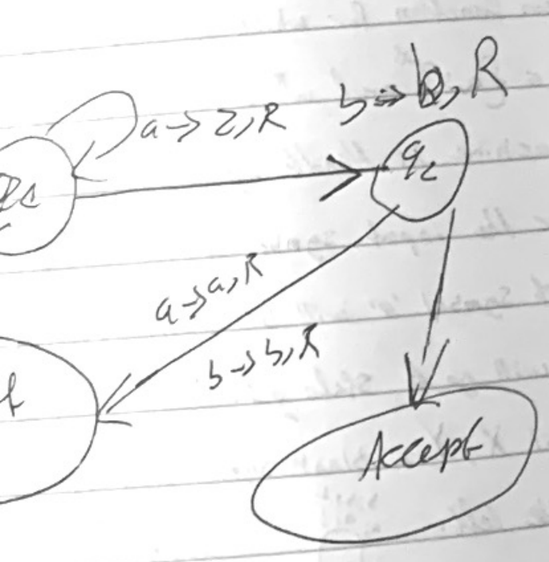
Blank

→ I/O tape

write head

are possible and read/write head

d read/write head moves in both directions



10) Prove that Recrusive Languages are closed under Union

-> Let M_U = TM for $L_1 \cup L_2$

-> Let M_U construction

1) Make 2-tapes and copy input W on both tapes
2) Simulate M_1 on tape 1
3) Simulate M_2 on tape 2
4) If either M_1 or M_2 accepts, then M_U accepts
5) Otherwise M_U reiects

Prove that recursive language are closed under intersection

-> Let M_n = TM for $L_1 \cap L_2$

-> Let M_n construction

1) Make 2-tapes and copy input W on both sides
2) Simulate M_1 on tape 1
3) Simulate M_2 on tape 2
4) If M_1 and M_2 accepts, then M_n accepts
5) Otherwise M_n rejects.

Recursive languages are also closed under:

- Concatention
- Kleen closure (Star operator)
- Homomuphism and inverse Homomorphism

Recursive Languages are not closed under

- Complementation.