Practice Sheet - 6

Q.1 what is Ackermann's function, calculate $A(1,1)$ $A(1,2)$ $A(2,1)$.

Sol^n: (1) In computability theory, the Ackermann function, named after Wilhelm Ackermann, is one of the simplest and earliest-discovered examples of a total computable function that is not primitive recursive.

(2) All primitive reccursive fuction are total & computable, but the Ackermann f^n illustrates that not all total computable functions are primitive recursive.

(3) The two argument Ackermann peter function is defined as follows for non-negative integers m & n:

$$A(m,n) = \begin{cases} n+1 & \text{.... if } m=0 \\ A(m-1,1) & \text{.... if } m>0 \ \& \ n=0 \\ A(m-1, A(m,n-1)) & \text{.... if } m>0 \ \& \ n>0 \end{cases}$$

$$\begin{aligned} A(1,1) &= A(0, A(1,0)) \\ &= A(0, A(0,1)) \\ &= A(0, 2) \end{aligned}$$

$$\boxed{A(1,1) = 3}$$

$$\begin{aligned} A(1,2) &= A(0, A(1,1)) \\ &= A(0, 3) \quad \text{.... from above } A(1,1)=3 \end{aligned}$$

$$\boxed{A(1,2) = 4}$$

Note:- (Do not use shortcut in exam ☺)

$$
\begin{aligned}
A(2,1) &= A(1, A(2,0)) \\
&= A(1, A(1,1)) \\
&= A(1,3) \quad \text{--- from } A(1,1)=3 \\
&= A(0, A(1,2)) \\
&= A(0,4) \quad \text{--- from } A(1,2)=4
\end{aligned}
$$

$$\boxed{A(2,1) = 5}$$

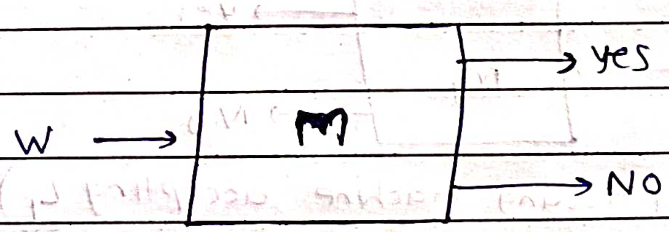
Q.2 Explain the properties of Recursively enumerable language. Give relation between recursive & recursive enumerable language.

Soln:- Properties of Recursive & Recursively enumerable languages :-

① complement of recursive language is always recursi[ve]

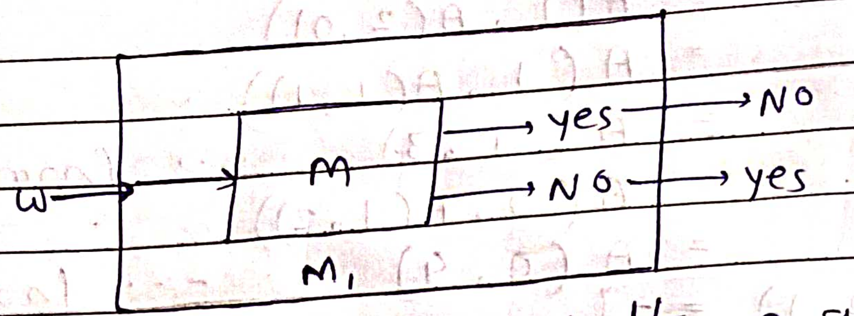
Proof :-

If L is recursive language, then there exist Turing machine M, accepting L & when halting is guaranteed.



we can construct a turing m/c M_1 such that M_1 simulates M on w and if M accept w then M_1 reject w & if M reject w then M_1 accepts w as show in fig.

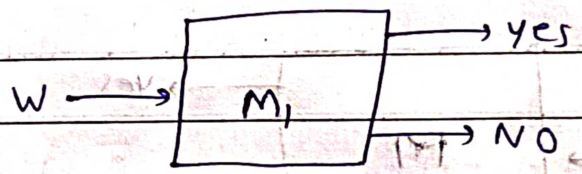
(2)



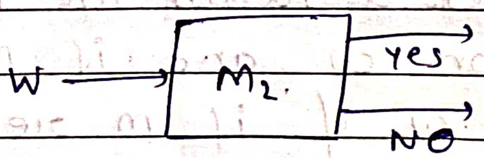
since, M_1 is guarteed to halt on all possible input & accepts complement of L, proving that complement of a recursive language is always recursive language.

(2) a) Union of two recursive language is recursive
b) Union of two recursively enumerable languages is recursive language.

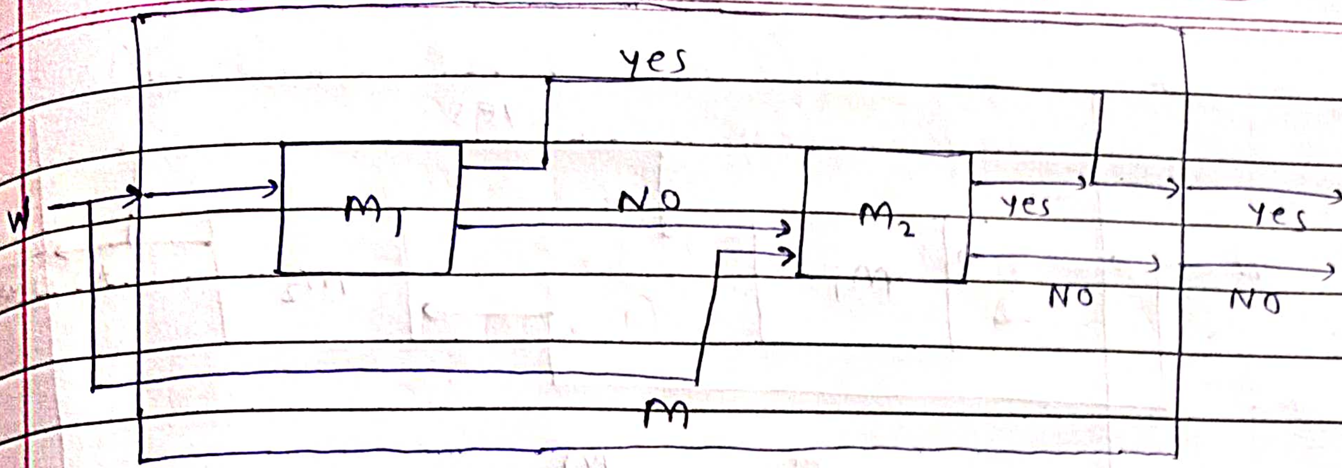
Proof :-
(a) If L_1 & L_2 are recursive languages then there exist turing machine M_1 & M_2 accepting L_1 & L_2 respectively and they are guaranteed to halt on all possible input.



(Turing machine accepting L_1)

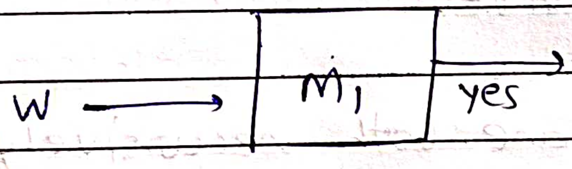


(Turing machine accepting L_2)

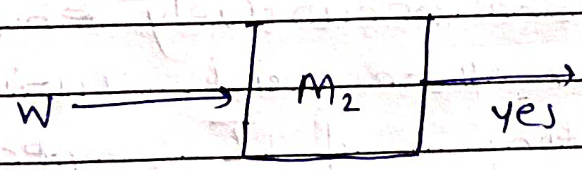


from above constsuction it is clear that, M accept ω if and only if either it acceptable to M_1 or acceptable to M_2, thereby accepting $L_1 \cup L_2$ and M is guaranteed to halt, thereby proving that $L_1 \cup L_2$ is recursive.

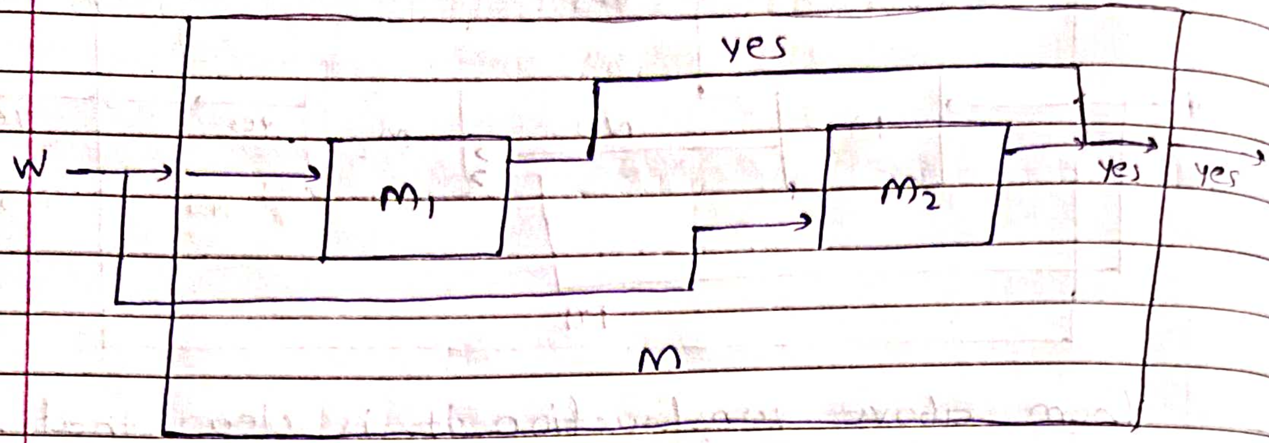
(b) If L_1 & L_2 are two recursively enumerable language, then there exists turing m/c M_1 & M_2 accepting L_1 & L_2 respectively.



(Turing m/c accepting L_1)



(Turing m/c accepting L_2)

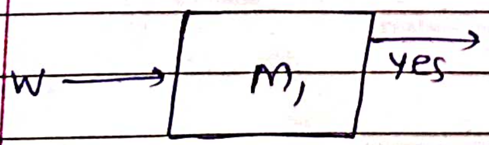


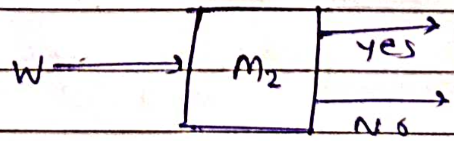
from the above construction it is clear that, M accepts ω if and only if either it acceptable to M or acceptable to M_2, thereby accepting $L_1 \cup L_2$, but if ω is not acceptable to both M_1 & M_2 then there is no guarantee of halt, thereby proved that $L_1 \cup L_2$ is recursively enumerable & not recursive.

③ If L & $\bar{L}$ are both recursively enumerable language, then both L and $\bar{L}$ are recursive.

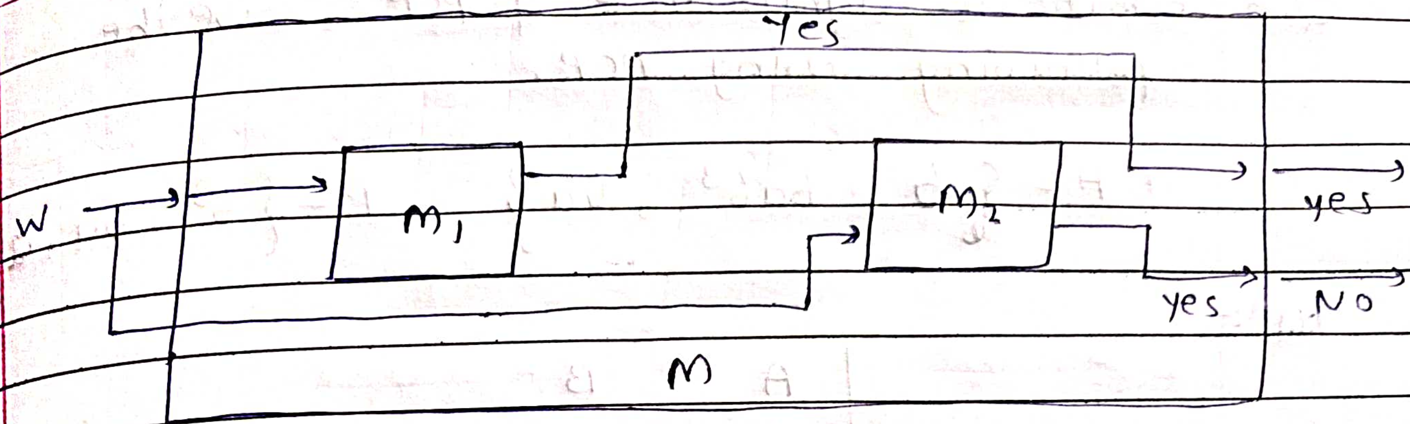
Proof :-

If L & $\bar{L}$ are recursively enumerable language then there exist Turing m/c M_1 & M_2 accepting L & $\bar{L}$ respectively.





(Turing m/c accepting L) (Turing m/c accepting $\bar{L}$)



from the above construction it is clear that, if M_1 accepts w, then M_1 will halts and M_2 will enter into accepts halt, if M_1 accepts w & then M_2 will halt and M will enters into its reject halt.

Therefore M is turing m/c which is guaranteed to halt, and accepts members of L, and rejects members of $\bar{L}$. Thereby proving that there exist a turing m/c accepting L. hence proving that L is recursive.

refer Q.4

Q.3 what is significance of PCP, solve the following using PCP.

$A = \{b, bab^3, ba\}$, $B = \{b^3, ba, a\}$

Soln:-

	A	B
1	b	b^3
2	bab^3	ba
3	ba	a

Step ① :- 2

$A = bab^3$

$B = ba$

Step ② :- 21

$A = bab^3b$

$B = bab^3$

Step ③ :- 211

$A = bab^3bb$

$B = bab^3b^3$

Step ④ :- 2113

$A = bab^3bbba$

$B = bab^3b^3a$

∴ Solution for the given PCP is 2113.

Q.4 Explain PCP. Consider the post correspondence system described by the following lists.

$A = \{10, 01, 0, 100, 1\}$

$B = \{101, 100, 10, 0, 010\}$

Does this PCS have a resolution?

Solⁿ:- ① **PCP :-** Post correspondence problem is a popular undecidable problem that was introduced by Emil Leon Post in 1946.

② It is simpler than halting problem.

③ In this problem we have N number of Dominos (tiles). The aim is to arrange tiles is such order that string made by Numerators is same as string made by denominators.

④ PCP can represented in two ways :-

ⓐ Domino's form :-

B	A
CA	AB

ⓑ Table form :-

	Numerator	Denominator
1	B	CA
2	A	AB
3	CA	A
4	ABC	C

	A	B
1	10	101
2	01	100
3	0	10
4	100	0
5	1	010

step (1) :- 1

A = 10

B = 101

step (2) :- 14

A = 10100

B = 1010

step (3) :- 145

A = 101001

B = 1010010

step (4) :- 1453

A = 1010010

B = 101001010

step (5) :- 14534

A = 1010010100

B = 1010010100

soln for given PCP is 14534

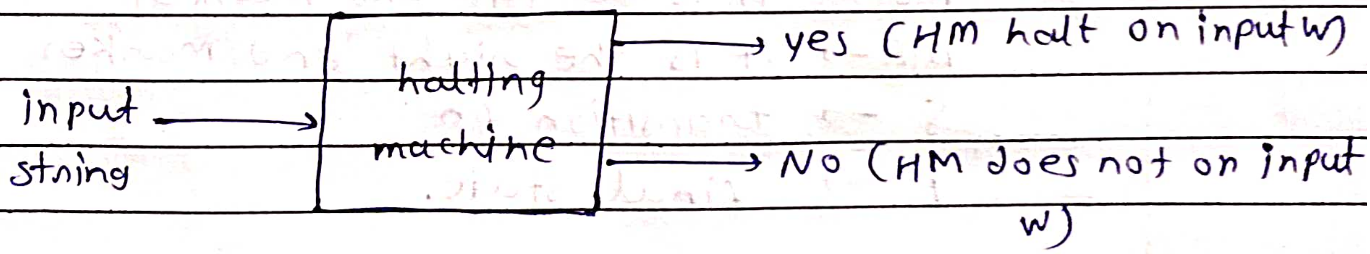
Q.5 ~~Does this PCP have a resoluti~~

Q.5 write short Note on
(i) halting problem of turing m/c
(ii) Linear bounded Automata
(iii) Primitive Recursive function

Solⁿ:- (i)

(1) Input :- A TM and an input string w.
(2) Problem :- Does the TM m/c finish computing of the string w in a finite no. of steps? The answer must be either yes or no.
(3) we will call this TM as a halting machine that produce a "yes" or "no" in a finite amount of time.
(4) If the halting m/c finishes in a finite amount of time, the output comes as "yes", otherwise as 'no'.

(5) Block diagram of halting m/c :-



(6) The halting problem is undecidable.

(ii) Linear bounded automata :-

① A linear bounded automata is a multi-track non-deterministic TM with a tape of some bounded finite length.

② The computation is restricted to the constant bounded area. The input alphabet contains two special symbols which serve as left end markers & right end markers which means the transitions neither move to the left of the left end marker nor to the right of right end marker of the tape.

③ A LBA can be defined as an 8-tuple

$(Q, X, \Sigma, q_0, ML, MR, \delta, F)$

$Q \rightarrow$ It is a finite set of states

$X \rightarrow$ it is a tape alphabet

$\Sigma \rightarrow$ it is a input symbol

$q_0 \rightarrow$ initial state

$ML \rightarrow$ it is the left end marker

$MR \rightarrow$ it is the right end marker

$\delta \rightarrow$ transition f^n

$F \rightarrow$ final state.

Date
Page

(iii) Primitive recursive f^n :-
A f^n is said to be primitive recursive if it can be obtained from the initial f^n by a finite no. of operations of composition & recursion.

ex.

Consider function $f(x,y) = x+y$ (addition f^n)
It is a primitive recursive f^n, because it can be defined by recursion & composition from initial f^n follow :-

$$f(x,0) = x = U_1^1(x)$$

&

$$f(x, y+1) = x+y+1 = f(x,y)+1$$
$$f(x, y+1) = S(U_3^3(x,y,f(x,y)))$$

Since U_1^1, U_3^3 and S are initial f^n $f(x,y)$ is primitive recursive.

Date
Page

Q.(6) write a short note on :-
(1) post correspondence problem. — (Q. 4)
(2) Primitive Recursive f^n ⟶ (Q. 5)

Q.7 Def^n Decidability & undecidability.

Sol^n:- It refers to getting an answer to the question, the problem wheather there exists an algorithm to solve a problem. If algorithm exists for solving problem is called decidablility. whereas when there exists no algo to solve proble, the proble is called undecidable / undecidability.

* Decidable language / language decidability :-
A language is called decidable or recursive if there is a TM which accepts and halts on every input string w. Every decidable languange is Turing acceptable.

* undecidable language :-
A language is called undecidable if there is no TM which accepts the language and makes a decision for every input string w.
A undecidable languages are not recursive language, but sometimes they may be recursively enumerable languages.

(refer Q.3)

Q.8 What is the significance of PCP, solve the following using PCP.

i	wi	xi
1	0	000
2	01000	01
3	01	1

Soln:- step ① :- i = 2

wi = 01000
xi = 01

step ② :- i = 21

wi = 010000
xi = 01000

step ③ :- i = 211

wi = 0100000
xi = 01000000

step ④ :- i = 2113

wi = 0100000001
xi = 0100000001

∴ soln for given PCP is 2113.

(refer Q.9)

Q.9 what is PCP & solve :-

Sr No.	List A	List B
1	01	011
2	1	10
3	1	11

(no soln found)

soln:-