-: Assignment 6 :-

C+

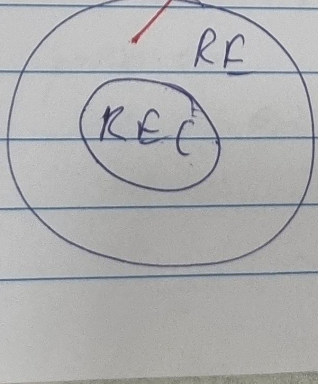
Q1). Write short note on recursive & Recursively enumerable language.

Ans: Recursively Enumerable (RE) or type 0 language

RE languages or type 0 language are generated by type 0 language are created by type 0 grammar. An RE language can be accepted or recognized by Turning machine which means it will enter into rejecting state for the string which are not part of the language. are also called as Turing countable lang.

Recursive language (REC).

A recursive language (subset of RE) can be decided by TM which mean it will enter into final state for the strings of language & rejecting state for the strings which are not part of the lang. eg: $L = \{ a^n b^n c^n \mid n >= 1 \}$ is recursive become we can construct a TM which will move to final state if the string is of the form $a^n b^n c^n$ else move to move to non-final state so the TM will always hal in this. The Relation betn RE & REC lang follows fig.



Closure Properties of Recursive language:

i) Union if L1 & If L2 are two recursive lag. then union L1 ∪ L2 will also be recursive because if TM halts for L1 & halts for L2 if will also halt for L1 ∪ L2

ii) Concatination if L1 & If L2 are two recursive language their concatation L1.L2 will also be recursive.

iii) Kleene Closure: If L1 is recursive its Kleene closure L1 * will also be recursive.

iv) Intersection & complement: If L1 & L2 are two recursive lang. their inter. L1 ∩ L2 will also be recursive.

What is Halting Problem? Explain in detail

The halting problem is determing whether a comp. prog. will eventually stop or run for ever creating a general algo. that can accurately predict this for all program is impossible Alan TM prof showed no way to solve this problem for all cases. An undecidable problem is a sort of computational problem requiring a yes/No answe but where no comp. prog. can give the proper ans. all of the time, that is, any possible algorithm or program would sometimes give the wrong ans. or run forever without providing any answer.

Proof:

i) Assume we can create a machine called (P, I) where HM is halting machine, P is the program, & I is the input. After seeing both i/p the machine HM will o/p whether or not the progm P terminates.

HM (P, I)

Yes (if the prog HALTS)

NO (if the prog does not HALT)

ii) Now, create an inverted machine TM that takes a prog. P as a i/p

IM(P)

if HM(P, P) == Yes → loop forever.

If HM(P, P) == NO → HALT

iii) Now, take a situation where the progm. TM is passed to the TM function as an input

6.5

TM (TM)

HM (TM, TM) == Yes (HALT)

loop forever.

HM (TM, TM) == No (NOTHALT)

It will never halt because of the above not halting condition.

It is impossible for the outer fun to halt if its inner func is in a loop (RHS) it is likewise impos. for outer fun. to halt even if its inner func. is halting (LHS).

Q3) Write detailed explanation on Rice theorem.

=> It states that any non-trivial semantic property of a lang. which is recognized by a turing machine is undecidable machines that satisfy that property.

formal Definition: If P is a lang of all Turing lang. hold the property L_P is recognized by TM M. $L_P = \{ \langle M \rangle \mid L(M) \in P \}$ is undecidable.

Proof: Suppose, a property P is non-trivial & $\emptyset \notin P$ since P is nontrivial at least one lang. satisfies P i.e. $(M_0) \in P$ $\exists$ Turing Machine M_0.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

A function that maps an instrce $M \times M =$ $\{ \langle M, w \rangle \mid M$ accepts input $w \}$ to a N such that

a) If M accept w & N accepts the same long as M_0, then $L(M) = L(M_0) \in P$.

b) If M does not accept w & N accept then $L(N) = \phi \in P$

Since ATM is undecible & it can be reduced to LP is also decidable.

Qn) Define post correspondence problem. Prove that PCP with two lists $x = \{b, bab^3, ba\}$ & $y = \{b^3, ba, a\}$ have a solution.

=> The post corresponding problem (PCP) is a problem in the field of Theoritical computer. Sci. & math. logic. It is decision problem that can be used to illution the concept of undecidability & it's closely related to the theory of formal lang. & automata. The problem is:

Given a finit set of domains: each with 2 strg can upper of a "lower" strg. written on it & a target strg can you find a sequence of domains choices from the set. such that when you concatenate the upperstrg it matches the concate noted lower strg.

$x = \{b, bab^3, ba\}$

$y = \{b^3, ba, a\}$.

6.7

1) b
2) babbb
3) ba

bbb
ba
a

i) find a domino where starting symb. are same.
∴ 2nd domino.
up : babbb   down : ba.

ii) lets match the no. of bs.
∴ pick first domino.

babbbb   babbb.

iii) To even out b's we pick first one again

babbbbb   babbbbbb

iv) No, upper part has one b lesser so we can pick the best domino.

babbbbbb a   babbbbbb a.

final sequence 2113.