## Automata Ilrory (cs 1.302) End Sem Exam

Abhinar 5 Menon 2020114001

PDA The Halting Problem for PDAS is decidable, while the Halting Problem for TMS is not. I hey do not whatch.

The HP for TMs can proved to be undecidable by reduction from the halting problem acceptance problem ATM; which we know to be undecidable by diagonalisation.

For PDAs, conversely, we have an algorithmic procedure to convert them to CFGs. Thus a TM for Hern can convert the PDA to a CFG in CNF & test for halting algorithmically.

2. (A) (RE, UREZ) is recursively enumerable.

http://wecan show this by dovetailing the recognisers

for RE, & REZ and accepting as soon as either

ene accepts. As there is no guarantee that both will

halt, we cannot have a decider.

(B) (R, nRE2) is recursively enumberable. We need both the decider for R, and the recogniser for RE2 to halt in order to give an answer. As the latter may not ever occur, we can only have a recognises.

(C) (R, -R2) is de recuesive.

As the deciders for R, & R2 both halt, we can rund from successively (there is no need of dovetailing) and return the XOR of their results, which is

3. We have  $Pre(L) = \frac{7}{2} \times e \times \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \times \frac$ 

Consider the DFA D for L, i.e. L(D)= L. If we make all states accept states (i.e. F=B), the new DFA D' will accept the (L). We can show this easily.

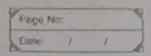
If D'accepts w, then to starting from w's fund state, go to the accept state of D. The string so generated will be st. ww'e L >we fre (L).

If we fre (L), a run of it on D' will stop at some state. But this is an accepting state in D'; thus D'accepts w.

BED

4. We need to prove that

ESpER = Z(M, M2) (CM,) = LCM2)3.



We know that Epra = { (M) | LOM) = \$ 3 is reachable states in M). Now construct a TM & Jos ESPEA as follows:

8 ((M) 2, (M2)) := Construct a DFA for L(M2) o L(M) & L(M) n L(M2), 3 azy M, N. Check for (M) XWE EDFA.

If both are true, accept.

Otherwise reject. This works because we have algorithms for constructing DFAs for the complements and interections I regular languages, as in step 1. Furthermore, if L1 = LOM1) & L2 = LOM2) are such that L1 \ L2 = P & L2 \ L, = p, then L, = L2 & 12 5 L, = L2, QED.

5. (A) Regular C Context-Free C Recursive C Recursively

This is be cause each does can be proved as being included in the next, and there are examples of languages in a higher class but not a lower (hence strict subset).

E.g. 30"1" | n > 03 is a CFL but not regulas.
30"1"2" | n > 03 is secursive but not a CFL Am is RE but not recursive.

6. Let we = Suppose that L is a CFL. Then let p be its pumping length. Consider the string w = apparbabab = (app) = L. If we try to pump this string as w = www. uvxyz then consider v & y. In the strings w; = UV'sxy'z, the number of consider Wo = UNZE Deleting & & y from If v & y have no b's, then wo would be of the form a barbarb, or albarbarb, og arbalbarb, where k, l<p This is impossible > Now, v & y have some b's. Since we must have either 3 or 0 b's, it means that my must have 3 & wa must have O.

