Automata Theory (CS1.302)
End Sem Exam

Abhinav S Menon
2020114001

1. ~~PDA~~ The Halting Problem for PDAs is decidable, while the Halting Problem for TMs is not. They do not match.
The HP for TMs can proved to be undecidable by reduction from the ~~halting probl~~ acceptance problem A_{TM}, 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 H_{PDA} can convert the PDA to a CFG in CNF & test for halting algorithmically.

2. (A) ($RE_1 \cup RE_2$) is recursively enumerable.
~~When~~ We can show this by dovetailing the recognisers for RE_1 & RE_2 and accepting as soon as either one accepts. As there is no guarantee that both will halt, we cannot have a decider.

(B) ($R_1 \cap RE_2$) is recursively enumerable.
We need both the decider for R_1 and the recogniser for RE_2 to halt in order to give an answer. As

Page No:
Date: / /

the latter may not ever occur, we can only have a recogniser.

(c) $(R_1 - R_2)$ is ~~de~~ recursive.
As the deciders for R_1 & R_2 both halt, we can run them successively (there is no need of dovetailing) and return the XOR of their results, which is computable

3. We have $Pre(L) = \{x \in \Sigma^* \mid \exists y \in \Sigma^*: xy \in L\}$.
We need to show it is regular for regular L.

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

If D' accepts w, then starting from w's final state, go to the accept state of D. The string w' so generated will be s.t. $ww' \in L \Rightarrow w \in Pre(L)$.

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

QED.

4. We need to prove that
$$EQ_{DFA} = \{\langle M_1, M_2\rangle \mid L(M_1) = L(M_2)\}.$$
is recursive.

We know that $E_{DFA} = \{ \langle M \rangle \mid L(M) = \phi \}$ is recursive (it can be decided by searching for reachable states in M). Now construct a TM Q for EQ_{DFA} as follows:

$Q(\langle M_1 \rangle, \langle M_2 \rangle) :=$ Construct a DFA for $L(M_2) \cap \overline{L(M_1)}$ & $L(M_1) \cap \overline{L(M_2)}$, say M, N.
Check for $\langle M \rangle, \langle N \rangle \in E_{DFA}$.
If ~~both~~ are true, accept.
Otherwise reject.

This works because we have algorithms for constructing DFAs for the complements and intersections of regular languages, as in step 1. Furthermore, if $L_1 = L(M_1)$ & $L_2 = L(M_2)$ are such that $L_1 \setminus L_2 = \phi$ & $L_2 \setminus L_1 = \phi$, then $L_1 \subseteq L_2$ & $L_2 \subseteq L_1 \Rightarrow L_1 = L_2$, QED.

5. (A) Regular $\subset$ Context-Free $\subset$ Recursive $\subset$ Recursively Enumerable

This is because each class 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. $\{0^n 1^n \mid n \geq 0\}$ is a CFL but not regular.
$\{0^n 1^n 2^n \mid n \geq 0\}$ is recursive but not a CFL.
A_{TM} is RE but not recursive.

6. ~~Let w =~~ Suppose that L is a CFL. Then let p be its pumping length.
Consider the ~~string~~ string $w = a^p b a^p b a^p b = (a^p b)^3 \in L$.

If we try to pump this string, as $w =$ ~~uvwxy~~ $uvxyz$, then consider v & y. In the strings $w_i = uv^i xy^i z$, ~~the number of~~ consider $w_0 = uxz \in L$.

→ ~~Deleting v & y from~~ If v & y have no b's, then
- w_0 would be of the form $a^k b a^l b a^p b$, or $a^p b a^k b a^l b$, or $a^k b a^p b a^l b$, where $k, l < p$. This is impossible.

→ Now, v & y have some b's. Since w_0 must have either 3 or 0 b's, it means that vy must have 3 & w_0 must have 0.

→ Thus one of v or y has 2 b's. For this to happen, it must have at least p a's
⇒ $|vxy| > p$, which is not allowed.

∴ w cannot be pumped & L is not a CFL, QED.

7.

8. (a) $0^*(10^*)^* = (1^*0)^*1^*$

(b)