CS 3100 – SOLUTIONS to mock Final Exam – TOTAL 100 points

The multiple choice problems given here can earn you positive points (correct answer) or and negative points (incorrect). We show it as **[+m/-n]**. You must write a compact two-sentence (approx.) explanation in support of your answer, without which you won't gain any points. **You must put a check mark ($\surd$) in one of the squares associated with each question.**

PART-1 is similar to that of Midterm-2; just giving more practice below

1. **[+5/-1] Choose from various answers below.**

   A. A DFA reads its input fully before accepting a string Yes; this is by definition so.

   B. A multi-tape TM is equivalent to a single tape TM Yes; read how single tape can be simulated.

   C. A DTM may accept a string without reading its input Yes; a TM is a generalized computational device. It may be designed to read part of its input, skip over the rest and go into a loop. As a matter of fact, we can't force it to be always reading all of the input.

   D. The number of configurations of an LBA is fixed by its number of states Q. No; it also depends on the input length and the tape alphabet size.

   ☐ All these assertions are true.

   ☐ Assertions A, B, and C are true. This one.

   ☐ Assertion A and C alone are true.

   ☐ Assertions A and D are true.

   Explanation: See above.

2. **[+5/-1] Choose from various answers below.** The Schröder-Bernstein Theorem

   A. helps establish a bijection between two sets A and B by finding two one-to-one onto functions $f : A \to B$ and $g : B \to A$. No; see below.

   B. . . . by finding two one-to-one into functions $f : A \to B$ and $g : B \to A$. This one; the advantage is to require only 1-1 into maps.

   C. was used in class to show that the number of C programs is countably large. Yes indeed.

   D. is another way to present the Diagonalization proof. No.

   ☐ None of these assertions are true.

   ☐ Assertion A alone is true.

   ☐ All assertions except A are true.

   ☐ Assertions B and C alone are true. This one.

   Explanation: See above.

3. **[+5/-1]** Consider these assertions.

A. CFLs are closed under intersection. No; consider $a^m b^n c^n$ and $a^m b^m c^n$.

B. RE languages are closed under intersection. Yes; one can run TMs one after the other to get a TM for the intersection.

C. RE languages are closed under complementation. No; Else every language will be recursive.

D. Either a language L is RE or its complement $\overline{L}$ is RE. No; there are non-RE languages whose complements are also non-RE.

- ☐ All of these assertions are true.
- ☐ Assertion B alone is true. This one.
- ☐ All assertions except A are true.
- ☐ Assertions B and D alone are true.

Explanation: See above.

4. **[+5/-1]** Consider the Pumping Lemma proofs discussed in this course; call them RPL and CPL for the regular and context-free Pumping lemmas. Recall that the main parts of these PLs are as follows: (i) in RPL, a string $uvw \in L \Rightarrow \ \forall i : \ uv^iw \in L$. (ii) in CPL, a string $uvwxy \in L \Rightarrow \ \forall i : \ uv^iwx^iy \in L$. Now consider the assertions.

A. In RPL, $v \neq \varepsilon$ because the language L is not empty No.

B. In CPL, $vx \neq \varepsilon$ because the grammar of L is assumed to be unambiguous. No.

C. In RPL, $v \neq \varepsilon$ because the loop in the DFA has a length of at least 1. Yes.

D. In CPL, $vx \neq \varepsilon$ because the grammar of L is assumed to be in the Chomsky Normal form. Yes.

- ☐ Assertions A, B, and C are true
- ☐ Assertions C and D are true This one.
- ☐ Assertion C alone is true
- ☐ Assertions A, C, and D are true

Explanation: If $vx = \varepsilon$, we will have some non-terminal `V` lying along one path from the root to the leaves that supports two distinct parse trees spanning string uwy with the upper occurrence of `V` (see a diagram depicting the CFL Pumping Lemma, such as the one below) having some ε derivations. But in a Chomsky normal form grammar, a `V` that is not the same as `S` cannot have a ε derivation.

Now you might say that the `V` itself is `S`. But then, we have two facts: (i) only `S` can have the ε production, and (ii) the other non-terminals cannot involve `S` in their right-hand sides (the grammar is turned into a form where `S` is allowed to refer to other non-terminals, but not vice versa).

With these constraints, we cannot have both v and x turn into ε.

```
         S
         .
         .
         V
        /. \
       / .  \
      /  .   \
     /   V    \
    /   /.\    \
   /   / . \    \
  /   /  .  \    \
uuvvvvwwwwwwwxxxxxyyyy
```

PART-2 will be similar to that of Midterm-2