CSE491/596, Fall 2021 Problem Set 3 Due Wed. Oct. 27

Lectures and Reading (similar to last year's Assgt. 4) Starting from Debray's notes back in section 9, Wednesday's lecture will pick up the example in Theorem 9.1 but relate it to ALL_{TM} . Skim the discussion of Post's Correspondence Problem. The first part of section 10, Theorem 10.2, will be covered later in-tandem with Theorem 13.12 when we begin complexity theory on Friday. Skip page 30 with the subject of "oracles"—IMHO it has limited value until late in the complexity theory unit where binary search becomes a quintessential example of an oracle process. But page 31 picks up with useful examples of the kind also slated for the Wed. 10/20 lecture, when also Rice's Theorem will be proved by applying the all-or-nothing switch idea as a filter, pretty much as Debray does it on page 32. From there until the end of section 12, the material becomes presentation options as detailed further below. This all will enable us to connect section 13 to what has been going on in sections 8–10. Likewise the first half of section 14—but stop before Theorem 14.9 on page 46 of the notes.

——— *Assignment 3, due Oct. 27 "midnight stretchy" on CSE Autograder* ———

(1) Give in pseudocode a decision procedure for the following problem:

INSTANCE: Three DFAs M_A, M_B, M_C.

QUESTION: Does M_C accept every string that is accepted by *both* M_A and M_B? I.e., is $L(M_C) \supseteq L(M_A) \cap L(M_B)$?

Also say and justify whether your algorithm runs in polynomial time. (18 + 6 = 24 pts.)

(2) Given a Turing machine M, say that a string $x \in \Sigma^*$ is "covered" by M if for each character x_j in x, there is a substring w of x including x_j that M accepts. Put another way, for each j there are i, k such that $i \leq j \leq k$ and $x[i \ldots k]$ is in $L(M)$. Let $C(M)$ stand for the language of strings covered by M. Note that $C(M)$ includes $L(M)$, since every string in $L(M)$ covers all its characters in one go, but $C(M)$ can be larger. A tricky example is that the language of even-length palindromes covers the string *abbaababbaba* via the three substrings *abba*, *baab*, and *ababbaba*.

(a) Show that if M is total, then $C(M)$ is decidable. (15 pts.)

(b) What if M is not total—is $C(M)$ still at least computably enumerable? Justify your answer. (6 pts.)

(c) Suppose M runs in $O(n)$ time, where $n = |x|$ as usual. Does your algorithm in part (a) run in $O(n)$ time, or are you taking quadratic or cubic time? (6 pts., for 27 total)

(3) Prove by a mapping reduction from A_{TM} that the (language of the) following decision problem is undecidable:

INSTANCE: A deterministic Turing machine M with 2 tapes, and an input x to M.

QUESTION: Does $M(x)$ eventually write a non-blank character on Tape 2?

Also explain why the language of this problem **is** computably enumerable. ($18 + 3 = 21$ pts.)

(4) Prove by mapping reductions that the language of the following decision problem is neither c.e. nor co-c.e. You may use any suitable problems as sources of your reductions, though the basic choices $A_1 = A_{TM}$ (or K_{TM}) and $A_2 = D_{TM}$ can always be made to work.

INSTANCE: A deterministic one-tape Turing machine M with alphabet $\Sigma = \{0, 1\}$.

QUESTION: Is $L(M) = 0^*$, i.e., does M accept exactly the binary strings without a '1' in them?

(Note that quoting Rice's Theorem would only give you undecidability. $2 \times 18 = 36$ pts., for 108 total on the set)