CS46 practice problems 7

These practice problems are an opportunity for discussion and trying many different solutions. They are **not counted towards your grade**, and **you do not have to submit your solutions.** The purpose of these problems is to get more comfortable with reasoning and writing about Turing machines, decidability, and recognizability.

If you are stumped or looking for guidance, some of these problems are in the "selected solutions" portion of the textbook.

1. A Turing machine with stay put instead of left is similar to an ordinary Turing machine, but the transition function has the form

   $$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, S\}$$

   At each point the machine can move its head right or let it stay in the same position. Is this Turing machine variant equivalent to an ordinary Turing machine? If not, what class of languages do these machines recognize?

2. **Closure properties.** Show that the collection of Turing-recognizable languages is closed under the operations of:

   (a) union

   (b) concatenation

   (c) Kleene star

   (d) intersection

   (e) **Extra credit.** homomorphism (refer to homework 4 for definition)

   Keep in mind that on the previous lab, you were asked to show that the collection of Turing-decidable languages is closed under some operations. This question is different! — you might have to deal with Turing machines that do not halt.

3. **Useless variables.** Given a grammar G, we say that a variable $V \in G$ is "useless" if there is no string w for which a possible derivation of w contains the variable V. Formulate the problem of finding grammars containing useless variables as a language and show that this language is decidable.

4. Let A be a Turing-recognizable language consisting of descriptions of Turing machines, $A = \{\langle M_1 \rangle, \langle M_2 \rangle, \ldots\}$, where every M_i is a decider. Prove that some decidable language D is not decided by any decider M_i whose description appears in A.

   (Hint: you may find it helpful to consider an enumerator for A.)

5. Show that $INFINITE_{DFA}$ is decidable, where $INFINITE_{DFA}$ is defined as:

   $$INFINITE_{DFA} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) \text{ is an infinite language}\}$$

   (Hint: you may want to refer to the textbook for some other decidable languages related to DFAs, and recall which operations we know how to build DFAs for, back when we proved things like "regular languages are closed under intersection".)