18.404/6.840 Michael Sipser
Theory of Computation
Due via Gradescope 2:30pm sharp, Thursday, October 8, 2020 **Problem Set 3**

Read all of Chapter 5 and Section 6.1.

0.1 Read and solve, <u>but do not turn in</u>: Book, 5.14 . [TM left end overrun is undecidable]

1. Consider the problem of determining whether a single-tape Turing machine ever writes a blank symbol over a nonblank symbol during the course of its computation on any input string. Formulate this problem as a language and show that it is undecidable.

2. Let A be a language.

   (a) Show that A is Turing-recognizable iff $A \leq_{\mathrm{m}} A_{\mathsf{TM}}$.

   (b) Show that A is decidable iff $A \leq_{\mathrm{m}} \mathtt{0}^*\mathtt{1}^*$.

3. Let $AMBIG_{\mathsf{CFG}} = \{\langle G\rangle |\ G \text{ is an ambiguous } \mathsf{CFG}\}$. Show that $AMBIG_{\mathsf{CFG}}$ is undecidable. (Hint: Use a reduction from PCP. Given an instance

$$P = \left\{ \left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \ldots, \left[\frac{t_k}{b_k}\right] \right\}$$

   of the Post Correspondence Problem, construct a CFG G with the rules

$$\begin{aligned} &S \to T \mid B \\ &T \to t_1 T \mathtt{a}_1 \mid \cdots \mid t_k T \mathtt{a}_k \mid t_1 \mathtt{a}_1 \mid \cdots \mid t_k \mathtt{a}_k \\ &B \to b_1 B \mathtt{a}_1 \mid \cdots \mid b_k B \mathtt{a}_k \mid b_1 \mathtt{a}_1 \mid \cdots \mid b_k \mathtt{a}_k \end{aligned}$$

   where $\mathtt{a}_1, \ldots, \mathtt{a}_k$ are new terminal symbols. Prove that this reduction works.)

4. Say that a variable A in CFG G is ***redundant*** if removing it and its associated rules leaves $L(G)$ unchanged. Let $REDUNDANT_{\mathsf{CFG}} = \{\langle G, A\rangle |\ A \text{ is a redundant variable in } G\}$.

   (a) Show that $\overline{REDUNDANT_{\mathsf{CFG}}}$ is Turing-recognizable.

   (b) Show that $REDUNDANT_{\mathsf{CFG}}$ is undecidable.

5. Define a ***two-headed finite automaton*** (2DFA) to be a deterministic finite automaton that has two read-only, bidirectional heads that start at the left-hand end of the input tape and can be independently controlled to move in either direction. The tape of a 2DFA is finite and is just large enough to contain the input plus two additional blank tape cells, one on the left-hand end and one on the right-hand end, that serve as delimiters. A 2DFA accepts its input by entering a special accept state. For example, a 2DFA can recognize the language $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n |\ n \geq 0\}$.

   (a) Let $A_{\mathsf{2DFA}} = \{\langle M, x\rangle |\ M \text{ is a 2DFA and } M \text{ accepts } x\}$. Show that A_{2DFA} is decidable.

   (b) Let $E_{\mathsf{2DFA}} = \{\langle M\rangle |\ M \text{ is a 2DFA and } L(M) = \emptyset\}$. Show that E_{2DFA} is not decidable.

6. Give an example in the spirit of the recursion theorem of a program in a real programming language (or a reasonable approximation of a programming language) that prints itself out.

7.* (optional) Show that $EQ_{\mathsf{TM}} \not\leq_{\mathrm{m}} \overline{EQ_{\mathsf{TM}}}$.

Midterm exam: Thursday, October 15, 2020, 90 minutes, start time flexible. Covers Chapters 1, 2 (except 2.4), 3, 4, 5, and 6.1.

Final exam: Thursday, December 17, 2020, 3 hours, start time flexible. Covers the above plus Chapters 7, 8, 9.1, 9.2, 10.2 (except the section on primality), and 10.4 through Theorem 10.33.