Read all of Chapter 5 and Section 6.1.

0.1 Read and solve, but do not turn in: Book, 5.4 [$A \leq_{\mathrm{m}} B$ and B is regular]

0.2 Read and solve, but do not turn in: Show that $EQ_{\mathsf{TM}} \not\leq_{\mathrm{m}} \overline{A_{\mathsf{TM}}}$.

1. Consider the problem of determining whether a given Turing machine M on a given input w ever attempts to move its head left when its head is on the left-most tape cell. Formulate this problem as a language and show that it is undecidable by using a reduction from A_{TM}.

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 $A \leq_{\mathrm{m}} \mathtt{0^*1^*}$.

3. Define a ***2-pass pushdown automaton*** (2PPDA) to be identical to the standard PDA as we have defined, except that it reads the input twice: the input head makes two, left-to-right, passes over the input. When the 2PPDA reaches the end of the input for the first time, it is reset back to the start of the input without changing its state or stack, except that a special symbol is added to the the top of the stack to inform the 2PPDA that the reset has occurred. Thus the 2PPDA can detect when the second pass begins. When the 2PPDA reaches the end of the input for the second time, the input is accepted if the 2PPDA is in an accept state.

   (a) Show that a 2PPDA can recognize the language $\{\mathtt{a}^m\mathtt{b}^m\mathtt{c}^m\mathtt{d}^m | \ m \geq 0\}$.

   (b) Show that $E_{\mathsf{2PPDA}} = \{\langle P\rangle | \ P \text{ is a 2PPDA where } L(P) = \emptyset\}$ is undecidable.
   Your solution should be detailed enough to show why it does prove that E_{2PPDA} is undecidable, yet it doesn't prove that E_{PDA} is undecidable (since E_{PDA} *is* decidable).

4. Let $DISJOINT_{\mathsf{CFG}} = \{\langle G, H\rangle | \ G \text{ and } H \text{ are CFGs and } L(G) \cap L(H) = \emptyset\}$. Show that $DISJOINT_{\mathsf{CFG}}$ is undecidable. (Hint: Use a reduction from PCP. Given an instance

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

   of the Post Correspondence Problem, construct CFGs G and H with the rules

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

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

5. Let $FINITE_{\mathsf{TM}} = \{\langle T\rangle | \ T \text{ is a TM and } L(T) \text{ is a finite language}\}$.

   (a) Show that $A_{\mathsf{TM}} \leq_{\mathrm{m}} \overline{FINITE_{\mathsf{TM}}}$ to prove that $FINITE_{\mathsf{TM}}$ is not T-recognizable.

   (b) (harder) Show that $A_{\mathsf{TM}} \leq_{\mathrm{m}} FINITE_{\mathsf{TM}}$ to prove that $\overline{FINITE_{\mathsf{TM}}}$ is not T-recognizable.

6. Give a Python program that prints itself out, in the spirit of the recursion theorem. Submit a running program with its output. You may use escape characters such as `\"` and `\\` and the function `print()` but not special implementation features such as character codes or other pre-defined functions such as `repr()`. If you don't know Python, you may use some other programming language. (Hint: Assign a string to `S` to serve as a template. The substring operation `S[i:j]` and escaped quotes are useful.)

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