Problem 1. Draw a TM $M$ which decides the language $A = \{a^n b^n c^n \mid n \geq 0\}$, where the input alphabet $\Sigma = \{a, b, c\}$.

Problem 2. Give an algorithm for a TM $M$ which decides the language

$$C = \{\langle D, \omega \rangle \mid D \text{ is a DFA and } D \text{ accepts } \omega\}.$$  

Note that $\langle D, \omega \rangle$ is some reversible encoding of $D$ and $\omega$ into a string. The precise details of how this encoding is computed is irrelevant to the algorithm you give. The key point is you have access to each parameter (you should do more than write "Simulate $D$ on an input").

Problem 3. Explain why the class of decidable languages is closed under

a) union,

b) intersection,

c) concatenation,

d) Kleene star.

You do not need to formally prove each, but your explanation should sketch the outline of a proof.

Lastly, describe how your explanations would differ for the class of recognizable languages.

Problem 4. Define $\omega_i$ to be the $i^{th}$ element of $\Sigma^*$ in lexicographical order. When $\Sigma = \{0, 1\}$, for example, this order is $\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots$

To prove that enumerators are equivalent to Turing machines, we created an enumerator $E$ given a TM $M$ such that $L(E) = L(M)$. The algorithm for this is given below.

$$E = \text{"Ignore input,}$$

1. For $i = 1$ to $\infty$
   
   (a) For $j = 1$ to $i$
      
      i. Run $M$ on input $\omega_j$ for $i$ steps
      
      ii. If $M$ accepts, print $\omega_j$"
Explain why the following algorithm for $E$ does not work.

$$E = "\text{Ignore input,}\,$$

1. For $i = 1$ to $\infty$
   (a) Run $M$ on input $\omega_i$
   (b) If $M$ accepts, print $\omega_i$"

Problem 5. Let $E$ be an enumerator that prints its language in lexicographical order. Prove that $L(E)$ is decidable.