COT 4210 Homework #5: Decidability
Due Date: Tuesday October 19, 2010 (in class)

1) Let $ALL_{DFA} = \{ <A> \mid A \text{ is a DFA that recognizes } \Sigma^* \}$. Show that ALL_{DFA} is decidable.

2) Let $INFINITE_{DFA} = \{ <A> \mid A \text{ is a DFA and } L(A) \text{ contains an infinite number of strings} \}$. Show that $INFINITE_{DFA}$ is decidable.

3) A ordered k-tuple is a member of the set P(k) if it contains a permutation of the values 1, 2, 3, … , k. For example, $P(3) = \{ (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1) \}$. Consider the set $P = \{x \mid x \in P(i), i \in Z^+\}$. (Note: Z^+ is the set of positive integers.) Is P countable? Prove your answer.

Note: The last two questions are programming questions. Please just attach your code to your written homework. We won't actually run your code, but are simply looking for the thinking behind it. But, make sure you run it so (a) you know it works, (b) you can see the tangible result based on this theoretical material.

4) Write an enumerator that prints out all the fractions (in lowest terms) with denominators less than or equal to 1000, in between 0 and 1, not including either value. A gcd function will be useful.

5) Define the halting problem as follows:

$HALT_{TM} = \{ <M, w> \mid M \text{ runs and halts (either accepts or rejects) when run on w.} \}$

This language is NOT Turing decidable, but it is Turing recognizable. A very similar proof to the one shown in class to prove that A_{TM} is not decidable can be used to prove the same for $HALT_{TM}$. Give a programming illustration of this proof similar to the code handout given in class on Tuesday, 10/12. (Note: Your program might just run forever!)