Lectures and Reading. Same as designated last week. Computability and (un)decidability will begin on Monday. Here is the nub of next week’s diagonal argument: Define $D = \{ e(M) : M \text{ does not accept } e(M) \}$. Suppose (for sake of contradiction) that there were a “quixotic” TM $Q$ such that $L(Q) = D$. Then on input $x = e(Q)$ we would get this self-contradictory analysis:

$$x \in D \iff Q \text{ does not accept } x, \quad \text{(by definition of } D \text{ since } x = e(Q))$$
$$\iff Q \text{ does accept } x, \quad \text{(by } L(Q) = D).$$

Having a statement become equivalent to its negation is a contradiction in logic, so $Q$ cannot exist. Thus $D$ is not c.e. Its complement, for which I’ll use the traditional name $K$ but define it as $K = \{ e(M) : e(M) \in L(M) \}$, is c.e., since it is a “slice” of the language of the universal Turing machine to come next week. It follows that $K$ is c.e. but not decidable, so the class RE of c.e./r.e. languages is different from the class REC of decidable languages. (And by Myhill-Nerode it properly contains the class REG of regular languages—indeed we will be able to see why it properly contains larger classes such as $P$ for polynomial time.)

This will basically cap the coverage expected for the first prelim exam, which is set for Wed. Oct. 19 in class period.

(1) Prove that two of the following three languages are non-regular, via a Myhill-Nerode argument. For the regular one, give a regular expression. Here $\#a(x)$ denotes the number of occurrences of the character $a$ in the string $x$, and more generally, $\#w(x)$ denotes the number of occurrences of the substring $w$ in $x$. For example, $\#0100101001 = 2$ even though the two occurrences of the substring 0010 overlap each other. Also, for two strings $x, y$ of the same length, $x \oplus y$ denotes the bitwise exclusive-OR, e.g. $1011 \oplus 0010 = 1001$. All three languages are over the alphabet $\Sigma = \{0, 1\}$.

(i) $L_1 = \{ x : \#0(x) > \#1(x) \}$.

(ii) $L_2 = \{ x : \#01(x) > \#10(x) \}$.

(iii) $L_3 = \{ xy : |x| = |y| \land x \oplus y = 1^{|x|} \}$.

(Note incidentally that $L_3$ equals the language of assignment 1, part (c), if one requires the two strings being added as binary numbers to have the same length. $3 \times 12 = 36$ pts.)

(2) Now consider $L_4 = \{ x : \#010(x) = 0 \land \#101(x) = 0 \}$. Use the Myhill-Nerode technique to show that any DFA $M$ such that $L(M) = L_4$ requires at least 6 states. Then design such a DFA $M$—ideally showing how your proof guided you to it (or vice-versa). Finally explain why you can basically “collapse” $M$ into a generalized NFA with only 2 states $s, f$ such that

$$L(M) = L_{s,s} \cup L_{s,f} \cup L_{f,s} \cup L_{f,f},$$

and use that to give a regular expression for $L_4$. ($12 + 6 + 9 = 27$ pts., for 63 total on the set)