Reading: Tuesday’s lecture will finish the coverage of Chapter 1 and transition to Chapter 2. Hence please read section 2.1 in one block this weekend. Although it is not covered on Prelim I which is being given in class period on Tuesday, March 13, it is good to get the reading done while you are not preparing for the exam.

(1) This problem is “HW4 Online Part” on TopHat, worth 20 pts. as before. The following DFA is used for the TopHat questions 6–10 and is also used on problem (2) below.

(2) Convert the above DFA $M$ into a regular expression $r$ such that $L(M) = L(r)$. First reduce the machine as the TopHat part may suggest, then show the steps of the algorithm. It is AOK to do this one completely “graphically” with certain shortcuts—indeed it is IMHO almost “sight-readable” to begin with. (Hence only 12 pts.)

(3) Let $r_1 = (a + ba^*b)^*$ and $r_2 = (a + b)^*bb(a + b)^*$. Calculate a regular expression $r_3$ such that $L(r_3) = L(r_1) \setminus L(r_2)$ by applying the ideas and procedures at the end of the Tuesday 2/27 lecture. Note that $r_2$ and complementary machines were part of the TopHat portion, which may assist you in working out things here. (18 pts. total)

(4) Prove that the following languages are not regular by using the Myhill-Nerode technique. All are over the alphabet $\Sigma = \{a, b, c\}$. Recall that $\#c(w)$ means the number of $c$’s in the string $w$.

- $A = \{w \in \Sigma^* : \#a(w) + \#b(w) = \#c(w)\}$.
- $B = \{w \in \Sigma^* : \#a(w) + \#b(w) > \#c(w)\}$. (Same as A but with a $>$ comparison.)
- $C = \{w \in \Sigma^* : \text{for every two } c\text{'s in } cwc, \text{ if } v \text{ is the string between them, then } \#a(v) \geq \#b(v)\}$.

As a hint and intuition for $C$, think of $a$ as “spear” and $b$ as “dragon” and being able to hold as many spears as you like until you hit a $c$, at which point you have to drop all your spears and start again from scratch. When you are choosing $S$ and setting up your strings, don’t use any more $c$’s than are already implied in treating the input $w$ as $cwc$—figuratively meaning there are automatic “checkstops” at the beginning and end of the “dungeon.” Depending on how you did $A$, it is OK for your answer to $B$ to exhibit substantial “proof code reuse.” (9 + 6 + 9 = 24 pts., for 74 total on the set)