COMP 330 Winter 2021
Mid-term Examination Solutions

Prakash Panangaden

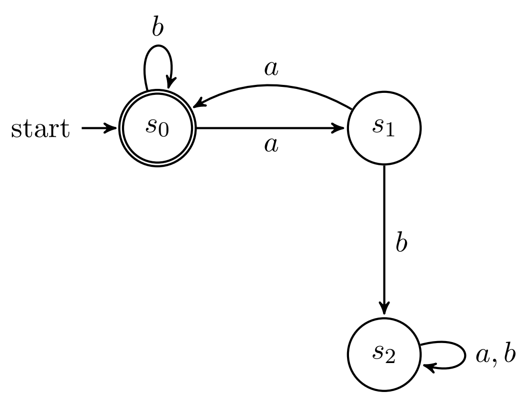
February 4, 2021

Question 1[40 points] In this question the alphabet is fixed as $\{a, b\}$.

- Write a regular expression for the language of strings containing a's only when they occur as part of a block of consecutive a's of *even length*. Thus the legal strings cannot contain an a by itself or a substring of 3 or 5 or $7\ldots$ consecutive a's. Thus *baabbb* is accepted, so is *aabaabaaaabbaabbaa* and so is *bbbbbbb* which has no consecutive pair of a's. However *baaab* is not allowed as this has three consecutive a's nor is *bababaab* or *baaaaab*.
  Solution: $b^*((aa)^*b^*)^*$. The a's have to appear in pairs, but there can be any number (including 0) of b's before the pair of a's. This pattern can be repeated any number of times. Finally there could be a block of b's at the end. There are other correct answers as well.

- Design a DFA (not an NFA) for this language. A picture is preferred. **You must show the dead state if there is one.** For full credit your machine must have 3 states including the dead state (if there is one).

  Solution:



Question 2[40 points] Show, using the pumping lemma, that the following language is not regular. The alphabet is $\Sigma = \{a, b\}$. I prefer answers formatted as a game against the demon.

$$L = \{a^r b^t | r - t = 2, r, t > 0\}.$$

Solution:

The demon chooses some pumping length p. The angel chooses $a^{p+2}b^p$ as the string w. The demon has to choose x, y, z in such a way that the length of xy is less than p. This means that the string y has to be part of the initial block of a's. The demon has no choice. So we have $y = a^k$ where $0 < k \leq p$. The angel chooses $i = 2$ so the new string is $xyyz = a^{p+2+k}b^p$. But the difference is now $p + 2 + k - p = 2 + k \neq 2$, so the pumped string is not in the language and so, the language cannot be regular.

Question 3[20 points]

Are the following statements true or false? No explanations are required. We have some fixed alphabet that we are working with.

1. If L is a non-regular language and R is a regular language then $L \cap R$ must be regular. **FALSE**
2. If L is a non-regular language and R is a regular language then $L \cap R$ cannot be regular. **FALSE**
3. For every regular language there is a unique minimal NFA. **FALSE**
4. When we run the minimization algorithm on a DFA we cannot be sure that it will always terminate. **FALSE**
5. If L_1 is an infinite regular language and L_2 is a finite language then the DFA to recognize L_1 must have more states that the DFA to recognize L_2. **FALSE**