Refo Yudhanto

CS350

HW 4

1. Let Mα be an NFA that accepts the language that contains word α only. We do the same thing for Mβ as an NFA that accepts β. We then construct M1 and M2 as Subseq(Mα) and Subseq(Mβ). That is, M1 accepts the set of all subsequences in α while M2 accepts β. Let M12 be an NFA that accepts" all strings that do not contain abb". Now lets create NFA M that accept the language from M1∩M12∩M2. Thus since M1, M2, and M12 are longest word, the intersection of all 3 is the longest word that satisfy all the conditions.

2. We do the same thing as the question from number 1 without the "abb" condition. Thus we have M that accepts α and β. Now we can eliminate the Λ-transitions in M with 2 cases to consider

   a. Where M accepts an infinite language, thus making D = infinite
   b. Where M accepts a finite language, thus making D exactly the length of the longest word accepted by M, or longest path by DAG

3. Yes, the idea is to take several strings, vectorize those strings, then pass a sliding window over the resulting vectors. If two vectors have the same value in the same window position, mark them as candidates for more fine-grained similarity analysis.