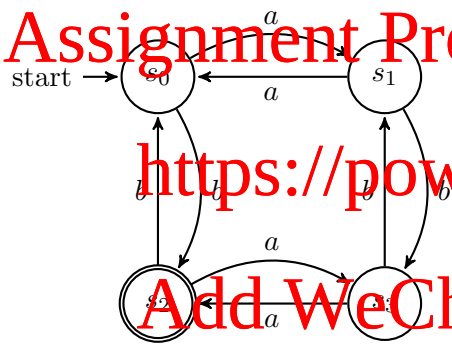
COMP 330 Winter 2021
Quiz 2 Questions and Solutions

Prakash Panangaden

20th January 2021

Question 1 Consider the DFA shown below over the alphabet $\{a, b\}$:



This DFA accepts strings satisfying the following condition:

(a) Every string must contain at least one a but no b's.

(b) Every string must contain at least one b but no a's.

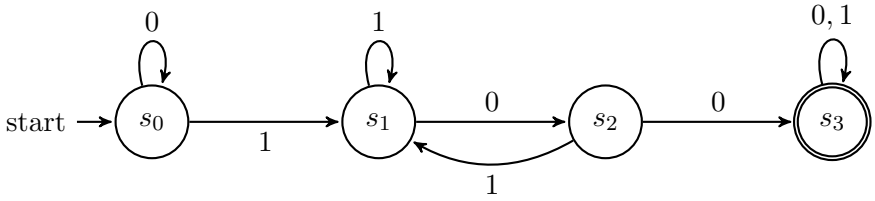
(c) Every string must contain an even number of a's and an odd number of b's.

(d) Every string must contain an even number of a's and any number of b's.

(e) Every string except the empty string is accepted.

Solution (c). This machine toggles between the left and right column as it reads a's and between the upper and lower row as it reads b's. Since the only accept state is in the left column and the lower row it must see an odd number of b's and an even number of a's.

Question 2 In Question 5 of Assignment 1, you were asked to design a DFA over the alphabet $\{0, 1\}$ that would accept any string that contains 100 **or** 110. You were then asked to prove that any such DFA must contain at least 5 states. Some students were convinced that they had a solution with only 4 states and hence the question was wrong. Here is a proposed solution to this question.



The problem with this solution is:

(a) It is an NFA rather than a DFA.

(b) It accepts all the required strings but fails to reject some strings that should be rejected.

(c) It fails to accept some strings that should be accepted but it does correctly reject every string that should be rejected.

(d) It fails to accept some strings that should be accepted and it fails to reject some strings that should be rejected.

(e) It is actually a correct solution and the homework question was wrong.

Solution (c). It does not accept 110. This is actually the correct design for only accepting strings that contain 100 so it does reject all the bad strings.

Question 3 In Question 5 Assignment 1, we asked for a DFA. Those who did it correctly were able to show that the smallest DFA required 5 states. Now consider doing the same question with an NFA.

(a) It is impossible to solve this problem with an NFA.

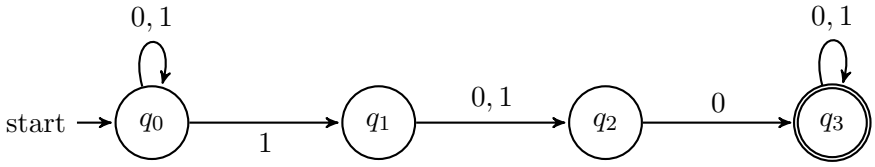
(b) It can be done with an NFA but the smallest NFA will still require 5 states.

(c) It can be solved with an NFA using only 4 states.

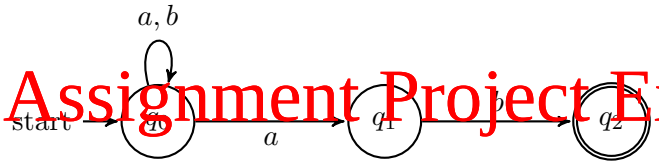
(d) With an NFA the problem can be solved with only 1 state.

(e) With an NFA the problem can be solved but it will need many more states than a DFA for the same problem, perhaps exponentially more.

Solution (c). Here is a solution with 4 states. Using 1 state is clearly impossible.



Question 4 Consider the following NFA:



The regular expression describing the language accepted by the above NFA is:

(a) $(a+b)^*(ab)$

(b) $(a^*+b^*)(a+ab)$

(c) $(ab)^*(a+b)$

(d) $(a+ab)^*$

(e) a^*b^*

Do **not** try to work it out using the algorithm; that will take a ridiculously long time. Just try to look at it and write down the answer.

Solution (a). Think about it!!

Question 5 Exactly one of the following statements is true. Which one is true?

(a) An NFA over a 2-letter alphabet with only one state must either accept every string or reject every string.

(b) There are languages that can be recognized by an NFA but which cannot be recognized by any DFA.

(c) An NFA with ε-moves can accept languages that cannot be defined by a regular expression.

(d) NFA's, DFA's, NFA's with ε-moves and regular expressions all define the same kind of languages.

(e) An NFA only works correctly on Sundays.

Solution (d). But it is interesting to think about why (a) is wrong. Make sure you understand that!