Adam Livingston

5 Apr 2022

BASV 376 Mathematics for Applied Technology

Assistant Professor Emily Carroll

Practical Exercise 4

**Question 1**

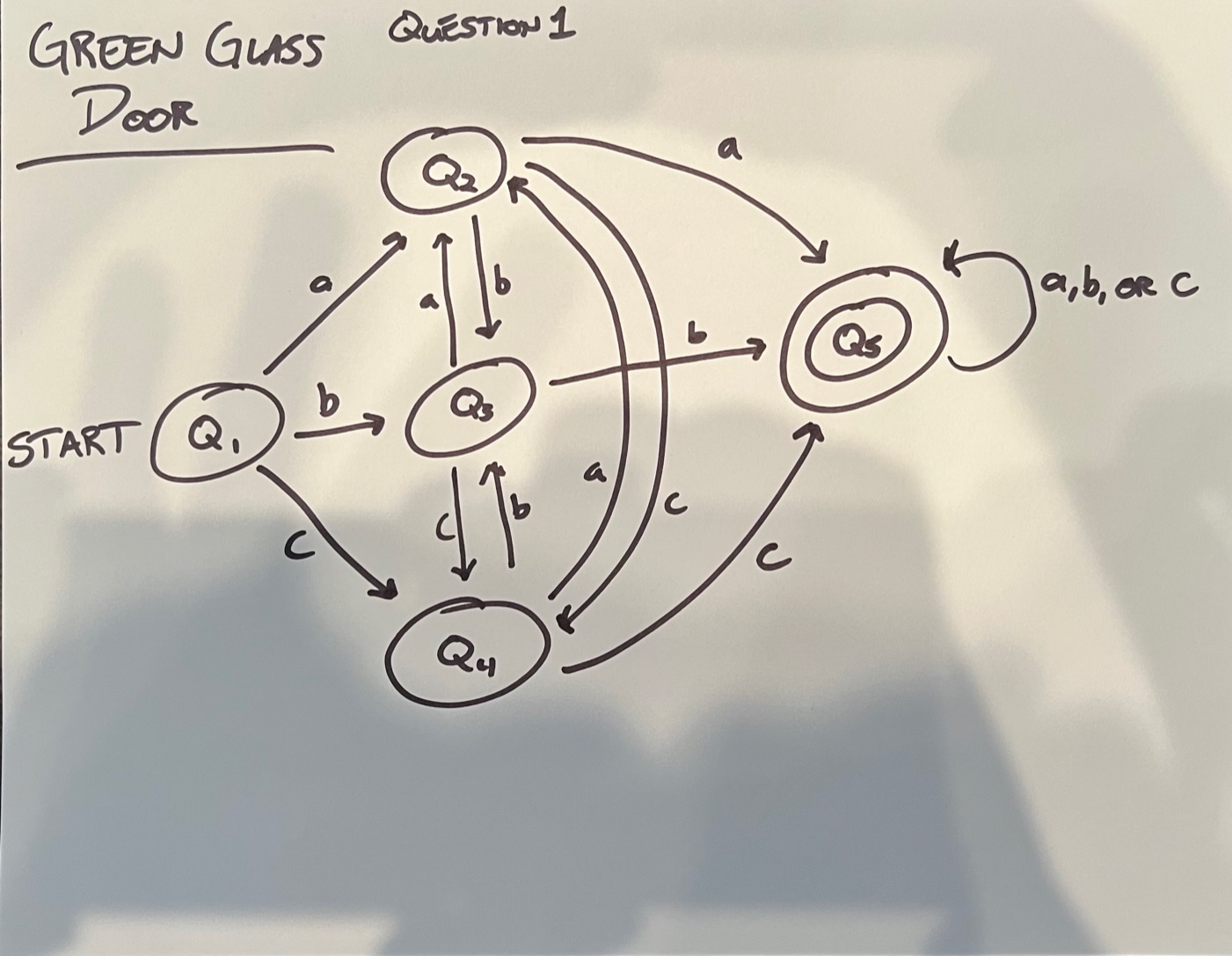
*a. The “Green Glass Door” is a word game in which players take turns naming words with*

*double letters, as in “green” (double e’s) “glass” (double s’s) and “door” (double o’s).*

*Design a finite automaton which only accepts strings with at least one pair of double letters. For the sake of this problem, limit your alphabet to {a, b, c}. Accepted strings do not need to be real English words.*

*Rejected: abc, cab, acbc*

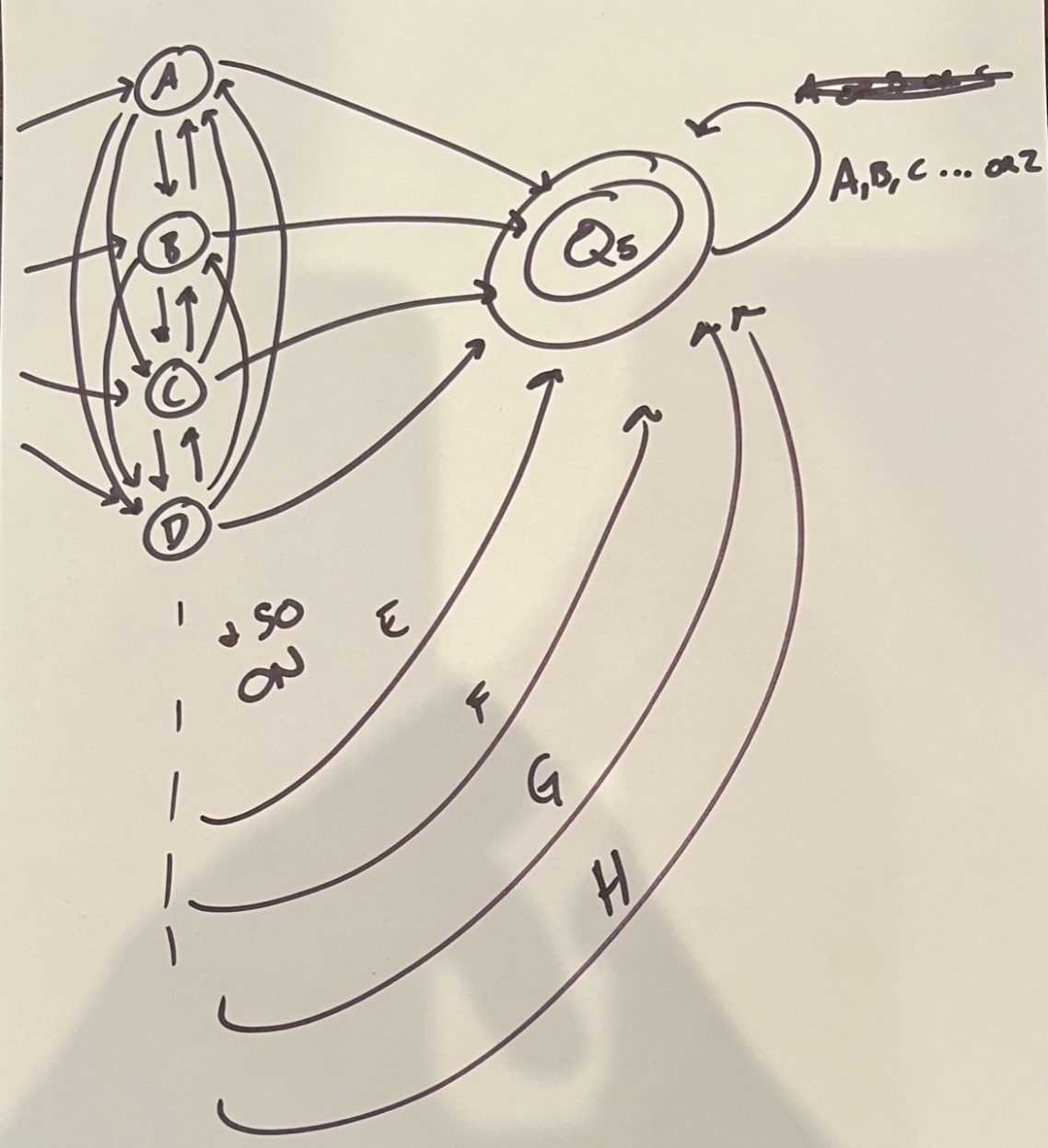
*Accepted: acc, bba, ccbb*



*b. Is your automata deterministic or nondeterministic? How do you know?*

This is a deterministic automaton because the transition function output is a single state and there is a single start state. Additionally, the final state is entirely determined by the input string. This automaton is scalable to any string length of abc since once the criteria of double letters is achieved, it goes into a loop at Q5.

This simple automaton is also scalable to include the entire alphabet for the indicated parameter of determining a string with one pair double letters. From the start point of Q1, there will need to be 26 individual nodes of A through Z but a maximum number of nodes being 28 (26 letters + 1 start node + 1 final state node). Each of the 26 nodes will have 26 outbound edges and 26 inbound edges, the outbound edges indicating if the next letter in the string is another letter in the alphabet or the same letter leading to the final state (Q5 in this case). The additional inbound edge minus a loop on the same letter (accounted for edge leading to Q5) will be from the start point, Q1. To prevent brain-melt of the assistant professor grading this assignment, a shorthand example of how this is scalable is provided below.



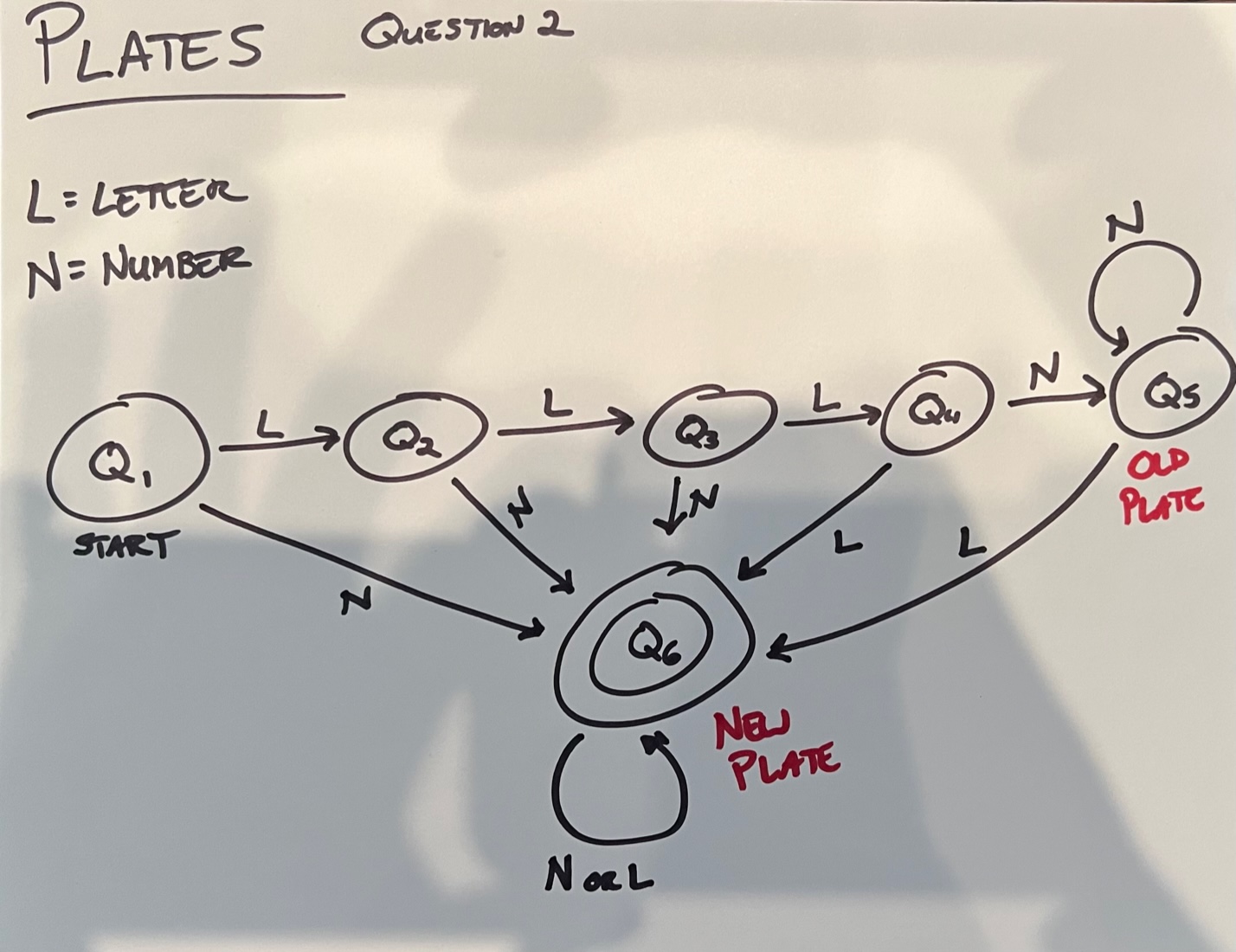
**Question 2**

*a. Most Arizona license plates are 7 characters. Prior to April 2020, they were in the form AAABBBB where A represents a letter and B represents a number. In April 2020, to expand the number of possible plates, the state changed the format to any 7 alphanumeric characters.*

*Design a finite automaton that decides if a plate is old or new. It should have 2 accept states: one for old plates, and one for new plates.*

*Old: OPS4028, ABC1234*

*New: AZ7W332, 2DJ8T6M*



*b. Is your automata deterministic or nondeterministic? How do you know?*

This is a deterministic automaton because the transition function output is a single state and there is a single start state. Additionally, the final state is entirely determined by the input string. The only thing that matters is the interpreting the end state.

This automaton is designed to answer the simple question, “Is this license plate the newer version?” which has either a “yes” or “no” (true or false) answer. If the end state is in Q5, then the plate is older (false/no), made prior to April 2020. If the end state is in Q6 (true/yes), then it is a new plate. Since we know that all older versions of license plates follow LLLNNNN, any deviation from that pattern would indicate a new plate. If the first three characters contain a number, then it is automatically a new plate and the automaton will revert to a loop in Q6. If the first three characters are letters, then the automaton will loop in Q5 unless an integer is detected, indicating a new plate and then loop in Q6.

Since all license plates consist of 7 alphanumeric characters in this problem, the only possible final/end states in this automaton are nodes Q5 and Q6. Since Q6 is the only “true” final state, indicated by the double circle, this easily determines the true/false parameter if a license plate is new or not.