HW5.7. Finite State Machine

Feel free to check out the guide that we have prepared to help you in this problem.

Recall that a Finite State Machine is a computing model that can be used to model a sequential circuit that receives, as input, a string of symbols (one per clock cycle) and outputs one symbol per cycle. For the purposes of this class, we will generally restrict our discussion to FSMs whose input and output streams are bits (1/0).

An FSM is composed of:

A finite set of states, one of which is designated as a "Start" state

For each combination of state and input bit, a rule dictating which state to move to next, and which bit to output.

There are two common ways that we represent a FSM: as a graph, or as a truth table

For this question, we want to design a FSM that outputs 1 if two consecutive zeros have been seen in the most recent three input bits, and 0 otherwise.
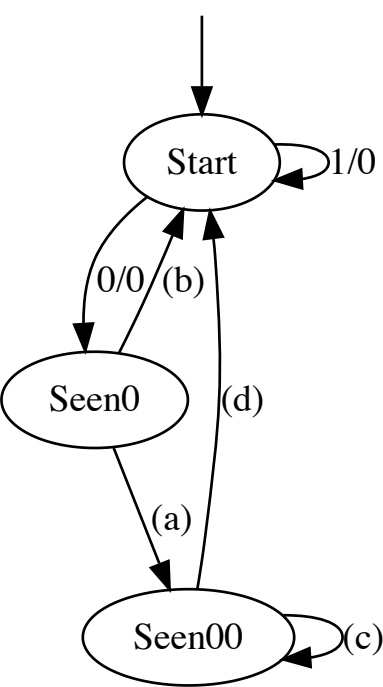
1: Given the input 11000110101001000000, what would the correct output be?

Hint: The output sequence should have the same length as the given input sequence.

Q1.1: [   ] ?

2: We want to make a three-state FSM that works as described above. Fill in the blanks on this graph:

Note that we use an arrow from nowhere to signify the start state, and arrows to signify each transition. The labels on each arrow show the input bit that causes that arrow to be taken, and the output bit to use when that arrow is taken. For example, from the Start state, if we receive an input of 0, we move to the Seen0 state, and output 0. If we receive an input of 1 instead, we move to the Start state, and output 0. Note that each state must have exactly two arrows; one for an input of 1, and the other for an input of 0. Express your answers in the same format as the other labelled edges, in the form "x/y", where x and y are the input and output respectively, with no spaces.

Q2.1: (a): [   ] ?

Q2.2: (b): [   ] ?

Q2.3: (c): [   ] ?

Q2.4: (d): [   ] (?)

Q3.1: Fill out the corresponding truth table for this FSM. The first two lines have been done for you.

Current State	Input	Next State	Output
Start	0	Seen0	0
Start	1	Start	0
Seen0	0	[▽]	[▽]
Seen0	1	[▽]	[▽]
Seen00	0	[▽]	[▽]
Seen00	1	[▽]	[▽]

[Save & Grade *20 attempts left*]   [Save only]   *Additional attempts available with new variants* (?)