Finite State Machine Worksheet

1. Determine which of the following strings would be accepted by the FSM below.
   1. 0
   2. 0111010
   3. 0000000001
   4. The empty string
   5. 010

A diagram of a number and circles

Description automatically generated

1. Fill in the missing information (e.g., start state and transitions) in the FSM below such that it accepts binary strings (strings composed of just “0” and “1”) with the prefix “01”. For example, 0100 would be accepted but 00100 would not.

A diagram of a diagram

Description automatically generated

1. Fill in the missing information in the FSM below such that it accepts binary strings where the number of “1”s is a multiple of 5. For example, 101111 would be accepted but 011110 would not.

A diagram of a hexagon with circles and arrows

Description automatically generated

1. Create an FSM that accepts binary strings of exactly length 4. For example, 0010 would be accepted but 00101 would not.
2. Create an FSM that accepts binary strings with an even number of “1”s. For example, 011011 would be accepted but 010110 would not.
3. Create an FSM that accepts binary strings that start and end with the same symbol. For example, 101 would be accepted but 1010 would not.
4. Create an FSM that accepts binary strings containing the substring "11”. For example, 001110 would be accepted but 01010 would not.
5. Create an FSM that accepts binary strings that end with "110”. For example, 00110 would be accepted, but 01101 would not.
6. Imagine a CTA turnstile that unlocks when a coin is inserted (sensor input “0”) and locks after a person pushes through (sensor input “1”). Model this with a two-state FSM (one state for locked and one state for unlocked).
7. Imagine an automatic door with sensors. The door opens when someone approaches and closes when no one is there. Model this with an FSM.