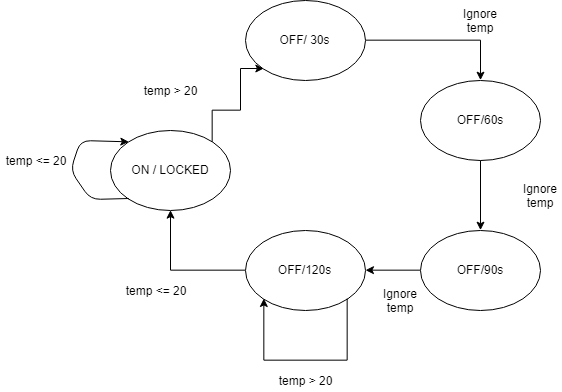
**ECSE 421 – Assignment 1**

**Question 1: FSM Design and Analysis**

**a)**

****

**b)** My thermostat has 5 states. There are always alternative designs with other states, therefore it is not limited to this FSM diagram.

**c)** The FSM model does not have any time-scale invariance property since there exist two timeouts which are a fixed amount of time. Therefore, if the example had a different time instead of 30s, then the diagram will have to adjust the number of states accordingly.

**Question 2: FSM Design**

**a)**

States = {red, yellow, green}

Inputs = ({ticks} -> {present, absent})

Outputs = ({stop, go}-> {present, absent})

Update(state, input)

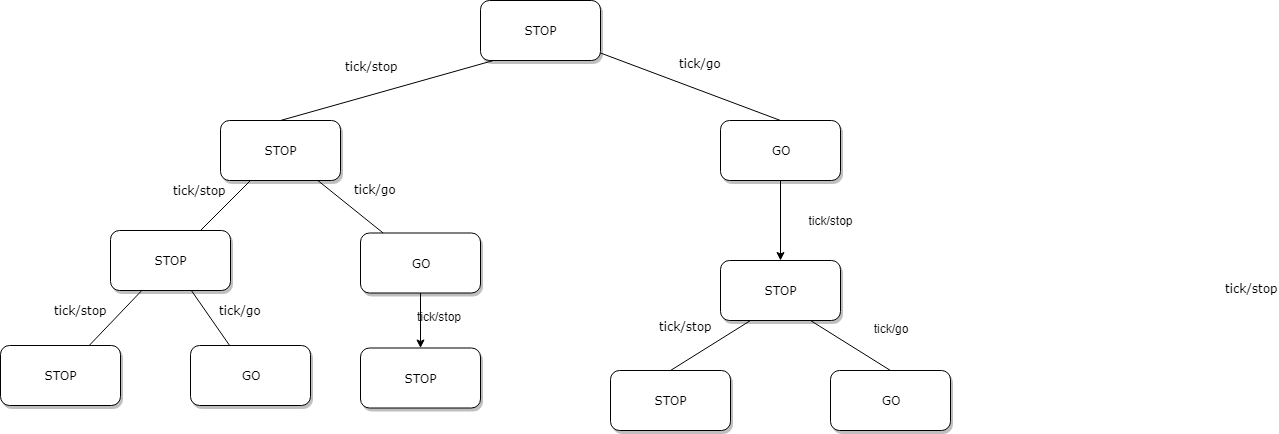
\*\* Assume function isPresent returns true or false depending on if the input is present or not

iniState = red

**b)**

****

**c)** When merging the red and yellow state together into one, the FSM is non-deterministic due to the fact that there are 2 possible routes from that state now since it corresponds to what used to be 2 states.



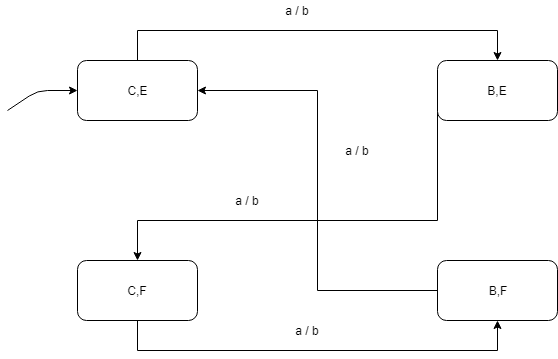
**Question 3: FSM Design**

**\*\*Note: True = possible behaviour of the machine, False = not possible behaviour of the machine**

1. False
2. True
3. False
4. True
5. True

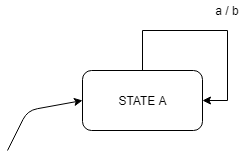
**Question 4: Composition of Finite State Machines**

**a)** Below is the flattened FSM demonstrating the semantics and while ignoring the unreachable states.

****

**b)** The behaviour of this machine is quite simple, when the input is **present** a, it will produce a **present** output of b. Similarly, an **absent** input results in an **absent** output.

A simplified example of the machine is below.



**Question 5: Composition of Finite State Machines**

There are 2 reachable states, since both of the top level states A and B are preemptive. The conditions for the transitions will be enabled when there is a transition from initial state refinements allowing the refinements to always remain within their initial state.