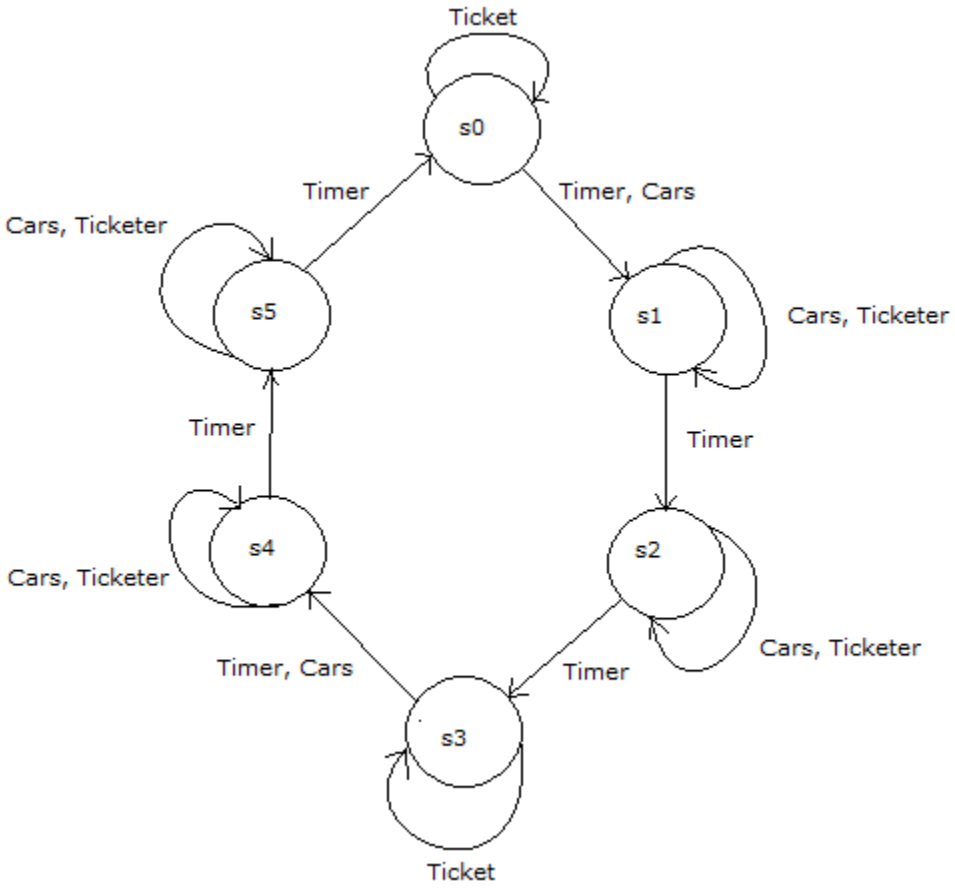
Chip Design for Automated Ticketing on Traffic Light Violations

William Zhang
wxz1067
James McKelvy
jmm0468

CMPS430 Fall 2006

Finite State Machine

The following FSM represents a series of states for an automatic ticketing and a light changing system on a crossroads of streets known as EW and NS. The FSM is actually a model for a small chip controller that will be embedded to control this intersection of busy roads. The roads have equal priority, when too many cars build up at a red light, the system is supposed to trigger a change of state to let them pass. Also, if less than the "maximum buildup" of cars has been waiting for a specified amount of time, the system also triggers a change of state to let them pass. The light system is supposed to cycle through the normal light changes (green, yellow, red, etc..) but also needs to ticket cars that run red lights.



States:

0. EW is Red, NS is Red
1. EW is Green, NS is Red
2. EW is Yellow, NS is Red
3. EW is Red, NS is Red
4. EW is Red, NS is Green
5. EW is Red, NS is Yellow

Definitions:

- Timer: represents a timer that specifies a certain delay between a transition event. For example,

if the FSM is in state 0, and the timer has elapsed 30 seconds, then it should trigger a change to state 1. If the state was 1 and the timer elapsed 5 seconds total in state 1, then it should trigger a change to state 2. Etc…

- Cars: represents a sensor that is triggered when there is a buildup of too many cars. For example, if there were say, more than 10 cars waiting at a red light and the *timer* sensor has not been triggered, then a change of state would be triggered by this (*cars*) sensor. This sensor only applies to a light that has been Red (for the current street) the previous three states and is also Red (for the current street) in its current state.
- Ticket: this is a representation of a sensor that triggers a series of events that occur within the current state. This sensor is only set off if the current street has a Red light and a car has crossed its path. Then the sensor handles the "dirty work" of automatically sending a speeding ticket to the owner of the vehicle, by capturing the license plate number or something similar.

Testing information

To verify that our system is working correctly, we will simulate a series of cars passing down each roadway. The system should behave normally, by cycling through each light state, but the two lights are only allowed to both be red at the same time, no other color. We can test this by outputting the state information at the entrance of a new state. Also, we need to simulate a timer elapsing, and it needs to trigger a change of state after a certain amount of time. We need to specify regular intervals in the model and see whether the system has changed states, and also that it has changed to the correct state. Also we need to simulate a buildup of cars at a red light to see if this will trigger a change of state and to the correct state. We will need to also simulate a few cars passing on the opposite street (that is green or yellow) to see if the system will automatically send a ticket and loop back to the same state that it was in. In short, we should be outputting any change in state and the variables associated with it each time an event happens to verify that the expected behavior is occurring.