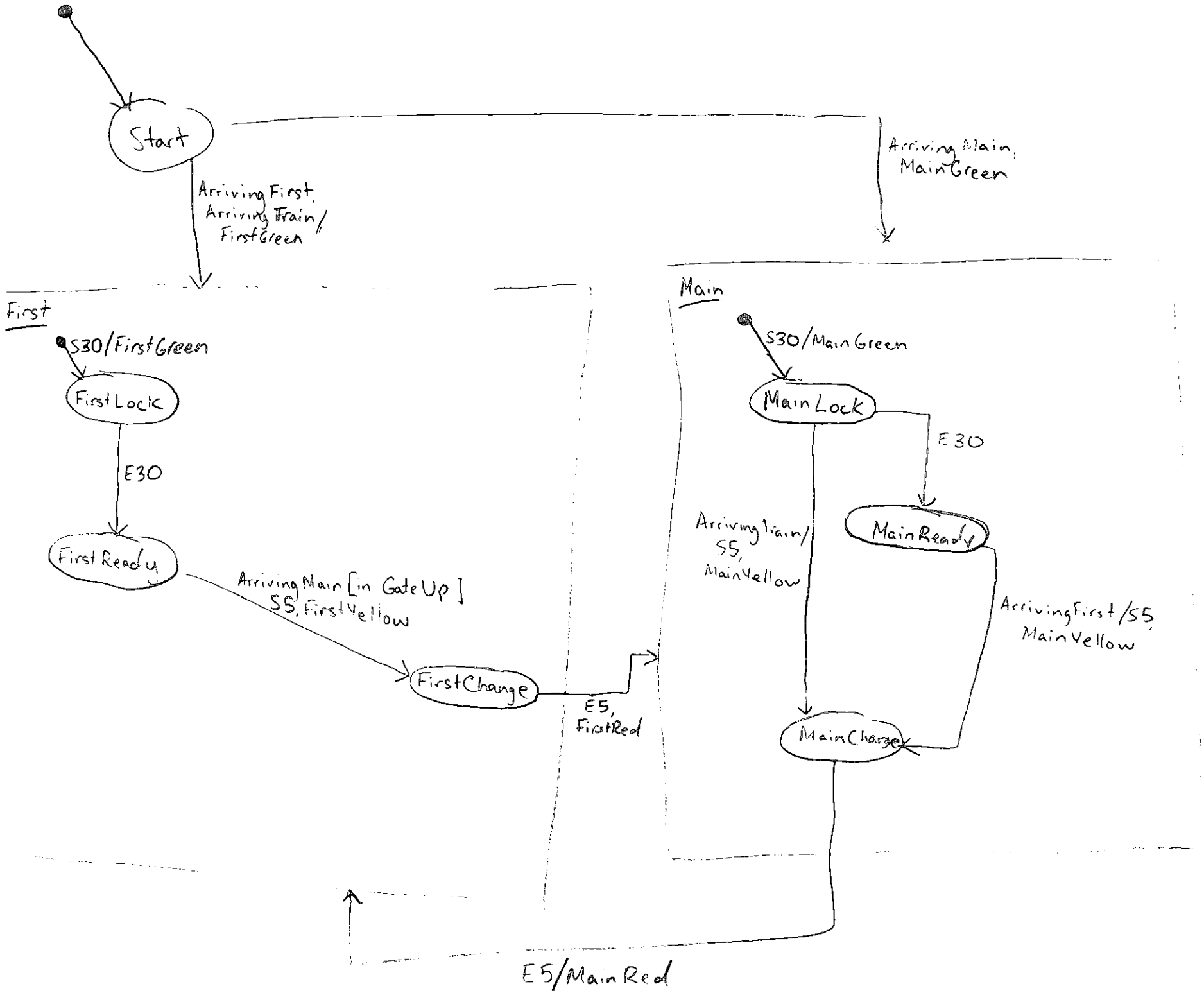
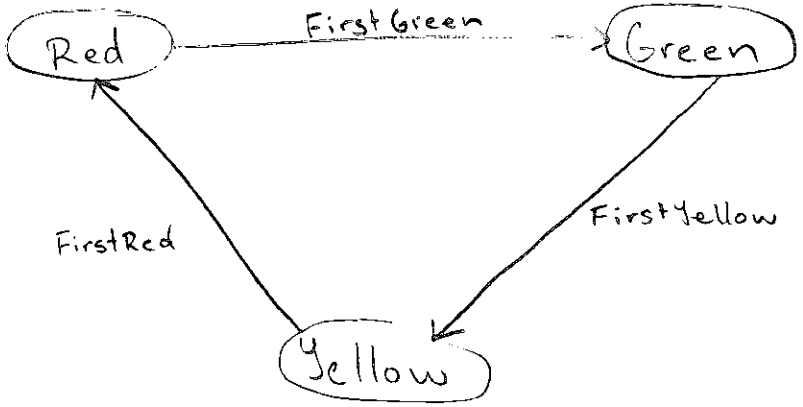
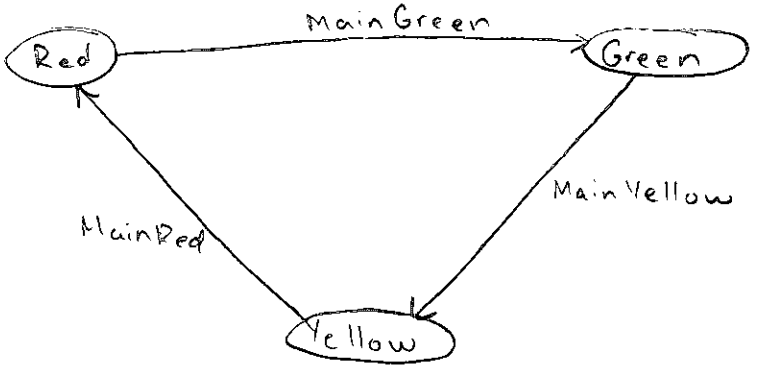
My state machine ensures that the traffic light signals will prevent collisions (assuming they are obeyed) because it ensures that in order for one light to be green, the other must be red. When one street light changes from red to green, the other light must change from yellow to red. It also ensures that if a train arrives, the lights on Main street are Red and the lights on First street are green, since First street runs parallel to Main street. My state machine also ensures that the lights will only change if there are cars waiting at the other light and after 30 seconds. This is ensured by the lock state, which will only change after 30 seconds, once the 30 seconds has passed it moves into the Ready state, where a car arriving at the other light will cause the light to transition.


Start
Arriving First,
Arriving Train/
FirstGreen
Arriving Main,
MainGreen
First
S30/FirstGreen
FirstLock
E30
FirstReady
Arriving Main [in GateUp]
S5, FirstYellow
FirstChange
E5,
FirstRed
Main
S30/MainGreen
MainLock
E30
MainReady
Arriving Train/
S5,
MainYellow
ArrivingFirst/S5
MainYellow
MainChange
E5/MainRed

First

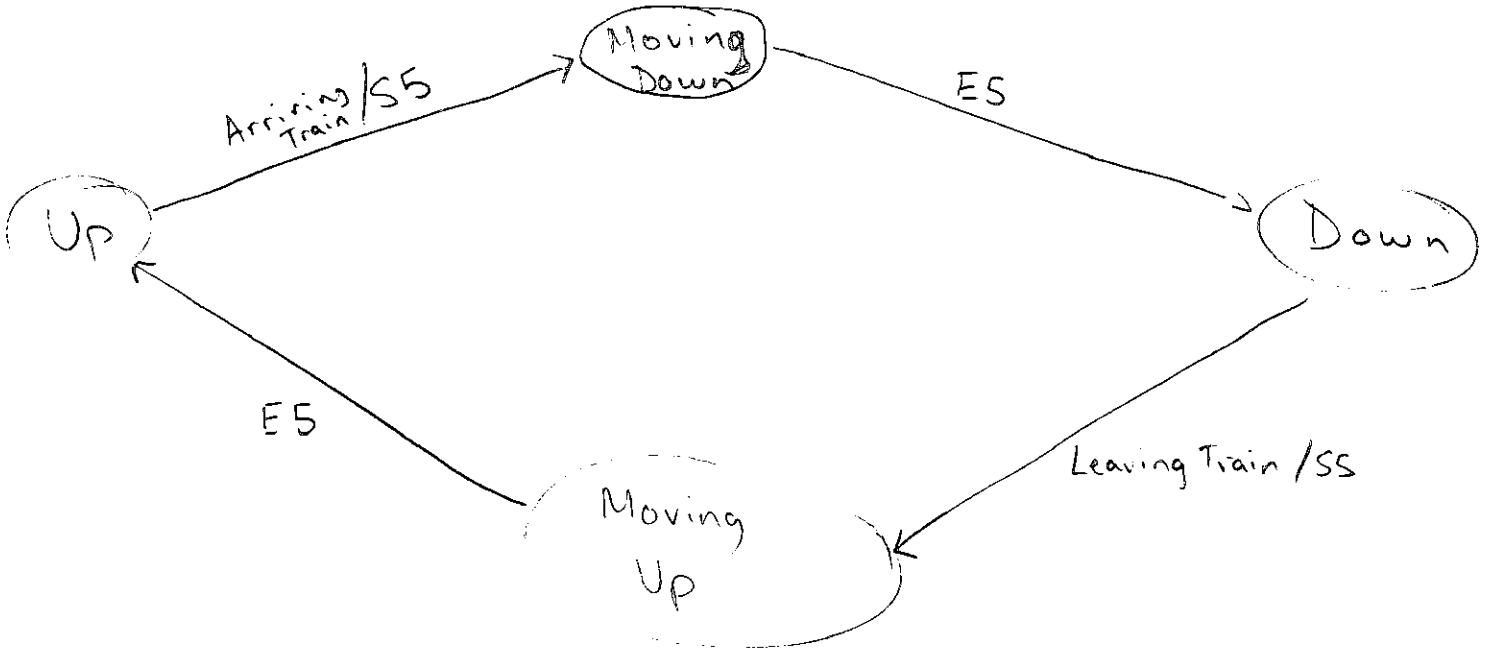


Main



Car sensors?

Gate



UML Diagram Grading Rubric

Criteria (weight)	5 Exemplary	3 Satisfactory	1 Needs Improvement	Weighted Score
Professional-ism (x2)	Document is neatly drawn. (Apart from any problems with the formalism) it could be shared with a "real-world" customer without changes.	Document is somewhat sloppy, but could be shared with a "real-world" customer after some revisions.	Document is unprofessional. It would have to be completely redrawn before sharing the document with a "real-world" customer.	
Clarity of Formalism (x4)	Diagram is well-labeled and at an appropriate level of abstraction so that a customer familiar with the problem domain could readily understand the specification.	Diagram is mostly well-labeled, with at most three cryptic labels. Diagram is generally at an appropriate level of abstraction, though a customer familiar with the problem domain might need some guidance to understand the specification.	Labels are cryptic or abstraction is used to the point that the actual specification would be obscured to all but an expert in the notation.	
Conciseness of Formalism (x4)	Specification appropriately uses the abstraction features of the notation to minimize unhelpful redundancy.	Specification may include some unhelpful redundancy, but the general requirements are still readily comprehensible	Specification is highly redundant. The volume of redundancy makes comprehension of the specification very difficult.	
Depth of Analysis (x5)	Specification covers all important corner cases. It demonstrates a deep understanding of the problem.	Specification covers many important corner cases. Some cases might be treated in an unusual manner, but such treatment is documented.	Specification treats few or no corner cases. It demonstrates just a superficial understanding of the problem.	
Correct Use of Notation (x5)	All notation used in the diagram is appropriate to the diagram type and is used correctly.	All notation used in the diagram is appropriate to the diagram type. At most two sorts of errors are made in the application of the notation.	Diagram uses notation inappropriate to the diagram type or contains a large variety of errors in the application of the notation.	
			Total Score:	95