The University of Waikato
Department of Computer Science
Dr. Robi Malik

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

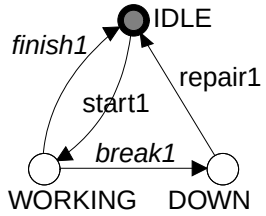
COMP 424-06A

Topics in Software Engineering

Assignment 1

Exercise 1 (5 marks)

A simple manufacturing system "small factory" consists of two machines and a buffer that can store a single workpiece. **Machine1** starts the processing of workpieces and puts them into the buffer when finished. When **Machine2** is started, workpieces are removed from the buffer and completed. The plant behaviour of **Machine1**, e.g., is given by the depicted automaton. The starting and repairing of machines are controllable events, while finishing and breakdown are uncontrollable.



Use the VALID Toolset to create a model of this system. Create the two machine automata as plants, and design additional specification automata to control the two machines such that the controlled system satisfies the following requirements.

- The **Buffer** does never overflow or underflow.
- **Machine2** has repair and return-to-service priority over **Machine1**, i.e., whenever both machines are broken, **Machine2** must be repaired first.

Use the VALID Simulator to see that your model behaves as expected. Verify that your model is controllable and nonblocking.

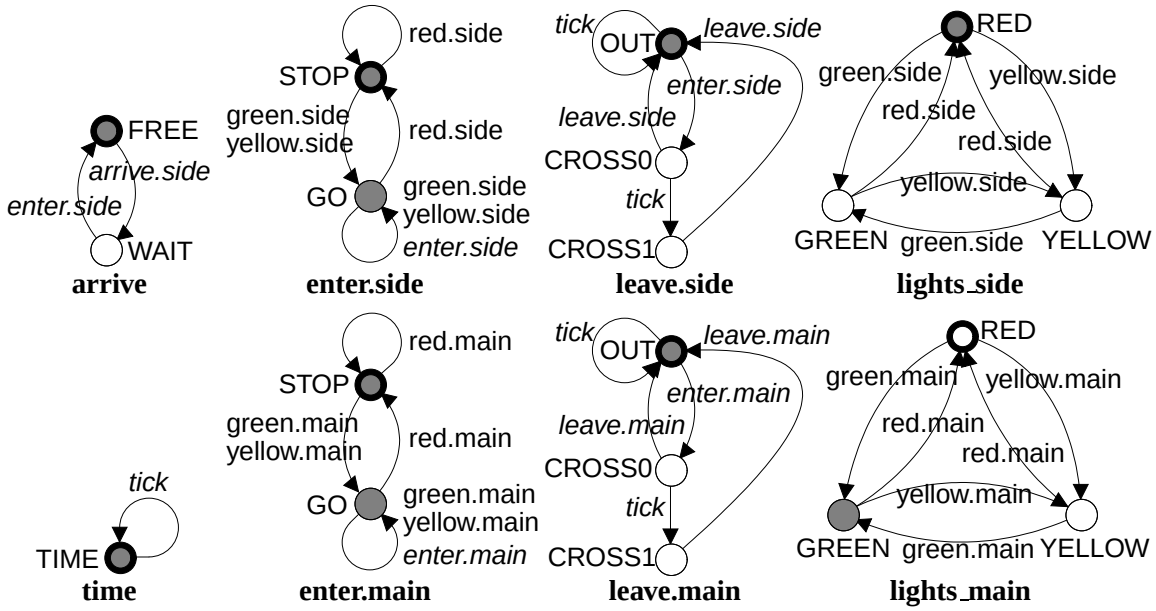
Exercise 2 (15 marks)

The intersection between a larger main street and a smaller side street is to be controlled using traffic lights. The side street is equipped with a sensor that can detect the arrival of a car, but there is no such facility on the main street.

The traffic lights can be switched using controllable events red.side, red.main, yellow.side, yellow.main, green.side, and green.main, representing a change of the side or main street traffic lights to red, yellow, or green. The uncontrollable events *enter.side*, *enter.main*, *leave.side*, and *leave.main*, indicate that a car from the side or main street enters or leaves the intersection. The uncontrollable event *arrive.side* represents the side street sensor and indicates that a car has arrived in front of the side street lights. Furthermore, the system includes a timer that sends an uncontrollable *tick* event every five seconds.

The following figure shows a finite-state machine model of the possible behaviour of this plant. The automata describe assumptions about the behaviour of cars at the intersection: cars do not enter the intersection if the lights are red, and they do not stay on the intersection for more than five seconds. Also, the side street sensor detects every car before it enters the intersection. A VALID project containing these automata can be downloaded from the COMP 424-06A course homepage

`http://cs.waikato.ac.nz/Teaching/COMP424A/`



Please take some time to look at the automata and see how they work.

Your task is to design a controller for this plant that meets the following requirements.

- The controller only uses the uncontrollable events *arrive.side*, *enter.side*, and *tick*, and the controllable events that set the lights. All other events are "unobservable" to the controller because it does not have the necessary sensors.
- Accidents are avoided. It does never happen that there are cars from the side street and the main street on the intersection at the same time.
- Lights always follow the sequence green–yellow–red–green.
- Lights stay yellow for at least five seconds, and green for at least 15 seconds.
- The side street light never stays green for longer than 20 seconds.
- Normally, the system keeps the main street lights green and the side street lights red. Only if a car is detected in the side street, the side street lights change to green as soon as possible while still satisfying all the above constraints.

Create additional specification automata in VALID and add them to the project. Verify that your model is controllable and nonblocking. Run the VALID simulator to check that the model behaves as you expect. What problems do you encounter with the last two requirements?

Note. Do not modify the given plant model. Only add specification automata to the project.

Submission

The best way of submitting a VALID project is by putting all its files into a directory, and packing the contents into a `.tar.gz` archive, which can be submitted through the assignment submission system at `http://byerley.cs.waikato.ac.nz/~tonym/submit/`.

Please add a text file `README.txt` to each archive, containing your name and student ID number, plus a brief explanation of the automata you created. For exercise 2, please also include a brief discussion of your experience with this kind of modelling and the problems encountered in the `README.txt` file.

Please submit two projects, one for each exercise. Each project should include the plant and specification automata, but no property automata used for language inclusion checks. If you have such automata, please keep them separate and explain them in your `README.txt` file.

Due date: Monday, 13th March 2006, 9:00 A.M.