The Level Crossing

David Kendall

1 Train Gate Controller

A simple example of a system with real time constraints is an automatic controller for a gate on a railway crossing. The example has been studied extensively. We consider a simple system consisting of three components: a train, a gate and a controller. Each of these can be modelled as a timed automaton (see figure 1). Timing constraints are expressed using 3 clocks: x for the train, y for the gate and z for the controller. The train advises the controller of its approach at least 2 minutes before it enters the crossing. The approach of the train is indicated by the event name `approach!`. The train is reckoned to have entered the crossing when it reaches the location labelled `IN`. The maximum delay between the events `approach` and `exit` is 5 minutes. The gate is open in location `GATE.OPEN` and closed in location `GATE.CLOSED`. The events `raise` and `lower` are used to indicate requests for service from the gate by the controller. The locations `UP` (resp. `DOWN`) indicate that the gate has been completely raised (resp. lowered). The controller idles in location `CONTROLLER.Idle`. Whenever it detects that the train is approaching, it requests that the gate should be lowered. Similarly, whenever it detects that the train has left the crossing, it requests that the gate should be raised. The complete system is expressed as the composition of the three components *TRAIN* | *GATE* | *CONTROLLER*. The safety requirement for the system is straight forward: whenever the train is in the crossing, the gate should be closed.

2 Exercises

1. Use the UPPAAL tool to construct the level crossing model exactly as shown in Figure 1.
2. Experiment with the model using the UPPAAL simulator, until you are confident that you understand both the model and the major features of the simulator.
3. How does a timed automaton (TA) differ from a standard FSA? Give examples from the model of the level crossing of:
   (a) a location,
   (b) a guard,

(c) an invariant,

(d) a synchronization label,

(e) a clock reset set (give an alternative notation for your clock reset example).

4. How does the timed automaton (TA) model show that there is a delay of at least 2 minutes between the approach of the train and its entry into the crossing?

5. How does the TA model show that the maximum delay between the approach of the train and its exit from the crossing is 5 minutes? How would you modify the model to change the maximum delay to 8 minutes?

6. What are the bounds on the time taken to:

   (a) completely raise the gate, following a raise request from the controller;

   (b) completely lower the gate, following a lower request from the controller?

7. Modify the TA model to show that it takes between 1 and 2 minutes to lower the gate and exactly 1 minute to raise the gate.

8. What is the response time of the controller to the events `approach!` and `exit!`? Ask for help if you're not sure what is meant by *response time*.

9. In the TA model, what corresponds to:

   (a) "the train is in the crossing", and

   (b) "the gate is down"?

   Is it possible for the system to reach a state satisfying (a) but not (b)? How can you check your answer?

10. In the TA model, what will happen if a train approaches when there is already a train in the crossing?

11. What would be a suitable *liveness* requirement for the system?

12. Assume that we can write down a description of the system state as follows: $((\mathit{Safe}, \mathit{OPEN}, \mathit{Idle}), (0, 1, 2))$, where the first tuple $(\mathit{Safe}, \mathit{OPEN}, \mathit{Idle})$ represents the current location for the Train, Gate and Controller components, respectively, i.e. the Train is in location Safe, the gate is in location OPEN and the controller is in location Idle; and the second tuple $(0, 1, 2)$ represents the values of the clocks x, y and z, respectively, i.e. $x = 0$, $y = 1$ and $z = 2$. A transition can be shown as, e.g., $((\mathit{Safe}, \mathit{OPEN}, \mathit{Idle}), (0, 0, 0)) \xrightarrow{3.5} ((\mathit{Safe}, \mathit{OPEN}, \mathit{Idle}), (3.5, 3.5, 3.5))$.

    Draw a trace of the system, in this style, which shows the complete passage of a train through the crossing. Any trace which respects the rules for generating discrete and timed transitions is acceptable.

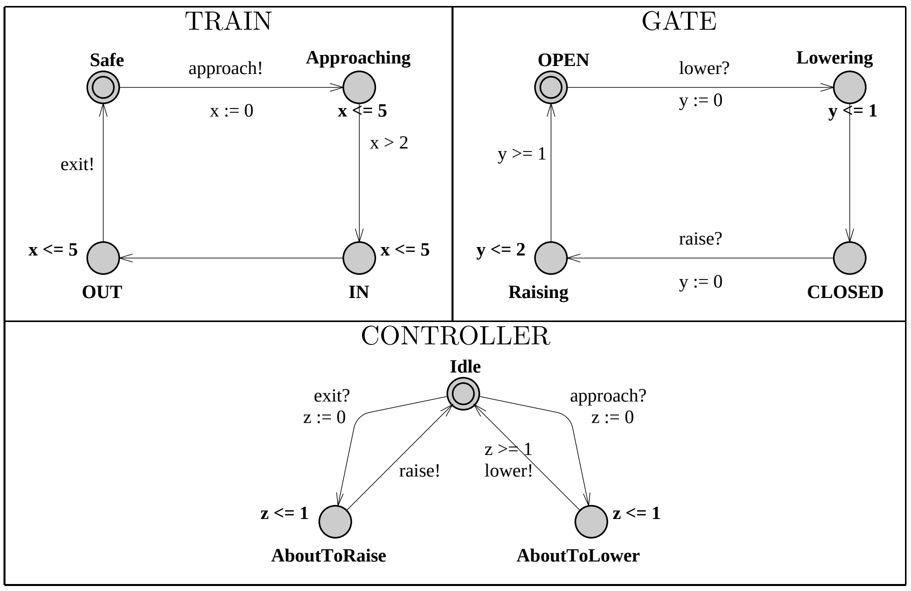

Figure 1: Train Gate Controller