Quantitative Verification Handout 2

October 27, 2016

1 Train and Gate Case Study

Construct a system modeling trains from multiple tracks crossing a bridge with a single track. The expected behaviour of the train is elaborated below.

- When the Train approaches a bridge, it sends a signal to the controller. If the bridge is occupied, the controller sends a stop signal to the train.
- If the train doesn't receive a stop signal within 10 time units, it enters the bridge within 20 time units. It takes the train 3 to 5 time units to cross the bridge.
- If the train receives a stop signal within 10 time units, it comes to a stop. When it receives a go signal from the controller, it starts moving within 15 time units and it takes atleast 7 time units to enter the bridge.

Design a controller which uses an FCFS strategy to process requests. If the bridge is free and a train requests to use it, add it to a queue. When the train leaves, remove it from the queue. If the bridge is being used, always add the train to the queue and ask it to wait until its turn.

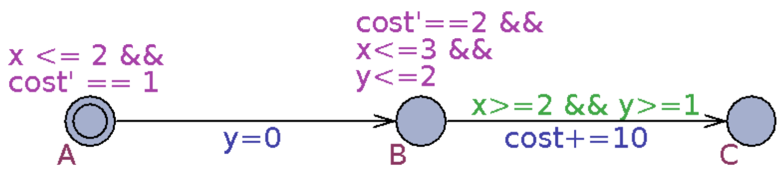
Verify the following properties

- Gate can receive (and store in queue) msg's from approaching trains.
- Train 1 can reach crossing.
- Train 0 can be crossing bridge while Train 1 is waiting to cross.
- Train 0 can cross bridge while the other trains are waiting to cross.
- There is never more than one train crossing the bridge (at any time instance).
- There can never be N elements in the queue (thus the array will not overflow).
- Whenever a train approaches the bridge, it will eventually cross.
- The system is deadlock-free.

Source: UPPAAL Demos

2 A simple Priced Timed Automata

Use UPPAAL to find the minimum cost with which the system can reach C.



3 Aircraft Landing Problem

The problem consists of a number of aircrafts that need to land on limited number of runways. Each aircraft has a preferred target landing time, corresponding to arriving at the runway with cruise speed and landing immediately. However, the aircraft can speed up and land earlier or stay longer in the air and land later if necessary. Due to extra gas used in accelerating, landing earlier has an added cost which grows linearly until some fixed earliest landing time. Late landings have an immediate constant penalty for landing late and a cost growing linearly with the amount of gas used for staying in the air until all latest possible (controlled) landing time.

Furthermore, aircrafts cannot land back to back on the same runway due to wake turbulence. Thus, there are certain minimum constraints on the separation delay between aircrafts of different types.

Your task is to use UPPAAL Cora to model a flight and a runway and run in parallel, a system which is composed of 10 flights and 2 runways. You may use `https://www7.in.tum.de/~ashok/tutorials/qv/apl-system.txt` as your Systems declaration.

Now, the problem is to assign landing times and runways to all aircrafts while respecting individual timing constraints for the aircrafts and the wake turbulence constraints.

Source: http://people.cs.aau.dk/ adavid/cora/casestudies.html, UPPAAL Cora Demos