

# Quantitative Verification Session 1

October 19, 2017

## Motivating Model Checking

We presented a factory robot (*photo/video coming soon*), the task of which was to filter red bricks while letting bricks of other colors pass. The robot was programmed (Fig. 1) using the visual programming language which was supplied along with the LEGO Mindstorms EV3 software. The first task was to figure out whether its possible to fool the robot into letting a red brick pass.

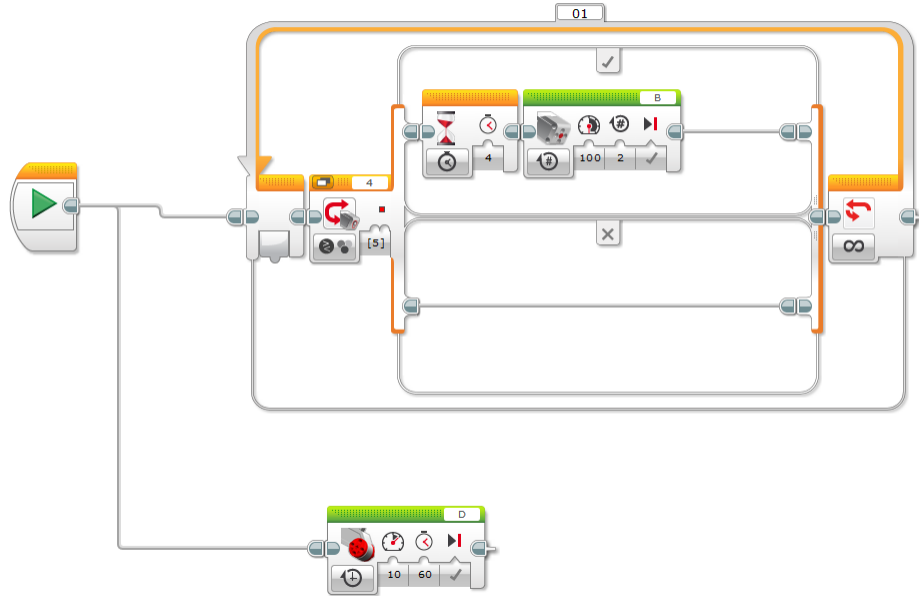


Figure 1: Program which was loaded onto the robot

We then modeled the robot compositionally using the tool UPPAAL (<http://www.uppaal.org>). Each of the logical components of the robot were modeled separately – the controller (Fig. 2), color sensor (Fig. 3b), the kick actuator (Fig. 3a) and the blocks (Fig. 4). The model in UPPAAL formal can be downloaded at <https://www7.in.tum.de/~kretinsk/teaching/legokicker.xml>.

*Note: This is just one of the ways which could be used for modeling this system. The verbosity of the models are usually decided based on the property one wants to check. In this case, we wanted to check if it is the case that always a red block would be kicked out.*

We then used UPPAAL's verifier in order to check whether there is a case where a red block can pass the belt without being kicked out ( $\text{Red1.onbelt} \rightarrow \text{Red1.kicked}$ ). The counterexample can be visualized in the simulator by setting Options -> Diagnostic Trace -> Some before checking the property.

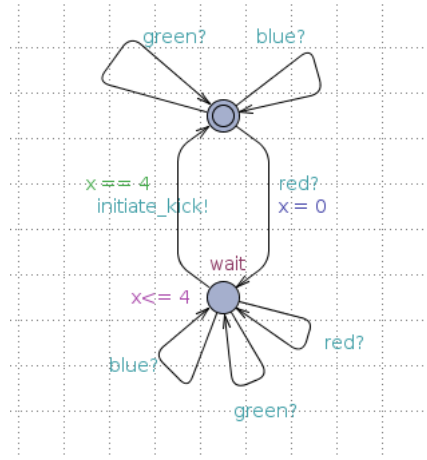


Figure 2: Controller

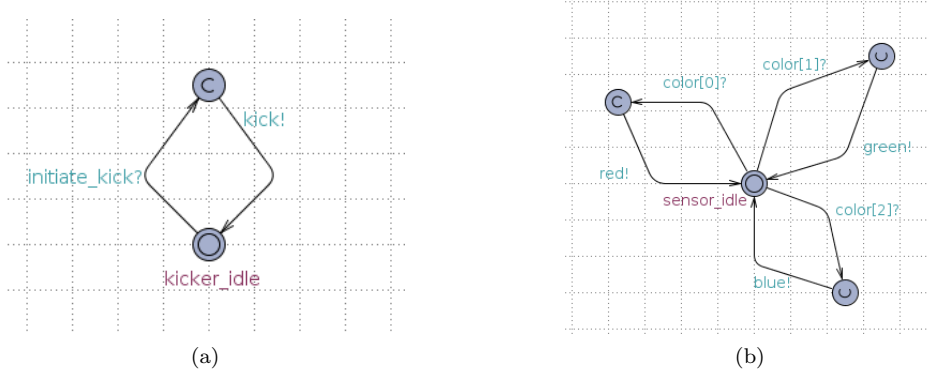


Figure 3: (a) Kick actuator and (b) color sensor

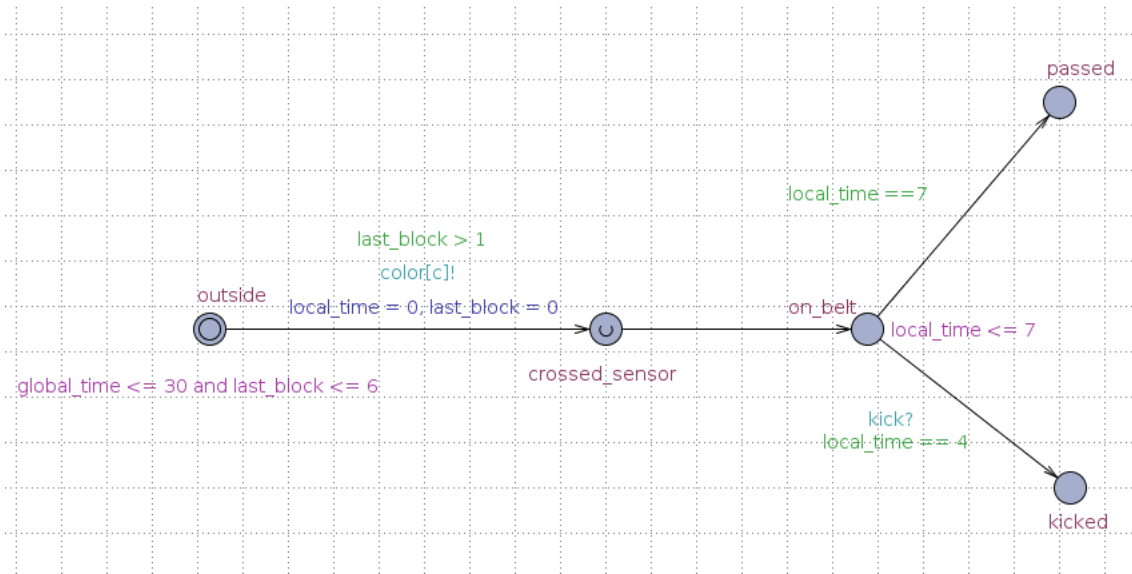


Figure 4: Block