Modeling and Analysis of Hybrid Systems - SS 2015

Series 1

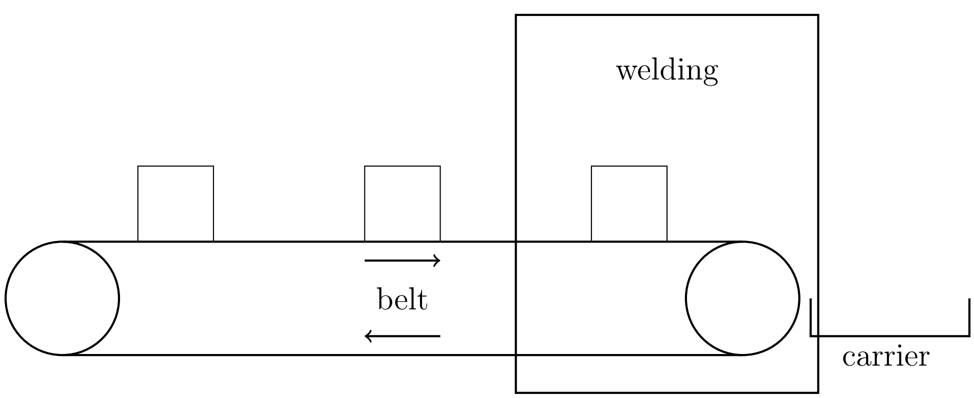
Notations

- A state of a hybrid automaton is a tuple $\langle \ell, (v_1, \ldots, v_d) \rangle$ wherein l is a location (mode) and $(v_1, \ldots, v_d) \in \mathbb{R}^d$ are values of the d (ordered continuous) model variables.
- A time transition $\langle \ell, (v_1, \ldots, v_d) \rangle \xrightarrow{\delta} \langle \ell, (v'_1, \ldots, v'_d) \rangle$ models a time delay of duration δ in location ℓ, where the values of the variables evolve continuously, starting from $(v_1, \ldots, v_d)$ and reaching $(v'_1, \ldots, v'_d)$ at time δ.
- A discrete transition $\langle \ell, (v_1, \ldots, v_d) \rangle \xrightarrow{a} \langle \ell', (v'_1, \ldots, v'_d) \rangle$ moves the control from location ℓ to location ℓ', changing the values of the continuous variables from $(v_1, \ldots, v_d)$ to $(v'_1, \ldots, v'_d)$, wherein a is the transition label.

Please use these notations in your solutions.

In the following we use typewrite font to denote `states`, and italic to denote *actions* (transition synchronization labels).

Exercise 1



Consider the following production chain:

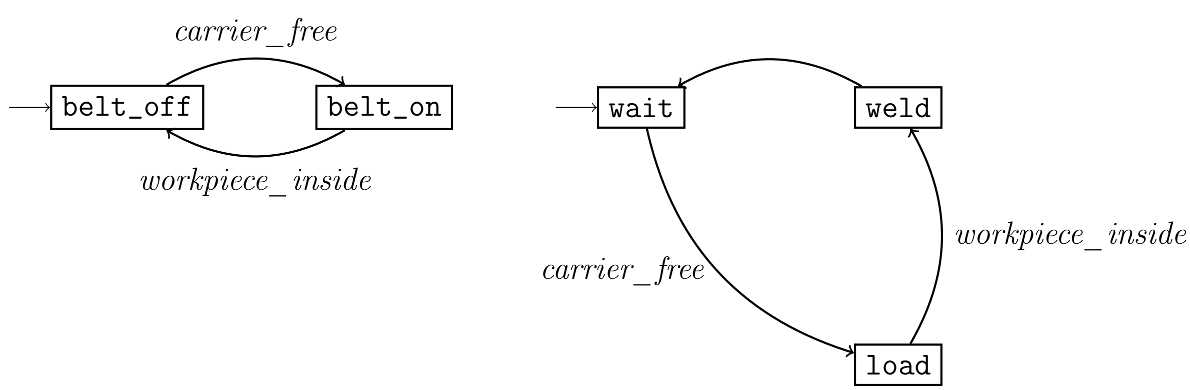
- The production task is to

- move exactly one workpiece in the welding station using the conveyor belt,
- stop the belt till the robot performs a certain job on the workpiece,
- transport the workpiece into a carrier, and
- repeat the procedure.

- A conveyor belt is controlled by a controller, and can be moving (`belt_on`) or stopped (`belt_off`). Initially, the belt is stopped.
- The belt controller and the robot have both access to two sensors *workpiece_ inside* and *carrier_free* which are enabled, when a workpiece is inside the welding station respectively if the carrier after the welding station is free.
- A welded workpiece cannot be put on a full carrier. Thus the robot has to wait (`wait`) until the signal *carrier_free* is enabled. After that the welding station can be loaded again (`load`).
- We assume that the distance between the workpieces on the belt is sufficiently large, such two workpieces cannot be in the welding station simultaneously.

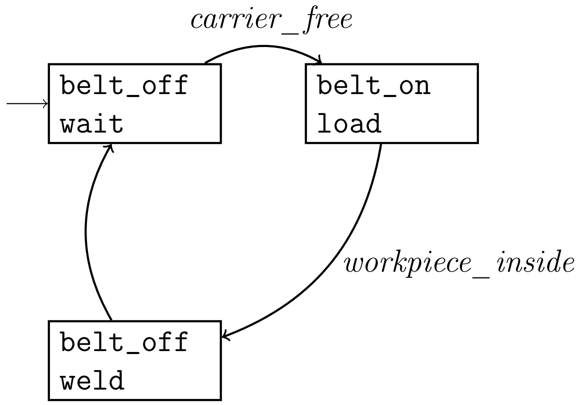
a) Give LSTS (labeled state transition system) models (i) for the belt controller and (ii) for the welding robot. Make use of synchronizing labels for the sensor states *workpiece_ inside* and *carrier_free*.

Solution:



b) Give the parallel composition of both LSTS models. Draw only reachable states.

Solution:

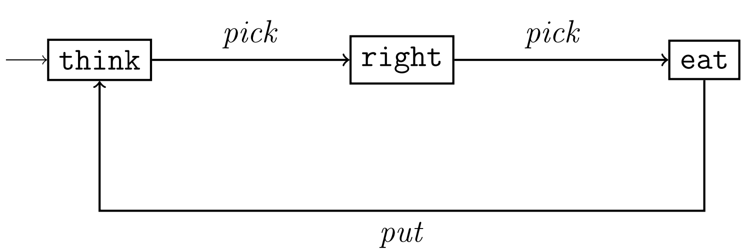


Exercise 2

a) Give an LSTS (labeled state transition system) model for the following system. Introduce labels for the actions *pick* and *put*:

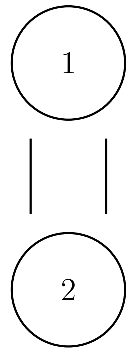
A philosopher is either thinking (`think`) or she is hungry and wants to eat. However, eating requires two chopsticks (we are located in ancient china). As our philosopher is a creature of habit, she picks one stick at a time (*pick*), starting with the right stick first. When she has the right stick (`right`), she picks the left one and can eat (`eat`). After eating, she puts back the two sticks simultaneously (*put*) and continue with thinking (`think`). As she is a born philosopher, she starts thinking right ahead.

Solution:



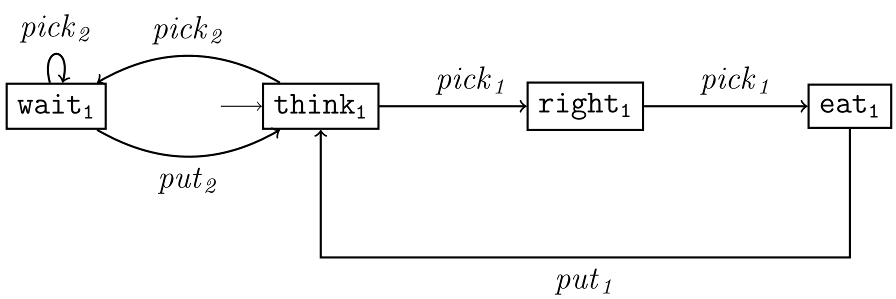
b) Now we want to extend our model:

As our philosopher is also profiting from other great minds, she usually sits with **one** friend (see drawing). Great minds like challenges and also they are short on money so every philosopher brings only one chopstick to the meetings.



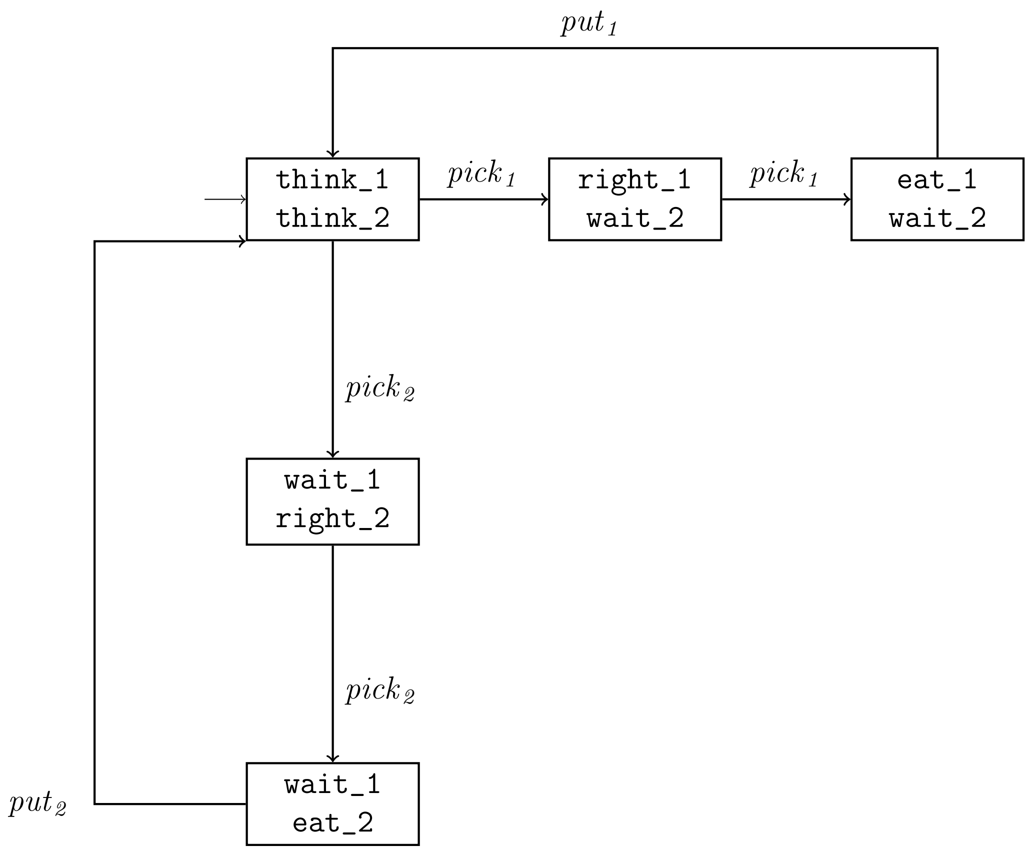
As you might have guessed, we are modelling the dining philosophers problem. Please model each philosopher by an LSTS, such that their parallel composition is deadlock-free, and such that both of the philosophers always have the possibility to eat if they are hungry after (regarding the number of actions) finite waiting. You can assume that each action is executed atomically.

Solution:



c) Give the parallel composition of the **two** philosopher models. Draw only reachable states. Argue why the system deadlock-free.

Solution:

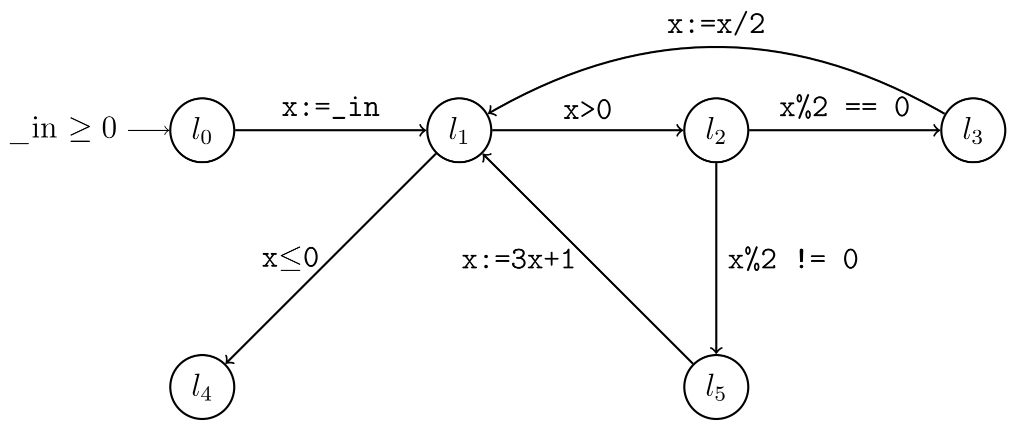


Exercise 3

Give a LTS (labeled transition system) for the following program:

```
method collatz(unsigned _in) {
        unsigned x = _in;
        while(x > 0) {
                if(x % 2 == 0)
                        x = x/2;
                else
                        x = 3x + 1;
        }
}
```

Solution:



Exercise 4

A gas burner is a device to generate a flame to heat up products using a gaseous fuel. We assume there is a gas burner, such that

(a) any leakage of it can be detected and stopped within **1** second,

(b) it does not leak initially, and

(c) it never leaks for at least **30** seconds after a leakage has been stopped.

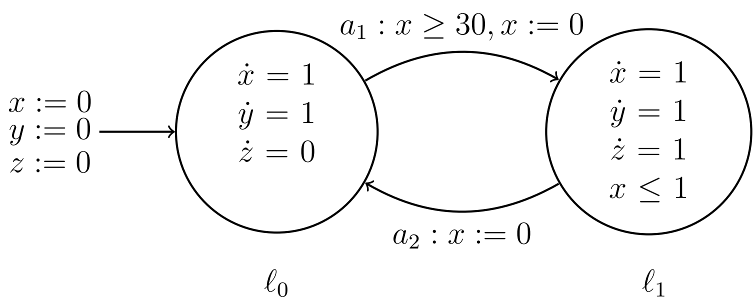
The gas burner records the cumulative leakage time and the total elapsed time.

(1) Please model the gas burner by a hybrid automaton and try to keep it as simple as possible.

(2) Is it possible to have a (cumulative) 2-second leakage time in 70 seconds? If so, please give a sample execution.

Solution:

1. The gas burner can be modeled by the following hybrid automaton.



2. Yes, it is possible. We give the sample execution as follows.

$$(\ell_0,(0,0,0)) \xrightarrow{30} (\ell_0,(30,30,0)) \xrightarrow{a_1} (\ell_1,(0,30,0)) \xrightarrow{1} (\ell_1,(1,31,1)) \xrightarrow{a_2} (\ell_0,(0,31,1))$$
$$\xrightarrow{32} (\ell_0,(32,63,1)) \xrightarrow{a_1} (\ell_1,(0,63,1)) \xrightarrow{1} (\ell_1,(1,64,2)) \xrightarrow{a_2} (\ell_0,(0,64,2)) \xrightarrow{6} (\ell_0,(6,70,2))$$

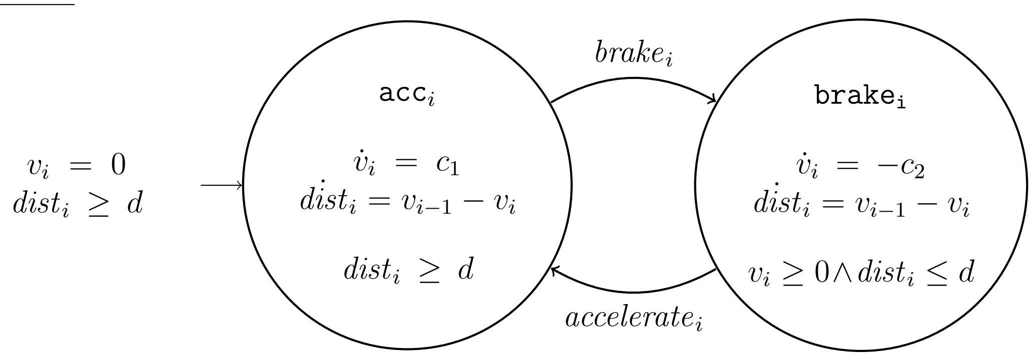
Exercise 5

a) Give a hybrid automaton model for the ith car ($i > 1$) in the following highway car platoon:

- The car i has velocity v_i.
- If the ith car's distance $dist_i$ to the $(i-1)$st car ahead is at least $d > 0$ then the car accelerates ($accelerate_i$, $\texttt{acc}_i$) constantly with an acceleration $\dot{v}_i = c_1$, $c_1 > 0$, as long as $dist_i$ remains at least d.
- If the ith car's distance $dist_i$ to the $(i-1)$st car ahead is at most d then the car brakes ($brake_i$, $\texttt{brake}_\texttt{i}$) constantly with negative acceleration $\dot{v}_i = -c_2$, $c_2 > 0$, as long as its velocity is non-negative and $dist_i$ remains at most d.

Initially our car is standing ($v_i = 0$) $dist_i \geq d$ meters away from its predecessor and is in the accelerating mode. The velocity of the predecessor car is v_{i-1}.

Solution:



b) What happens, when eventually $dist_i = d$ holds?

Solution:

The car i can constantly switch between its two locations infinitely often in finite (even zero) time. This behavior is called Zeno-behavior.

c) For safety reasons, the car company has decided to introduce a controller for a breaking light, which is (of course) synchronized to the car's behavior and enables the breaking light one second longer (hysteresis) than the braking process of the car. Give a hybrid automaton model for this controller, which makes use of the synchronization labels *accelerate* and *brake* from the previous tasks and where the location labels reflect the state of the breaking light (i.e. $\texttt{light}_{\texttt{on}}, \texttt{light}_{\texttt{hyst}}, \texttt{light}_{\texttt{off}}$).

Solution: