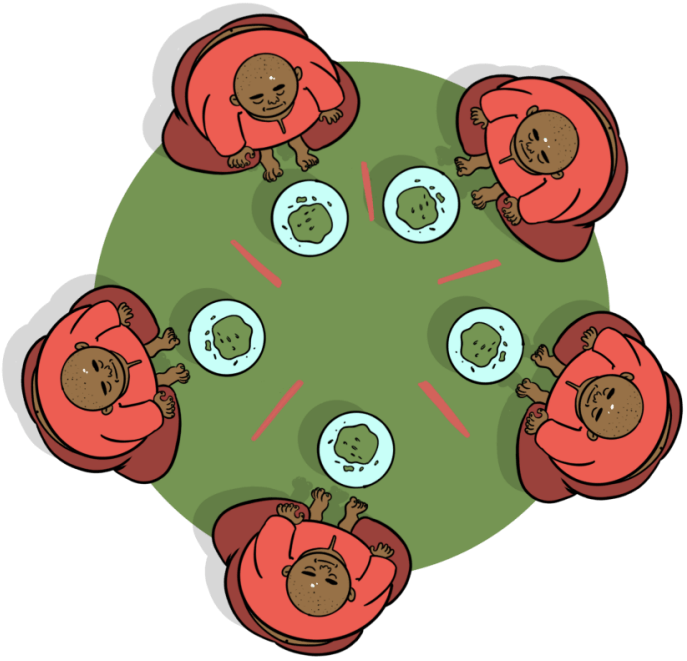
Model Checking Homework 6



Deadline: 06 Mai 4:00pm
Send solution to: modelchecking@iaik.tugraz.at

The Dining-Philosophers Verification-Problem

We consider a variant of the dining philosophers problem. There are n philosophers sitting at a round table. There is one chopstick between each pair of adjacent philosophers. Because each philosopher needs two chopsticks to eat, adjacent philosophers cannot eat simultaneously. We are interested in schedulers that use input variables h_i signifying that philosopher i is **hungry** and output variables e_i signifying that philosopher i is **eating**.

[4 Points] Formulate the following requirements.

Many specifications consist of environment assumptions and system guarantees. Formulate the following guarantees and assumptions in LTL:

- Guarantee 1: An eating philosopher prevents her neighbours from eating.
- Guarantee 2: An eating philosopher eats until she is no longer hungry.
- Guarantee 3: Every hungry philosopher eats eventually.
- Assumption: An eating philosopher eventually loses her appetite.

Give the final specification of the system that specifies **that if all assumptions are satisfied, then all guarantees need to be satisfied as well**. (Note that for n philosophers, we have *n* assumptions and *3n* guarantees.)

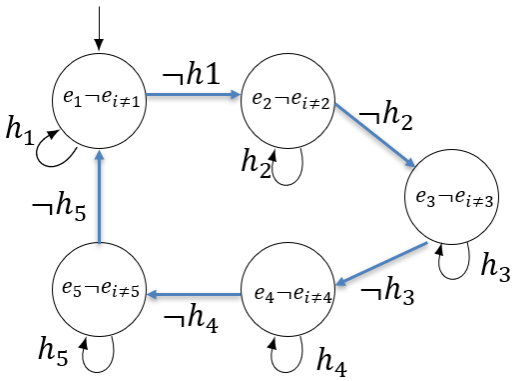
Which properties are **safety properties** and which are **liveness properties**?

[6 Points] Design a ROBUST system

If all assumptions are satisfied, a correct system has to satisfy all guarantees. However, a **robust** system should act in some "reasonable" way, even if the environment does not fulfil the assumptions. We define that a system is robust with respect to liveness properties, if for any number of environment liveness assumptions that is violated, a minimal number of system liveness guarantees must still be satisfied.

Consider the dining philosophers example with **n = 5** philosophers given in the introduction and a system **D1** modelled **as Moore Machine** given below. ($\neg e_{i\neq 1}$ is short for $\neg e_2\ \neg e_3\ \neg e_4\ \neg e_5$)
D1 always lets one philosopher eat until she is not hungry anymore and then moves to the next hungry philosopher in a round robin manner. **If one philosopher is hungry forever, then no other philosopher gets to eat again.** Thus, the violation of **one** liveness assumption leads to the violation of **four** liveness guarantees.

Moore Machine D1:



Your Task: Design a system as Moore machine or Mealy machine for 5 dining philosophers that is

- **Correct**, i.e., it satisfies the specification,
- **and Robust** in the sense that if one philosopher is hungry forever, she eats forever and the only two other philosophers starve**.**