BCS2213 – Formal methods

Teaching assignment 4. Liveness and Fairness.

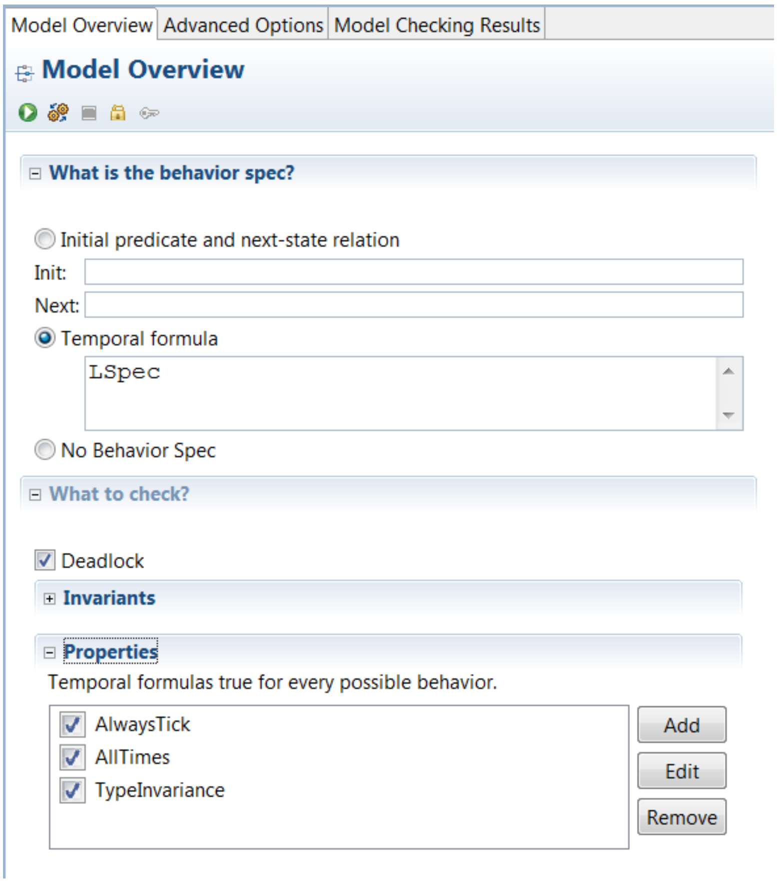
1. Run TLA+ Toolbox.

2. Develop **LiveHourClock** TLA specification, as shown below

```
-------------------------------- MODULE LiveHourClock --------------------------------
EXTENDS HourClock

(* Conjoin the specification HC with the Week Fairness condition *)

LSpec == HC /\ WF_hr(HCnxt)


(* Define some Liveness and Fairness properties that LSpec satisfies. *)

(* Asserts that infinitely many <<HCnxt>>_hr steps occur. *)

AlwaysTick == []<><<HCnxt>>_hr


(* Asserts that, for each time n in 1..12, hr infinitely often equals n. *)

AllTimes == \A n \in (1..12) : []<>(hr = n)


TypeInvariance == []HCini

(* The temporal formula asserting that HCini is always true. *)

(* It is stated in this way to show another way of telling TLC to check an invariant. *)

---------------------------------------------------------------------------

(* LSpec – is a liveness specification, combining 3 temporal properties *)

THEOREM LSpec => AlwaysTick /\ AllTimes /\ TypeInvariance

============================================================
```

3. Note, **LiveHourClock** extends **HourClock** module you developed on first lab, so HourClock.tla should be in one folder with LiveHourClock.tla

4. To check temporal properties (AlwaysTick, AllTimes, TypeInvariance) by TLC you need add them in the Properties window on the Model overview page (see figure below).



Model Overview | Advanced Options | Model Checking Results

Model Overview

What is the behavior spec?

○ Initial predicate and next-state relation

Init:

Next:

● Temporal formula

```
LSpec
```

○ No Behavior Spec

What to check?

☑ Deadlock

Invariants

Properties

Temporal formulas true for every possible behavior.

☑ AlwaysTick

☑ AllTimes

☑ TypeInvariance

5. Modify your specification to violate AllTimes property. For it, change the specification of HourClock in order it can takes e.g. only 10 steps. Check the model.

Note, you can violate AllTimes property in the different ways, e.g. specify that HourClock ticks with steps 2 (i.e. each 2 hours – 2, 4, 6, 8, …). Check the model.

6. Restore the correct specification of HourClock.

7. Specify the new temporal property, that when **hr** is equal to 1, it implies that **hr** eventually will have value 2. Check it. Next comment the property in your module (in order it remains available for evaluation).

8. Is it possible to prove the property that when **hr** is equal to 1, it implies that **hr** always will have value 2. Check it. Comment the property in your module.

9. Check, if this property *infinitely often* true. Comment this property in your module.

10. Check, if this property *eventually always* true. Comment this property in your module.

11. Please upload your labsheet with commented properties and possible explanations into Moodle.