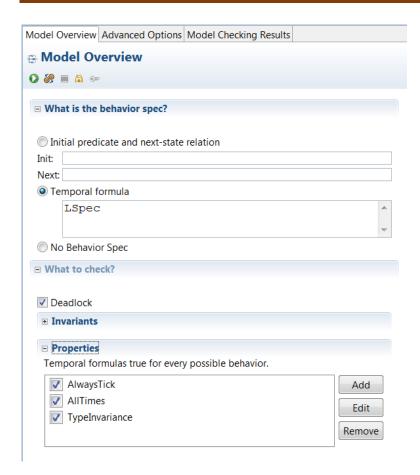**BCS2213 – Formal methods**

Teaching assignment 6. Liveness and Fairness.

1. Run TLA+ Toolbox.

2. Develop **LiveHourClock** TLA specification, as shown below (please name module Lab_6_<Your_ID>)

------------------------------- **MODULE LiveHourClock** -------------------------------

EXTENDS HourClock

(* Conjoin the specification **HC** with the **W**eek **F**airness 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 the first lab, so HourClock.tla **should be in one folder** with LiveHourClock.tla

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

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. Comment your findings in TLA module.

6. Restore the correct specification of HourClock.

7. Specify the New property, such that

```
New == (hr = 1) => (hr = 2)
```

What is the meaning of it? Is it true statement? Check the model. Comment your findings in TLA module.

8. Change the New property, in order it becomes *temporal*: when **hr** is equal to 1, it implies that **hr** *eventually* will have value 2. Check it. Comment your findings in the module (all your comments are needed for evaluation).

8.1. Check the property that when **hr** is equal to 1, it implies that **hr** *always* will have value 2. Comment your findings.

8.2. Check, if the property *infinitely often* true. Comment your findings.

8.3. Check, if the property *eventually always* true. Comment your findings.

9. Specify the property, that there are *exist* a value *n* from 1 to 12, which equal to *hr*. Check it and comment (please not delete the property, just comment it).

9.1. Modify the property, that there are NOT *exist* a value *n* from 1 to 12, which equal to *hr*. Check and comment.

9.2. How to make this property temporal? Please give an example and check it.

10. Specify an *action* temporal formula, that on the step when **hr** is equal to 1, it implies that on the next step **hr** *always* have value 2. Check and comment this property.

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