
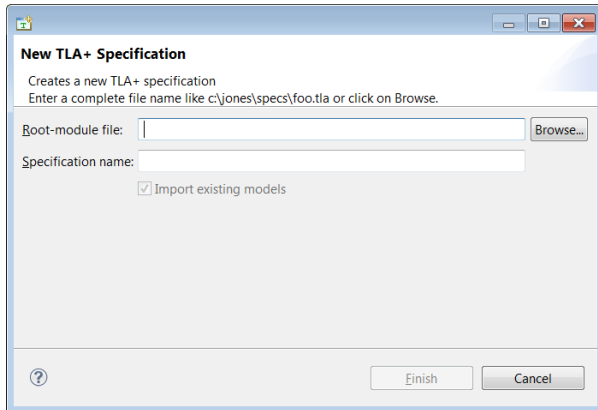
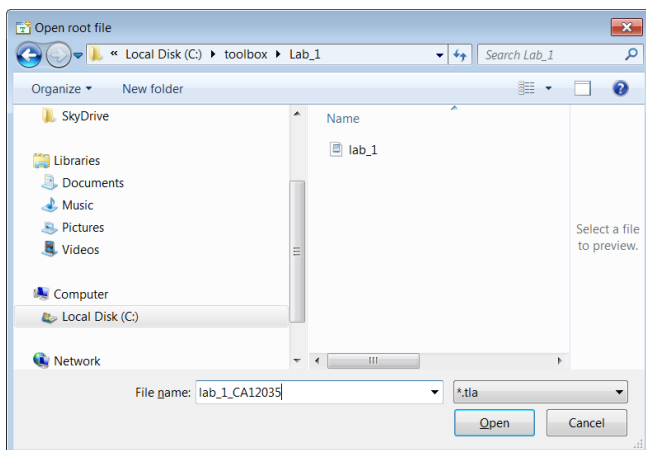


**BCS2213 – Formal methods****Teaching assignment 1.** Specification of a simple system using TLA+ Toolbox.

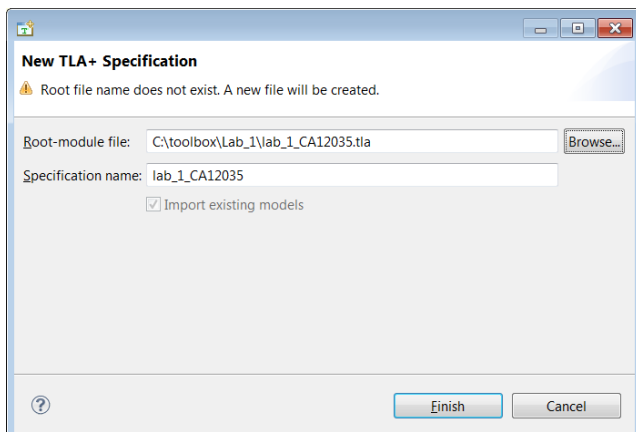
1. Run TLA+ Toolbox and click on  button (left panel) to open **Spec Explorer** window.
2. Inside **Spec Explorer** window click by the right mouse button and choose from popup menu “New Specification”.

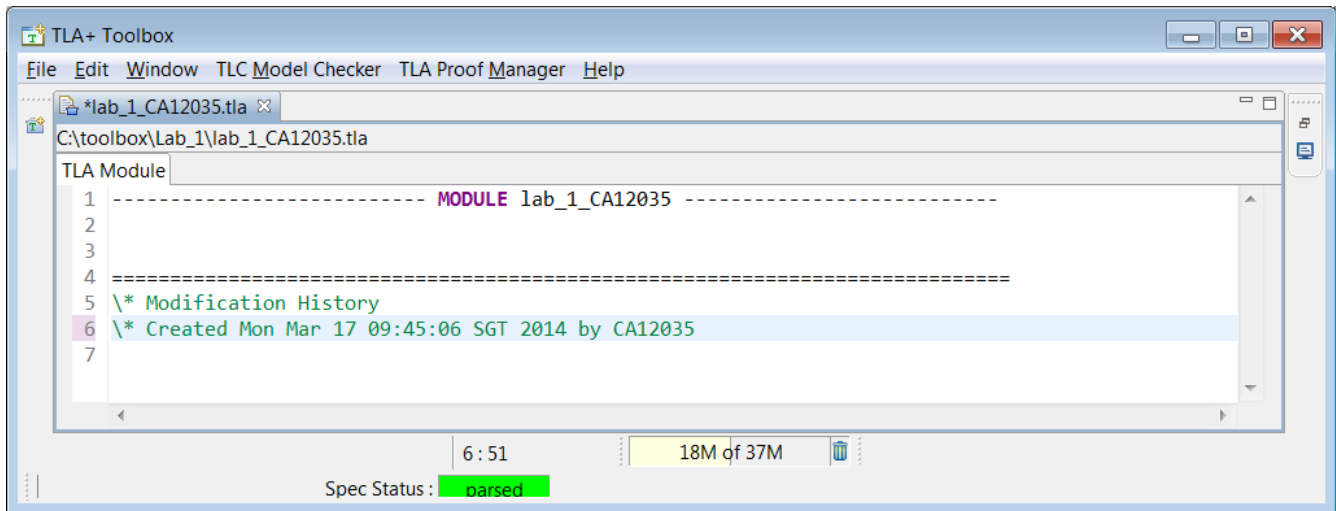


3. Click **Browse...** and in the Formal Methods folder create your personal folder with your student Id as a name.
4. Inside this folder create the folder with the name “lab\_1” and next specify the file name “lab\_1\_<your\_ID>”, an example is shown on the figure below (to the file the extension .tla will be automatically added).

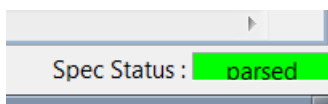


5. Click **Finish**. As result, you will see newly created TLA module lab\_1\_<your\_ID>.

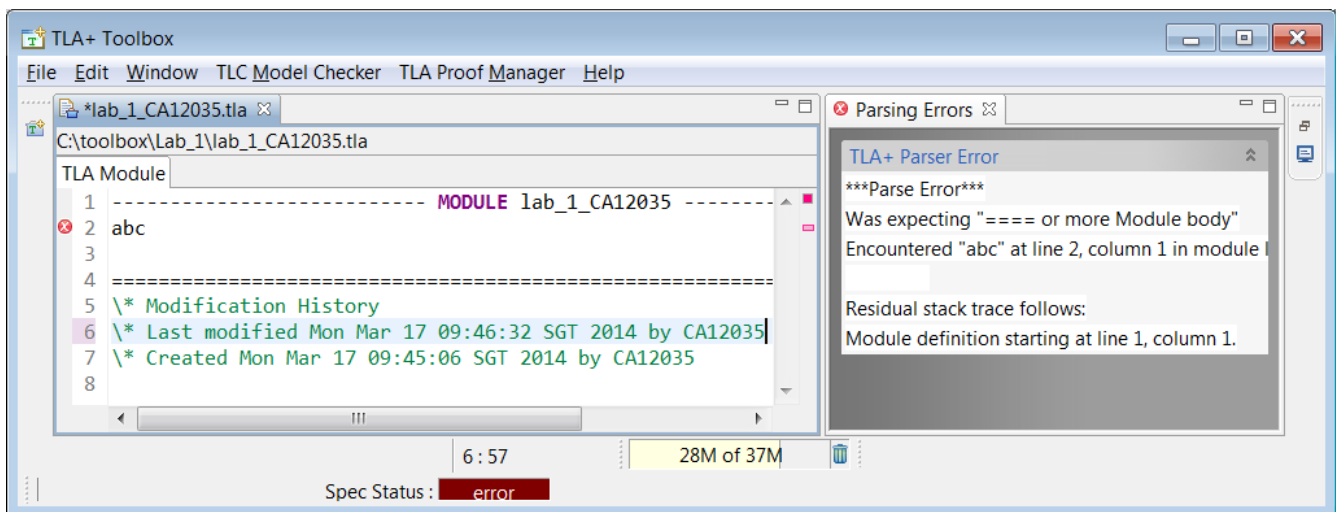




6. Have a look on the status of your module in its right bottom corner. It is parsed and if module contains no errors, the status is highlighted by green colour.



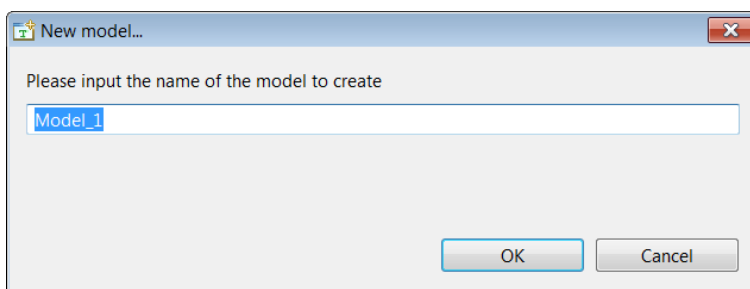
7. Write “abc” in the line 2 and reparse module (use File -> Parse Module or press Ctrl+R). You will see the error message



8. Delete the erroneous “abc” and reparse the module to have the “green” status.

9. Create the new model of specification in TLC Model Checker (use the main menu item TLC Model Checker -> New model...).

By default the model name is “Model\_1”






10. Press green arrow button  to run TLC (Temporal Model Checker) on the model.

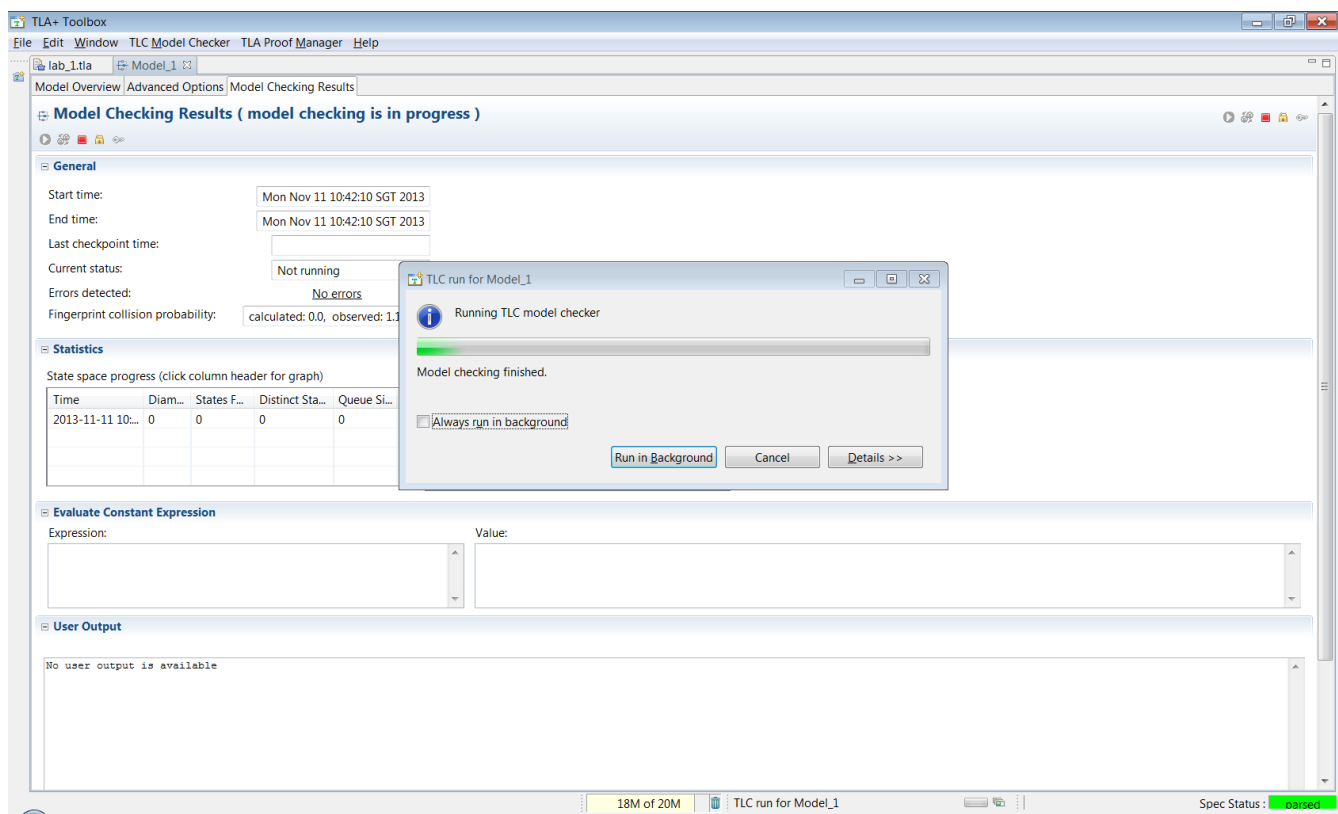
The Toolbox uses the TLC to check the current specification. More precisely, TLC checks a *model* of the specification.

Model editor has three pages:

- Model Overview Page
- Advanced Options Page
- Model Checking Results Page

You can edit these pages in the usual way. Sections can be opened or closed by clicking the + or - . When entering text in fields, you can use your system's standard editing commands.

The Toolbox does some validation of the model as you edit it. If that validation finds no errors, clicking on *validate* (  ) or *run* (  ) checks for other errors. The Toolbox reports any errors in the model that it finds by placing error balloons like this  near the part of the model containing the error. Moving the mouse cursor on top of an error balloon will raise a message explaining the error. The Toolbox also puts an *Errors Detected* field at the top of the page. Moving the mouse cursor on top of it raises all the error messages for the page.



11. Learn the results of evaluation. For your empty model TLC have found zero states and zero distinct states (Statistics information). Note,

**States Found** - The total number of states TLC has examined so far.

**Distinct States** - The number of distinct states among the states found.

12. Specify the HourClock system, following the next TLA definitions in TLATEX format

MODULE *HourClock*EXTENDS *Naturals*VARIABLE *hr* $HCini \triangleq hr \in (1 \dots 12)$  $HCnxt \triangleq hr' = \text{IF } hr \neq 12 \text{ THEN } hr + 1 \text{ ELSE } 1$  $HC \triangleq HCini \wedge \Box[HCnxt]_{hr}$ THEOREM  $HC \Rightarrow \Box HCini$ 

13. Please note, in TLA+ Toolbox you need use the ASCII equivalent of TLA notation (See the next figure).

```

----- MODULE HourClock -----
EXTENDS Naturals
VARIABLE hr
HCini == hr \in (1 .. 12)
HCnxt == hr' = IF hr # 12 THEN hr + 1 ELSE 1
HC == HCini /\ [] [HCnxt]_hr
-----
THEOREM HC => []HCini
=====

```

Explanations:

- the variable **hr** represent the clock's display from 1 to 12
- the initial predicate is formula **HCini**, specifying the possible value of **hr**
- the next-state relation **HCnxt** is a formula expressing the relation between the values of **hr** in the previous state and the next state of a system
- the next-state relation is that **hr'** equals **hr+1** except **if** **hr** equals **12**, in which case **hr'** should equal **1**.
- **HCnxt** is an ordinary mathematical formula, except that it contains primed and unprimed variables. Such a formula is called an **action**.
- an action formula (**HCnxt**) can be true or false of a step.
- the specification of the hour clock is the definition of temporal formula **HC**, including the definitions of the both formulas **HCnxt** and **HCini**

$$HC \triangleq HCini \wedge \Box[HCnxt]_{hr}$$

- to express **HC** we use the TLA operator  $\Box$  (pronounced box). The temporal formula  $\Box F$  asserts that formula **F** is always true.
- so **hr** should be an integer from 1 through 12 in every state of any behaviour, satisfying the clock's specification, i.e. the formula **HC**

- a temporal formula satisfied by every behavior is called a theorem, so  $HC \Rightarrow \Box HCini$  is a theorem. Or, in Tlatex notation

THEOREM  $HC \Rightarrow \Box HCini$

14. Specify in the Model overview page *what* to check.

There are two ways to write the behaviour spec:

#### Init and Next

A pair of formulas that specify the initial state and the next-state relation, respectively

OR

#### Temporal formula

A single temporal formula of the form  $Init \wedge \Box [Next]_{vars}$ , where  $Init$  is the initial predicate,  $Next$  is the next-state relation,  $vars$  is the tuple of state variables, e.g.  $HCini \wedge \Box [HCnxt]_{hr}$

Specify behaviour by initial predicate (HCini) and next state relation (HCNext).

☒ What is the behavior spec?

☒ Initial predicate and next-state relation

Init:

Next:

☐ Temporal formula

☐ No Behavior Spec

15. Run TLC on the HourClock specification and see results.

16. Make an error in specification by modifying HCini

$HCini == hr \in (1 .. 11)$

and check the results with TLC. What is your explanation?

17. Write specification in the alternative form (you can do it in the same file or open new specification)

$$HCnxt2 \triangleq hr' = (hr \% 12) + 1$$

$$HC2 \triangleq HCini \wedge \Box [HCnxt2]_{hr}$$

Where % is the modulus operator.

18. Check the HC2 specification.

19. Click on the + **What to check** in the model overview. There are three kinds of properties of the behaviour spec that TLC can check:

**Deadlock.** A *deadlock* is occur in a state for which the next-state relation allows no successor states. Termination is deadlock that is not considered an error. If you want the behavior spec to allow termination, then you should uncheck the deadlock option.

**Invariants.** An invariant is a state predicate that is true of all reachable states--that is, states that can occur in a behavior allowed by the behavior spec. You can include a list of invariants. The checking of each invariant can be enabled or disabled by checking or unchecking its box.

**Properties.** TLC can check if the behavior spec satisfies (implies) a temporal property, which is expressed as a temporal-logic formula. You can specify a list of such properties, each with a check-box for enabling or disabling its checking.

20. Modify your specification by adding variable min (minutes). Please note, you need modify initial and next state predicates. Check with TLC, how much states you have now.

21. Upload results of your work (i.e. the file lab\_1\_<your\_ID>.tla) into Moodle. If you have several files, please zip it into lab\_1\_<your\_ID>.zip.