IFT 6755 -H2019 | PROJECT 3: TEMPORAL LOGIC, MODEL CHECKING, BDD

The project's due date is Tuesday April 30th, 24:00 AoE. Submit your solutions on StudiUM as a ZIP file containing all relevant PDFs, text, source files and screenshots.

Provide source files with the full plain-text listings when appropriate. Your ZIP file should also contain a README.txt describing exactly where I can find the answer for each part.

It is best if you typeset your answers with LaTeX. If you choose to do the exercises by hand, make sure your handwriting is very clear (no cursive!) and submit a legible scan of your solution.

PART 1 (20%) CTL

Solve the following exercises from the book (Huth & Ryan, Logic in Computer Science, Second Edition):

1. Page 247, Exercise 2 a,f,g
2. Page 248, Exercise 6 a, b (i, iii, iv, vi, vii)
3. Page 249, Exercise 9
4. Page 249, Exercise 10 a,b,c,e,h

PART 2 (30%) NUSMV

1. Using NuSMV, check whether the Kripke Structure in Figure 3.40 on page 248 of the book satisfies the CTL properties in Exercise 6, page 248.
2. Using NuSMV, check whether the Kripke Structure in Figure 3.41 on page 249 of the book satisfies the CTL properties in Exercise 8, page 249.

You can download NuSMV from: http://nusmv.fbk.eu/

PART 3 (20%) BDD[1]

Solve the following exercises from the book (Huth & Ryan, Logic in Computer Science, Second Edition):

1. Page 399, Exercise 6.2.2
2. Page 399, Exercise 6.3.2
3. Page 400, Exercise 6.4.3

[1] Maybe this will be helpful: https://tex.stackexchange.com/questions/258788/how-can-i-draw-these-pictures-with-tikz

PART 4 (30%) MODEL CHECKING WITH ALLOY

PART 4.1: TWO-BIT COUNTER

Fill in the gaps (labelled with "TODO") in the file **tbc.als** such that, using Alloy (version 5) it performs the model checking example of the two-bit counter from slide 14 of the slide deck "07-Model Checking". Show that the safety property is satisfied for an Alloy check with global bound 2 (i.e., Ω(2) is UNSAT) , but not for 3 (i.e., Ω(3) is SAT).

PART 4.2: SINGLE PROCESS EXAMPLE

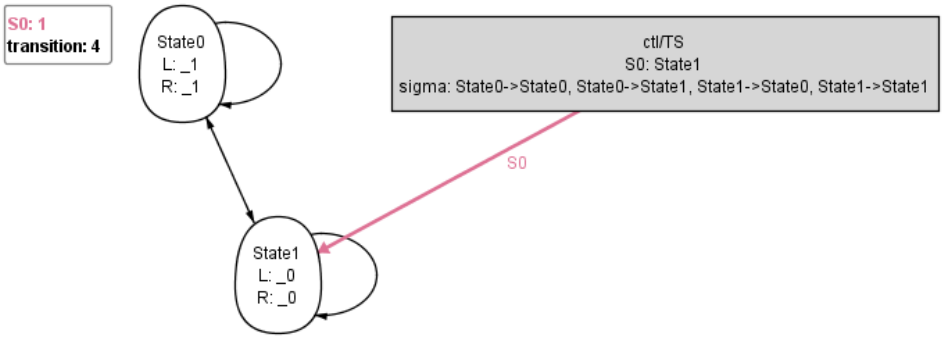
Use the same specification structure as for the two-bit counter to express the "Single Process" example (from slide 36 of the same slide deck) and check the CTL property: **AG(request -> AF state=busy)**, appropriately translated for your encoding.

IMPORTANT INFORMATION

Really important: Make sure to have the supplied file **ctl.als** (that contains the CTL module) in the same folder.

You can find more examples of usage of the CTL module at https://cs.uwaterloo.ca/~nday/models/TCMC-in-Alloy/

You are encouraged to look deeper in application of the CTL module to the "Address book" example. The example itself is thoroughly discussed in the "Dynamic Modelling" Alloy tutorial, which you are also encouraged to study: http://alloy.lcs.mit.edu/alloy/tutorials/day-course/s4_dynamic.pdf



Use the supplied theme file **kripke.htm** when debugging your spec. In this theme, states are round shapes, and transitions are black arrows. The grey box "ctl/TS" shows the mapping that has been made between your state spec and the CTL module. The red arrow labelled "S0" shows which state(s) are considered initial by the CTL module.

If you project over the "ctl/TS" signature, you should be seeing the relationship "sigma" defined by the CTL module follow exactly your own transition relation. In this projection, the initial state is shown using a hexagon.

If you feel confident enough that you haven't broken the mapping you could completely hide the "sigma" arrows to get a clean view of your Kripke structures.

For Part 4.2, modify the theme to correctly show the labelling of states by variables.