CS 3SD3

Assignment #3. Due December 5 (Wednesday), 2018, 23:59 via the course' svn depository. Do not hesitate to discuss with TA or instructor all the problems as soon as you discover them. The deadline for this assignment cannot be extended. Start early!

Instructions: For the solutions that involve FSPs you should provide a plain text file with the corresponding LTSA source code (properly commented). The LTSA solution for each one of the questions must be in a separate file. For questions that do not involve using LTSA or producing FSPs, hand drawn pictures are allowed, but a solution should be in PDF format (you may do a scan or photo). You must upload your solutions in the course' svn repository. *Any problem with svn repository please discuss with Duy Vu <vud1@mcmaster.ca>, a TA for this course.*

For the implementation questions you should provide a Java program together with the corresponding instructions on how to compile it and run some test cases; good programming style will also be marked.

Furthermore, for each one of the questions involving Java, you must also submit either a plain text file or PDF file explaining in what way your Java program implements your model. Use properly structure diagrams for your FSP processes and class diagrams for your Java programs in your explanation.

1. A lift has a maximum property of ten people. In the model of the lisft control system. Passengers entering a lift are signalled by an `enter` action and passengers leaving the lift are signalled by an `exit` action.

    a. Provide a model of the lift in FSP.

    b. Specify a safety property in FSP which wen composed with the lift check that the system never allows the lift it controls to have more than ten occupants.

2. Specify a safety property for th car park problem of Chapter 5 of the textbook and Lecture Notes 7, which asserts that the car park does not overflow. Specify a progress property which asserts that cars eventually enter the car park. If car departure is lower priority (as defined by the textbook) than car arrival, does starvation occur?

3. The net model considered in Lecture Notes 12 assumed that each process can either read or write. Suppose that we have *three* kind of processes, *readers* that can only read, *writers* that can only write, and *rw-processes*, that can both read and write. Assume we have *n readers, m writers* and *k rw-processes*.

    a. Provide a Petri nets solution in the style of Lecture Notes 12 for this version of Readers and Writers problem. The solution should be for an **arbitrary** n, m, and k, **not** for any specific values of n, m and k (as for example n=3, m=2, k=3).

    b. Prove that your solution is deadlock-free by mimicking the proof of Proposition from page 27 of Lecture Notes 12.

4. A self-service gas station has a number of pumps for delivering gas to customers for their vehicles. Customers are expected to prepay a cashier for their gas. The cashier activates the pump to deliver gas.

   a. Provide a model for the gas station with *N* customers and *M* pumps. Include in the model a range for different amounts of payment and that customer is not satisfied (`ERROR`) if incorrect amount of gas is delivered.

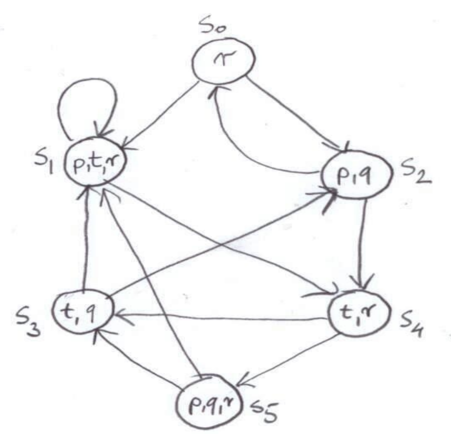
   b. Specify and check (with *N*=2, *M*=3) a safety property FIFO (First In First Out), which ensures that customers are served in the order in which they pay.

   c. Provide a simple Java implementation for the gas station system with *N*=2, *M*=3.

5. Extend the master-slave model of section 9.8 of the textbook and Lecture Notes 16 to cater for N slave processes.

6. This question deals with Model Checking.

   (a) Consider the system *M* defined below:



Determine whether $M, s_0 \models \varphi$ and $M, s_2 \models \varphi$ hold and justify your answer, where φ is the LTL or CTL formula:

(i) $\neg p \Rightarrow r$

(ii) $\neg$ EG r

(iii) E(t U q)

(iv) F q

(b) Express in LTL and CTL: 'Event *p* precedes *s* and *t* on all computational paths' (You may find it easier to code the negation of that specification first).

(c) Express in LTL and CTL: 'Between the events *q* and *r*, *p* is never true but *t* is always true'.

(d) Express in CTL: ‘Φ is true infinitely often along every paths starting at s’. What about LTL for this statement?

7. Consider *Readers-Writers* as described on page 14 of Lecture Notes 12 and analysed in Lecture Notes 12 after page 14. Take the case of three processes and provide a model in LTL or CTL. You have to provide a state machine that defines the model as figures on pages 30 and 33 of Lecture Notes 15 for *Mutual Exclusion*, appropriate atomic predicates as n_1, n_2, t_1, t_2, c_1, c_2 for Mutual Exclusion, and appropriate safety and liveness properties.