COMP SCI 3SD3 *Virtual Take Home Examination*

Term: December 2020
Instructor: Dr. R. Janicki
Duration: 2.5h (Exam Time Only)
McMaster University Final Examination **Total: 149 pts**

- THIS EXAMINATION PAPER HAS **5** PAGES AND **9** QUESTIONS.
- The exam starts at 16:00 and ends at 19:00, i.e. 180 (3hr) minutes (for students without extra time permissions). This includes the exam time (150 minutes) plus extra time for technology (30 minutes).
- Please submit the solutions via Avenue using the same procedure as for the assignments.
- Also, just in case, send the solutions via e-mail to janicki@mcmaster.ca .
- Any question during this midterm, ask by sending an e-mail to Ryszard Janicki (janicki@mcmaster.ca)

1.[30] A cook puts burgers in a pot. A client checks if there is at least one burger in the pot, and if so, the client must take one.

(a)[20] Assuming two clients, this can be modelled as

```
range Burgers = 0..2
CLIENT = ( check -> get -> CLIENT ).

POT = POT[0],
POT[p: Burgers] = ( when p > 0 check -> POT[p]
        | get -> POT[p-1]
        | fill[n: Burgers] -> POT[n] ).

COOK = ( fill[p: 1..2] -> COOK )+{fill[0]}.
||DS = ( c1: CLIENT || c2: CLIENT || POT || COOK )
/{ {c1, c2}.check/check, {c1, c2}.get/get }.
```

(i)[12] Is there a trace belonging to process DS leading to an error state? If so, give the trace.

(ii)[8] We wish a client to obtain exclusive access to the pot. That is, two clients cannot check the pot at the same time, and when a client checks, he/she must take a burger (if there is at least one).
Show how to modify the given model such that this behaviour is ensured.

(b)[10] Provide a solution with any kind of Petri nets. Note that the cook cannot put a burger in a full pot.

2.[15] The cheese counter in a supermarket is continuously mobbed by hungry customers. There are two sorts of customer: bold customers who push their way to the front of the mob and demand services; and meek customers who wait patiently for service. Request for service is denoted by the action *getcheese* and service completion is signalled by the action *cheese*.

(a)[5] Assuming that there is always cheese available, model the system with FSP for a fixed population of two bold customers and two meek customers.

(b)[5] Assuming that there is always cheese available, model the system with Petri nets (any kind).

(c)[5] For the FSP model, show that meek customers may never be served when their requests to get cheese have lower priority than those of bold customers.

3.[20] The dining savages: A tribe of savages eats communal dinners from a large pot capable of holding *M* servings of stewed missionaries. When a savage wants to eat, he helps himself from the pot, unless it is empty, in which case he waits until the cook refills the pot. If the pot is empty, the cook refill the pot with *M* servings.
The behavior of the savages and the cook is described by:

```
SAVAGE = ( get_serving -> SAVAGE ) .
COOK = ( fill_pot -> COOK ) .
```

a.[10] Model the behavior of the pot and of the system as FSP processes.

b.[10] Model the behaviour of the pot with Petri Nets (any kind, your choice)

4.[10] In an operating system, a binary semaphore is used to control access to the console. The console is used by user and system processes. Write down a model with FSP for this system. Discuss when user processes may be denied access to the console.

5.[10] Consider the Coloured Petri Net solution to Dining Philosophers with a butler, presented as a sample solution to Question 6 of Assignment 2. Prove that this solution is deadlock-free by mimicking the proof of Proposition from page 33 of Lecture Notes 12.

6.[15] Let

```
property Alpha = (a->b->Alpha | b->Beta)+{d}
         Beta  = (c->b->Alpha | b->Beta)
```

and

```
Alpha = (a->b->Alpha | b->Beta)+{d}
Beta  = (c->b->Alpha | b->Beta)
```

Let $LTS_{propertyAlpha}$ be the Labelled Transition System of the process `property Alpha`, and LTS_{Alpha} be the Labelled Transition System of the process `Alpha`.

a.[5] Produce $LTS_{propertyAlpha}$.

b.[5] Produce LTS_{Alpha}.

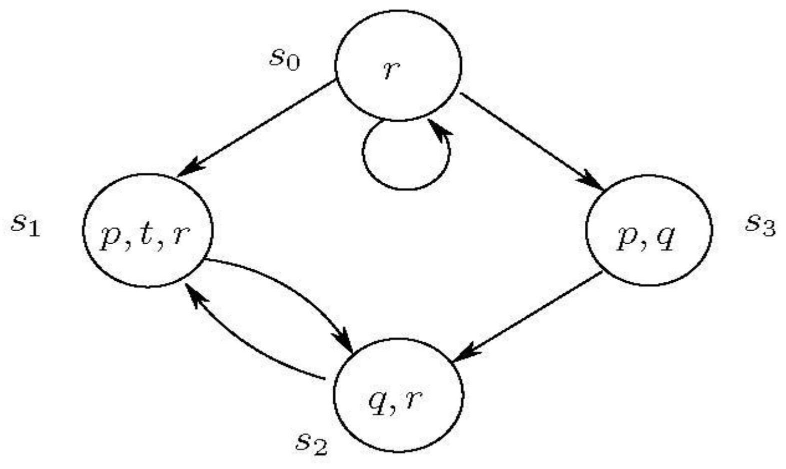
c.[5] Write a process `Alpha1` (but not property process) such that

$$LTS_{propertyAlpha} = LTS_{Alpha1}.$$

7.[15] For 'The Dining Philosophers Problem', simultaneous picking up of both forks is an abstraction of a general rule that 'the act of picking up both forks is atomic'. In other words, an order 'pick right', 'pick left' is arbitrary but once the process of picking starts, it cannot be interrupted. In practice quite often *conceptual simultaneity* is implemented as *atomicity*.
Provide a solution with FSP for Dining Philosophers with 'atomic act of (sequential) picking up both forks'.

8.[24] This question deals with Model Checking.

(a)[8] Consider the system *M* defined below:



Determine whether *M*, s_0 |= φ and *M*, s_2 |= φ hold and justify your answer, where φ is the LTL or CTL formula:

(i)[4] EG *r*

(ii)[4] G(r ∨ q)

(b)[4] Express in LTL:

If the process is ***enabled*** *infinitely often, then it* ***runs*** *infinitely often.*

(c)[4] Express in CTL:

If the process is ***enabled*** *infinitely often, then it* ***runs*** *infinitely often.*

(d)[8] Express in LTL and CTL:

A passenger entering the elevator at 5th floor and pushing 2nd floor button, will never reach 6th floor, unless 6th floor button is already lighten or somebody will push it, no matter if she/he entered an upwards or upward travelling elevator.

In this case you might use the following atomic predicates: *floor=2, direction=up, direction=down, ButtonPres2, floor=6, etc.*

9.[10] Consider the formulation of Smokers' Problem in plain English given in Lecture Notes 10, pages 5-7. The formulation of Dining Philosophers in the same style is in Lecture Notes 9 on page 7. Pages 16-18 of Lecture Notes 9, provide a solution to the Dining Philosophers problem with Elementary Petri Nets assuming that simultaneous picking forks (i.e. resources) is allowed, while pages 23-24 provide a solution to the same problem using Coloured Petri Nets, also assuming simultaneous picking forks (i.e. resources is allowed).

a.[5] Provide a solution to Smokers problem with Elementary Petri Nets and assuming that simultaneous picking (and delivery) resources is allowed (just mimic the solution for Dining Philosophers).

b.[5] Provide a solution to Smokers problem with Coloured Petri Nets and assuming that simultaneous picking (and delivery) resources is allowed (just mimic the solution for Dining Philosophers).

END OF EXAM