The University of Melbourne
School of Computing and Information Systems

SWEN90004 Modelling Complex Software Systems

Practice Exam questions—Semester 1, 2019

Note: This practice exam is intended to give you a feel for *some* of the types of question to expect in the actual exam. Exam papers vary in length. From last year, I switched from using generic "script books" for exam answers to a format in which answers are written directly onto the exam paper, therefore expect the 2019 exam paper will be longer than the practice exam, to accommodate space for answers.

Technical coverage:

- You may be asked to read and/or write pseudocode, FSP models and LTS diagrams, and Java and Go code (read only; and only to the very limited extent that we have covered Go in the subject).
- You will **not** be asked to read or write Matlab or NetLogo code.
- You may be asked questions about any of the complex systems models that we have covered in the Cx lectures. You will **not** be asked questions about any of the models from Assignment 2 (though you can of course refer to these if asked to provide examples).

The marks breakdown in this practice paper is approximately a 60/40 split between concurrency and complex systems. The mark breakdown of the 2019 exam paper will be approximately equivalent to this. Individual questions may of course have different mark values in the actual exam.

Part A – Concurrent Systems

Question 1 **[10 Marks]**

Consider the following class, designed to offer thread-safe access to a value.

```
public class ThreadSafeValue {

  private Object value;

  public ThreadSafeValue(Object value) {
    this.value = value;
  }

  public synchronized Object getValue() {
    return this.value;
  }

  public synchronized void setValue(Object v) {
    this.value = v;
  }

  public synchronized void swap(ThreadSafeValue v) {
    Object temporaryValue = this.value;
    this.value = v.getValue();
    v.setValue(temporaryValue);
  }
}
```

(a) [3 marks] Explain how the `swap` method can lead to deadlock. HINT: think about two threads executing `swap` on two shared objects.

(b) [7 marks] Provide an implementation of `swap` that avoids deadlock. HINT: You may (but do not need to) use the method `System.identityHashCode(o)`, which returns a unique integer identifier for each distinct object `o`.

Question 2 **[8 Marks]**

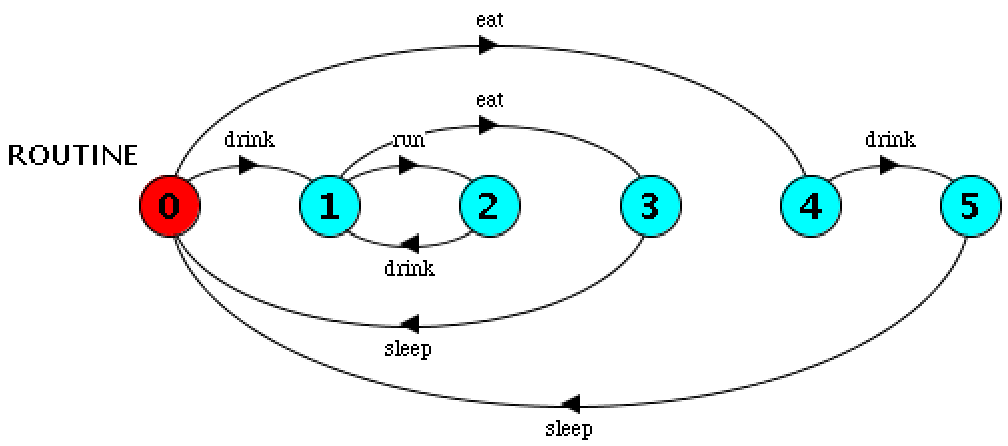
Consider the following FSP definitions.

```
P = (a -> (c -> b -> Q | b -> d -> Q)),
Q = (a -> (d -> b -> P | b -> c -> P)).

R = (a -> b -> c -> R).

||S = (P || R).
```

(a) [3 marks] Draw the labeled transition system (LTS) graph for `P`.

(b) [2 marks] Write an alternative definition for the process `S` without the use of the composition (`||`) operator.

(c) [3 marks] Give an FSP definition of the process `ROUTINE` that has the following LTS:



Question 3 **[12 Marks]**

A simple resource allocator controls access to a fixed number of resources (the parameter to the constructor defines the number of resources). Threads can request n resources from the controller by engaging in the action `get[n]`, which they must release at some later point in time by a call to `put[n]`. The call to `get[n]` should suspend until there are enough resources available. You may assume that the number of resources requested in a call to get is always greater than 0, but never greater than the total number of resources, `N`.

Assume the following declarations:

```
const N = 10
range T = 0..N
const U = 5
```

(a) [6 marks] Define an FSP model of the resource allocator, called *RESOURCE_ALLOC*.

(b) [3 marks] The following process model, `RESOURCE_USER`, continually gets, uses, and replaces resources:

```
RESOURCE_USER =
   (get[n:T] -> use[n] -> put[n] -> RESOURCE_USER).
```

Define a composite FSP model called `||SYSTEM` that composes together `U` `RESOURCE_USER` processes with one resource allocator.

(c) [3 marks] Define a linear temporal logic (LTL) property that specifies that every process will eventually be allocated the resources that it requests.

Question 4 [10 Marks]

A car park has the capacity to hold N cars. The car parks opens and closes at unpredictable times that are determined by its operator. When the car park is open, a car can enter, unless the car park is full. When the car park is closed, no car can enter, but cars can still leave.

The car park's interaction with its operator and a single car is modelled is FSP as follows:

```
CARPARK
  = CLOSED[0],
CLOSED[i:0..N]
  = ( when (i>0) depart -> CLOSED[i-1]
    | open -> OPEN[i]
    ),
OPEN[i:0..N]
  = ( when (i<N) arrive -> OPEN[i+1]
    | when (i>0) depart -> OPEN[i-1]
    | close -> CLOSED[i]
    ).
```

Implement this model of a car park as a Java class called `Carpark`, having methods `arrive()`, `depart()`, `open()` and `close()`.

Question 5 [10 Marks]

The following Go program (continued on the following page) implements a system that produces and consumes a finite sequence of integers in a concurrent fashion, printing output to the terminal.

```
package main

const N = 20

func producer(out chan<- int) {
    for i:=0; i<N; i++ {
        out <- i
    }
}

func adder(in <-chan int, out chan<- int, val int) {
    for {
        x := <-in
        out <- x + val
    }
}

func multiplier(in <-chan int, out chan<- int, val int) {
    for {
        x := <-in
        out <- x * val
    }
}

func consumer(in <-chan int, done chan<- bool, val int) {
    for i:=0; i<N; i++ {
        x := <-in
        if i%val == 0 {
            print(x, "\n")
        }
    }
    done <- true
}
```

(continued on following page...)

```
func main() {
    add_val := 3
    mult_val := 2
    filter_val := 3

    ch1 := make(chan int)
    ch2 := make(chan int)
    ch3 := make(chan int)
    done := make(chan bool)

    go producer(ch1)
    go adder(ch1, ch2, add_val)
    go multiplier(ch2, ch3, mult_val)
    go consumer(ch3, done, filter_val)

    <-done
}
```

(a) [6 marks] When the program terminates, what output will have been printed?

(b) [2 marks] What output would be printed if the final line `<-done` were omitted from the `main` function. Briefly explain your answer.

(c) [2 marks] Briefly explain the difference between a buffered and an unbuffered channel in Go.

Part B – Complex Systems

Question 6 **[7 Marks]**

(a) [3 marks] Explain what makes a system a *complex system*. Use examples to illustrate your answer.

(b) [4 marks] What are the four properties displayed by chaotic systems (in the dynamic systems sense). Briefly describe how these are exemplified by the logistic map.

Question 7 **[5 Marks]**

(a) [3 marks] Define and/or give examples for the following elements of a cellular automaton:

- interactions
- environment
- initial conditions

You should include a labelled diagram of a cellular automata in your answer.

(b) [2 marks] Define emergence. Provide an example of emergence that occurs in one of the CA models we studied during the course.

Question 8 **[10 Marks]**

(a) [3 marks] In the context of an agent-based model, describe the difference between the following terms when used to classify an agent's environment.

(a) deterministic *vs* non-deterministic
(b) discrete *vs* continuous

(b) [4 marks] Select one of the systems/models we studied in lectures/workshops/assignments. How do the agents interact with each other? Consider the key questions about space, information and interaction.

(c) [3 marks] Thinking about an *agent-based* version of the predator-prey model (eg, sheep and wolves, or foxes and rabbits), in which the agents are individual predator and prey animals located in a two-dimensional spatial field, describe one role that an agent's *neighbourhood* could play. How might the behaviour of the system change if the size of an agent's neighbourhood was increased? What might happen if predator and prey species had neighbourhoods of different sizes?

Question 9 **[8 Marks]**

(a) [3 marks] This question refers to small-world networks.

(i) Describe one technique that can be used to generate small-world networks.

(ii) Compare the relative values of key network metrics for small-world networks as compared to regular lattices.

(b) [2 marks] What is meant by the term *preferential attachment* when describing the formation of a scale-free network? What impact does this mechanism have on the degree distribution of the network?

(c) [3 marks] If we wanted to explore the SIR model over a network, how could the underlying network structure of the population be captured in the model? What real world data would you need to collect?

Part C – Modelling Complex Systems Problem

In this section, you will be presented with a small case study (about 1/2 page), followed by a series of questions.

As discussed in lectures, foxes are an introduced species in Australia, and are known to predate upon native animal and bird species. To protect the native wildlife in a particular park, the National Parks and Wildlife Service (NPWS) are undertaking an eradication program to trap and remove foxes. They have limited funding with which to undertake this project.

While the park is large, foxes and other species are distributed in a non-uniform fashion (ie, there are more in some areas than others). The NPWS hires a number of trappers to deploy traps, offering payment for each fox caught. The hired trappers each have a limited number of traps to deploy and collect each day. As the park is covered in dense bush, they have to travel by foot to deploy and collect their traps. Trappers will only continue to work if they earn sufficient money to make it worth their time.

The NPWS have asked you to build a model to help them decide how many trappers they should hire, what level of payment they should offer for each fox caught, and how long they need to continue the eradication program to ensure that fox numbers are reduced sufficiently to enable the survival of the native species.

If you find some of the requirements listed above to be ambiguous, do not panic. Simply record your (reasonable) assumptions and proceed with your solution to the question.

Question 10 **[10 Marks]**

Describe how you might design and implement an agent-based simulation model for this case study listed. In your answer, you should:

(a) [2 marks] Describe why an agent-based model is an appropriate modelling paradigm to use in this domain.

(b) [3 marks] Describe the agents, interactions and update rules in your model.

(c) [2 marks] Describe a question or hypothesis that your model could be used to address, and the experiments you would use to do so.

(d) [3 marks] Explain the assumptions and consequent limitations of your model.

Note: in the exam, you may be asked about different aspects of a scenario to the question above, or given a more specific or more general brief.

Question 11 **[10 Marks]**

To ensure that only a fixed number of trappers are operating in the national park at a given time, the NPWS have implemented a strict entry control agent. Upon entering the park, each trapper is issued with a token, which they return when they exit. If there are no tokens remaining, no further trappers can enter the park.

You may assume the following:

1. There is a single point of entry/exit into the park.
2. There are `N` trappers in the system.
3. There are `T` tokens available.

Define `TRAPPER` and `AGENT` processes, together with a composed process `||SYSTEM` that models the park access system described above.