College of Science – Computer Science

CSC385 – Modelling and Verification Techniques

Release Time: **9:20 am Friday 5/06/2020 (Time Zone: BST)**

Deadline: **17:20am Friday 5/06/2020 (Time Zone: BST)**

Alternative Assessment Information

- This is an open-book assessment. This means you may consult your notes, textbooks, and other resources, including calculators, as you see fit.
- You must submit before the deadline. Allow some spare time for technical submission issues.
- This assessment is designed to take 2 hours to complete (maybe a little longer to account for typing speed). The deadline has been set to give you a longer window than necessary to allow you time to deal with technical issues, provide some flexibility of starting times, and to help students with disability access plans that require rest breaks and extra time.
- It is suggested that you use Microsoft Word (or any other editor of your choice) to type your answers, then save as PDF when you are ready to submit. All submitted text must be word-processed, but you may include images (or photos of hand drawn images) as part of the document.
- This is an individual assessment. Under no circumstances are you to discuss any aspect of this assessment with anyone; nor are you allowed to share this document, ideas or solutions with others using email, social media, instant messaging, websites, or any other means. Your attempts at these questions must be entirely your own work. Those found to have collaborated with others will receive a mark of 0.

Special Instructions

None

Submission Instructions

- Please submit a **single PDF** file **named as your student number** (e.g. 123456.pdf) via the submission link located on the module page in Blackboard/Canvas.

By submitting, electronically and/or hardcopy, you state that you fully understand and are complying with the university's policy on Academic Integrity and Academic Misconduct. The policy can be found at https://myuni.swansea.ac.uk/academic-life/academic-misconduct.

Originator(s): ***Ulrich Berger***

In case of queries email both: and

Question 1

We discussed regular expressions as an example of a formal method. In particular, we discussed three ways of assigning a semantics to regular expressions:

(i) An assignment of a language, that is, a set of words, to each regular expression.

(ii) An assignment of a finite automaton to each regular expression.

(iii) An system of axioms and rules to derive valid equations between regular expressions.

Give the names of each of these semantics.

[3 marks (1 for each correct answer)]

Question 2

(a) Suppose you are given a tool that can test determinism, deadlock freeness and trace refinement of processes. Furthermore, suppose you are given two processes P and Q that are known to be deterministic.

Describe how the tool can be used to test whether P and Q are bisimilar. Say which theoretical result makes this possible.

[6 marks (4 for description, 2 for theoretical result)]

(b) Give an example of two processes that are trace equivalent, but not bisimilar (you may draw diagrams). Justify your answer, that is, explain why the two process are trace equivalent, but not bisimilar.

[4 marks (2 for example, 2 for justification)]

Question 3

Consider the following CSP process definitions:

$$\begin{aligned} P &= a \to c \to P \;\Box\; b \to d \to P \\ Q &= c \to d \to Q \\ S &= P \,[\![\{c, d\}]\!]\, Q \end{aligned}$$

Decide for each of these processes whether they are deterministic or deadlock free.

You may submit drawings of the labelled transition systems for the processes P, Q and S but this is not required.

[8 marks]

Question 4

Consider the following CSP process definition of a robot which repeatedly can move to the left or to the right in a cyclic fashion, or report its position, or pick up or drop some item.

```
min = 0
max = 5
Range = {min..max-1}
left(x) = (x-1)%max
right(x) = (x+1)%max

datatype Direction = L | R

channel move : Direction
channel position : Range
channel pick, drop

Robot(x) = move.L      -> Robot(left(x))   []
           move.R      -> Robot(right(x))  []
           position.x  -> Robot(x)         []
           pick        -> Robot(x)         []
           drop        -> Robot(x)
```

The behaviour of the robot should be constrained in such a way, that it can hold at most one item, and it can drop an item only if it holds one. Furthermore, any picked-up item must be dropped immediately to the left or to the right where it was picked-up.

Define a process `Constraint` and a synchronisation set `X` such that the process defined by

```
Robot(min) [| X |] Constraint
```

behaves as specified above.

[8 marks]

Question 5

A simplified version of the Needham-Schroeder authentication protocol is usually described in brief by the following rules:

(1) $A \longrightarrow B : \{\mathrm{N}_A, A\}_{\mathrm{pk}_B}$

(2) $B \longrightarrow A : \{\mathrm{N}_A, \mathrm{N}_B\}_{\mathrm{pk}_A}$

(3) $A \longrightarrow B : \{\mathrm{N}_B\}_{\mathrm{pk}_B}$

(a) Referring to (1), (2), (3) above, explain the Needham-Schroeder protocol. What is the purpose of the protocol? Which cryptographic mechanism is it based on?

[2 marks]

(b) Describe the attack on the Needham-Schroeder protocol found by Lowe, using a similar style of rules as above.

[2 marks]

(c) Sketch how the Needham-Schroeder protocol can be amended to prevent this attack. Briefly explain why the attack is no longer possible.

[2 marks]

(d) We modelled the amended protocol by a process `System(t)`, specified its correctness by a process `SPEC`, and checked its correctness by a suitable assertion:

```
SYSTEM(t) = ENV(t) [| {| send, receive |} |] USERS

SPEC = know ? n : noncesI -> SPEC

assert SPEC [T= System(20) \ {| send, receive |}
```

In the following, 'explain' means giving an informal explanation showing an understanding of the modelling.

(i) Explain what the process `System(t)` means (describe its components and explain the way they are connected).

(ii) Explain what the assertion means. Explain all parts of it, in particular, explain the meaning of `\ {| send, receive |}` and why this part is necessary.

(iii) Explain which correctness guarantees the successful model checking of this assertion provides for the protocol.

[6 marks (2 marks each)]

Question 6

(a) Describe the main elements of program extraction from constructive proofs. Your description should include

- an explanation of the difference between constructive and classical logic;
- at least two examples illustrating the correspondence between formulas and proof rules on the one hand and types and program constructs on the other hand.

[3 marks]

(b) Suppose one has proven constructively the following formula (where the variables n, m range over natural numbers):

$$\forall n\, \exists m\, (m^2 \leq n < (m+1)^2)$$

(i) Give an informal description what the extracted program computes?

(ii) What result would the extracted program return when run with the number 10 as input?

[6 marks (3 marks each)].

End of Paper