Swansea University
Department of Computer Science

U Berger
October 20, 2021

CSC385/CSCM85 Modelling and Verification Techniques

Coursework Assignment

Due date: Friday, 4th of November 2022

Submission via Canvas in groups of up to 4 students

(see the instructions on the last page)

Question 1

Consider two vending machines, VM_1 and VM_2 which accept a coin as payment and deliver tea. However, VM_1 requires payment upfront whereas VM_2 has the option of getting tea first and paying later. Here are the LTSs for these two machines

$$\{(\mathrm{VM}_1, \mathrm{coin}, C_1), (C_1, \mathrm{tea}, C_0), (C_1, \mathrm{coin}, C_2), (C_0, \mathrm{coin}, C_1), (C_2, \mathrm{tea}, C_1)\}$$

$$\{(\mathrm{VM}_2, \mathrm{coin}, P), (\mathrm{VM}_2, \mathrm{tea}, M), (P, \mathrm{tea}, \mathrm{VM}_2), (M, \mathrm{coin}, \mathrm{VM}_2)\}$$

(a) Draw both LTSs (you may use FDR but drawings by hand are fine as well).

(b) Give CSP definitions of both LTSs.

(c) Show that VM_1 and VM_2 are not trace equivalent.

(d) Find a state s in the LTS for VM_2 that is bisimilar to VM_1. Show bisimilarity by giving a bisimulation that contains the pair (VM_1, s).

[30 marks]

Question 2 Decide for each of the following statements whether it is true for all processes A, B:

(a) If A and B are trace equivalent and A is deadlock free, then B is deadlock free.

(b) If A and B are bisimilar and A is deadlock free, then B is deadlock free.

In each case, either prove the statement, or give a counterexample.

You may use the following definition of a deadlock:

> A process has S has a deadlock if there are a word w and a process S' such that $S \xrightarrow{w}{}^{*} S'$ and there is no transition from S'.

Hence, for example, part (b) can be equivalently reformulated as

(b) If A and B are bisimilar and B has a deadlock, then A has a deadlock.

[40 marks]

Question 3

Consider the following definition of a robot that repeatedly reports its positions, then moves to the left or right (within a given finite range), or does some work.

```
min = 0
max = 5

Range = {min..max}

datatype Direction = L | R

channel move : Direction
channel position : Range
channel work

Robot(x) = position.x ->
           ( (if x > min then move.L -> Robot(x-1) else STOP) []
             (if x < max then move.R -> Robot(x+1) else STOP) []
             (work -> Robot(x)) )
```

Suppose doing work empties the robot's battery so that it needs at least two movements to recharge the battery (for example using a solar panel). We assume that, initially, the robot's battery is empty.

Define a process `Empty` and a suitable synchronisation set `X` such that the process

```
Robot(0) [| X |] Empty
```

models this behaviour.

It will be convenient to define `Empty` simultaneously with a process `Full` that represents the fully charged battery.

[30 marks]

Submission instructions

When submitting as a group, **all group members** must submit an **identical** copy.

Each submission must show on the first page **course** (CSCM85 or CSC385), **student number**, and **name** of **all authors**.

Submitted files must be in pdf format and may be typed or scanned from a handwritten manuscript. Please make sure that your handwriting is legible.