Swansea University College of Science

Prifysgol Abertawe Coleg Gwyddoniaeth

May/June 2018/19

CSC385
Modelling and Verification Techniques

Time Available: 2 hours

Coordinator: Dr U. Berger

Queries: The Exams Office hold contact details for this paper

Only University-supplied dictionaries are permitted

Calculators? Not Permitted

(Attempt all questions)

Question 1

We identified *syntax*, *semantics*, and *method* as the main elements of formal methods.

(a) Briefly explain the meaning of these notions.

[3 marks (1 for each notion)]

(b) Name and describe three different styles of semantics. You may underpin your descriptions by suitable examples.

[3 marks (1 for each style)]

Question 2

A *labelled transition system (LTS)* over an alphabet A is a pair (S, α) where S is a set and $\alpha \subseteq S \times A \times S$. The elements of S are called *states* and α is called *transition relation*. Instead of $(s, a, s') \in \alpha$ one often writes $s \xrightarrow{a}_{\alpha} s'$, or just $s \xrightarrow{a} s'$ if the transition relation is clear from the context.

A *process* is a labelled transition system (S, α) together with a selected state $s \in S$, called *start state*. If the LTS is clear from the context, we may speak of a state s alone as a process.

(a) Let (S, α) and (T, β) be labelled transition systems, both over the same alphabet A.

(i) Define what it means for two processes $s \in S$ and $t \in T$ to be trace equivalent (the answer should include a definition of the set of traces of a process),

(ii) What is a bisimulation between (S, α) and (T, β)? When are two processes $s \in S$ and $t \in T$ called bisimilar?

[4 marks (2+2)]

(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]

(c) Explain (informally) the game characterisation of bisimilarity.

[3 marks]

Question 3

Consider the following CSP process definitions:

$$\begin{aligned} P &= (a \rightarrow \text{SKIP} \;|||\; b \rightarrow \text{SKIP}) \;;\; P \\ Q &= a \rightarrow Q \;\Box\; b \rightarrow Q \end{aligned}$$

(a) Explain the meanings of the operators ' ||| ', ' ; ', ' □ ', and the constant SKIP.
Hint: You may write down firing rules or give informal explanations.

[4 marks (1 each)]

(c) Explain the notion of trace refinement.

[2 marks]

(d) Does P trace refine Q? Does Q trace refine P? In each case justify your answer.

[2 marks]

Question 4

Consider the following CSP process definition of a robot which repeatedly can move to the East, West, North, or East (within a given finite range), report its position, or do some work.

```
min = 0
max = 5
Range = {min..max}
inc(x) = if x < max then x+1 else x
dec(x) = if x > min then x-1 else x

datatype Direction = E | W | N | S

channel move : Direction
channel position : Range.Range
channel work

Robot(x,y) = (move.E -> Robot(inc(x),y))  []
             (move.W -> Robot(dec(x),y))  []
             (move.N -> Robot(x,inc(y)))  []
             (move.S -> Robot(x,dec(y)))  []
             (position.x.y -> Robot(x,y)) []
             work -> Robot(x,y)
```

Suppose doing work empties the robot's battery so that it needs at least two non-work events to happen to recharge the battery.

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

```
Robot(min,min) [| X |] Full
```

models this behaviour.

[7 marks]

Question 5

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

$$
\begin{aligned}
(1) \quad & A \to B : \{N_A, A\}_{pk_B} \\
(2) \quad & B \to A : \{N_A, N_B\}_{pk_A} \\
(3) \quad & A \to B : \{N_B\}_{pk_B}
\end{aligned}
$$

(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?

[3 marks]

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

[3 marks]

(c) Sketch how the Needham-Schroeder protocol can be amended to make this attack impossible.

[1 mark]

(d) Let `SYSTEM(t)` be the process describing the amended protocol for `t` rounds, that is, the users (including the intruder) interacting with the environment, and let `SPEC` be the process that describes all sequences of events of the form `know.n` where `n` is a nonce the intruder is allowed to know, that is,

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

SPEC = know ? n : noncesI -> SPEC
```

Write down an assertion that can be used to verify that the amended protocol is safe for 3 rounds.

Give an informal explanation of what the assertion means.

[3 marks]

Question 6

(a) Describe the main elements of program extraction from constructive proofs.

[4 marks]

(b) Give an example of program extraction from proofs: write down the formula to be proven, give the type of the extracted program and explain (informally) what the extracted program computes.

[4 marks].

End of Paper