The University of Auckland

Department of Electrical and Computer Engineering

COMPSYS 705 Formal Methods for Safety Critical Software Test, 22 October 2021

Last name	Name	ID

Question	Mark
Q1	
Q2	
Q3	
Q4	
Q5	
Q6	
Total	

NOTE

1. **Answer ALL questions. Part I (Partha's part) covers 70 marks, while Part II (Avinash's Part) covers 30 marks in this test.**
2. **Questions 1-4 are for Part I and the rest are Part II.**
3. **The maximum score on this test is 100 marks.**
4. **Weighting is 50%.**
5. **Show all code for Questions 5 and 6.**
6. **The maximum time allowed is 1 hour and 30 minutes.**
7. **Write each answer on a separate sheet of paper.**
8. **Answers should be legible.**
9. **Scan and upload all answer sheets as a single pdf.**

1. Answer the following short-answer type questions realed to real-time systems:

   (a) A real-time system is not necessarily fast. Justify this statement using a suitable example. (4 Marks)

   (b) Measurement-based timing analysis of real-time systems may lead to catastrophic consequences for a hard real-time system. Justify this statement. (4 Marks)

   (c) Jitter in real-time systems is not good. Why? (4 Marks)

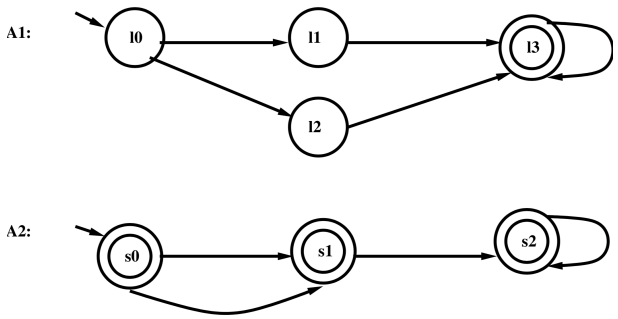

Figure 1: Two automata

   (d) Consider the two automata shown in Figure 1. You are required to compute the intersection of these two automata using the following definition shown in Figure 2 taught in the course. (8 Marks)

Intersection of automata.

Let $\mathcal{A} = (Q, q_0, \Sigma, \delta, F)$ and $\mathcal{A}' = (Q', q_0', \Sigma, \delta', F')$ be two automata over the same alphabet Σ. The *intersection* of $\mathcal{A}$ and $\mathcal{A}'$, denoted $\mathcal{A} \cap \mathcal{A}'$, is defined as $(Q \times Q', (q_0, q_0'), \Sigma, \delta \times \delta', F \times F')$, where $(\delta \times \delta')((q, q'), a) = (\delta(q, a), \delta'(q', a))$. We have $\mathcal{L}(\mathcal{A} \cap \mathcal{A}') = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{A}')$.

Figure 2: Definition of intersection of automata

2. (a) When two processes are simulation equivalent, they are also bisimilar. Is this statement true / false? Using a suitable example justify your answer. (5 Marks)

   (b) When two processes are bisimilar, they are also simulation equivalent. Is this statement true / false? Using a suitable example justify your answer. (5 Marks)

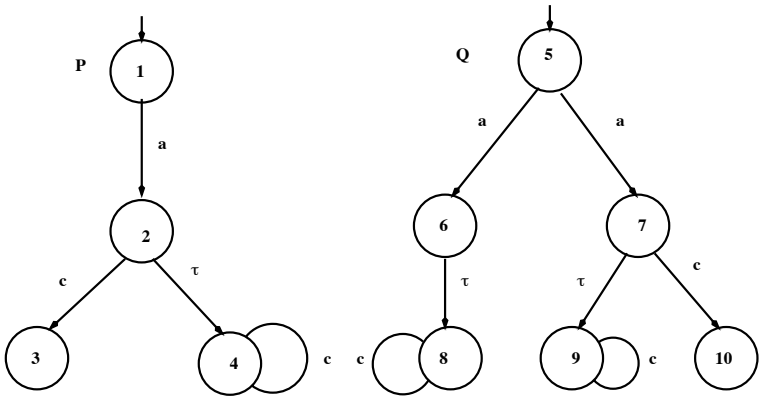

Figure 3: Two CCS Processes P, Q

(c) Consider the following two processes P, Q as shown in Figure 3. Identify a suitable equivalence / preorder between the processes by computing a relation R. Provide the mathematical definition of the equivalence / preorder used. (5 Marks)

(d) Give an example Boolean Automaton (used in Argos) that is non-reactive but deterministic. Using this example, explain the difference between *determinism* and *reactivity* relative to synchronous processes. (5 Marks)

3. (a) For the following temporal logic formulae, identify the ones which are valid CTL formulae with justification for your answer for each.

   i. $(p1\neg p2\neg r \Rightarrow q) \Rightarrow q)$
   ii. $AG(trueUreq \Rightarrow AFack)$.
   iii. $AG(p \vee q \wedge \neg p \Rightarrow EXq)$.
   iv. $\neg E[\neg gU(\neg f \wedge \neg g)] \wedge \neg EG\neg g]$
   v. $AG(AF[ns = g]) \wedge AG(AF[ew = g])$.

   (5 Marks)

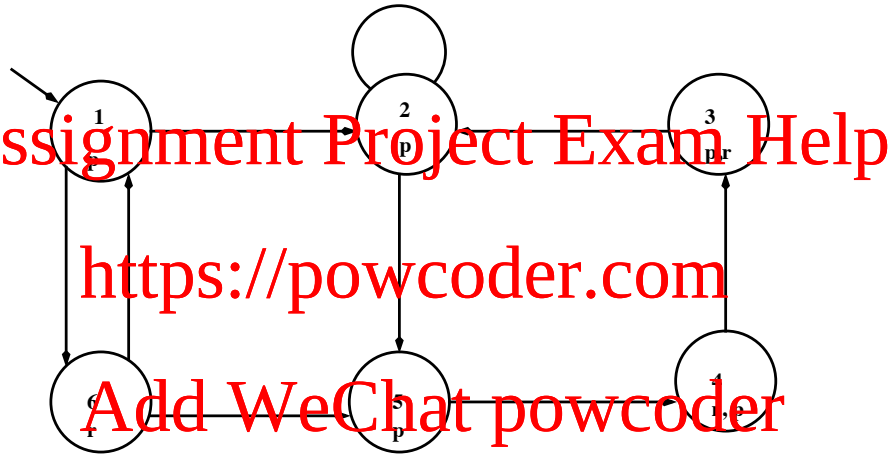

Figure 4: A simplified model

(b) Consider the Kripke structure shown in Figure 4. Answer the following based on this. You have to show all steps. (15 Marks)

- Apply the explicit-state model checking algorithm to verify the property $E(pUr)$.
- Apply the same algorithm to verify the property EGp.

4. Using the DDD-mode pacemaker, explain the following terms relative to cyber-physical systems. (10 Marks)

   (a) Explain the key differences between delay and deadline with suitable example of pacemaker timers.

   (b) Explain the key distinction between a timed automata (TA) and a hybrid automata (HA), illustrated using suitable examples and application domains.

   (c) Will it be better to model the pacemaker as an HA rather than a TA? Justify by stating the advantages and limitations of the HA-based pacemaker.

   (d) Use a suitable timing diagram, which starts with a ventricular pulse, provide a trace to show how certain timers take priority.

5. Prove that the `swap1` and `swap2` functions below return the same swapped values for *all* the same inputs `a` and `b` of type `int`. Encode the equivalence (validation) problem as SMT constraints in Z3 Python API. Show the Python code along with the output of running the Z3 solver. (15 marks)

```
def swap1(a, b):
 tmp = b
 b = a
 a = tmp
 return a, b
```

```
def swap2(a, b):
 a = a + b
 b = a - b
 a = a - b
 return a, b
```

6. Let x, y, z of some generic type T represent the sides an *equilateral* triangle. Let an uninterpreted function L with type signature $L : T \rightarrow Int$ give the length of any side of the equilateral triangle. The goal is to prove the *conjecture*: "the sum of the lengths of any two sides of the equilateral triangle is greater than the (other) third side". Encode the above goal as SMT constraints in Z3 via Python API. Show the Python code along with the output of running the Z3 solver. Is the aforementioned conjecture true? (15 marks)

Scratch Pad