SOLUTION NOTES

Specification and Verification II 2002 Paper 7 Question 2 (MJCG)

(a) Devise a state space [4 marks] and transition relation to represent the behavior of the array of switches [6 marks].

The state space can consist of the set of vectors

`(v0,v1,v2,v3,v4,v5,v6,v7,v8)`

where the boolean variable `v`i represents switch number i+1, and is true if and only if switch i+1 is `T`.

A transition relation `Trans` is then defined by

```
Trans((v0,v1,v2,v3,v4,v5,v6,v7,v8),
      (v0',v1',v2',v3',v4',v5',v6',v7',v8')) =
 ((v0'=¬v0)∧(v1'=¬v1)∧(v2'=v2)∧(v3'=¬v3)∧(v4'=v4)∧
  (v5'=v5)∧(v6'=v6)∧(v7'=v7)∧(v8'=v8))     (toggle switch 1)
 ∨
 ((v0'=¬v0)∧(v1'=¬v1)∧(v2'=¬v2)∧(v3'=v3)∧(v4'=¬v4)∧
  (v5'=v5)∧(v6'=v6)∧(v7'=v7)∧(v8'=v8))     (toggle switch 2)
 ∨
 ((v0'=v0)∧(v1'=¬v1)∧(v2'=¬v2)∧(v3'=v3)∧(v4'=v4)∧
  (v5'=¬v5)∧(v6'=v6)∧(v7'=v7)∧(v8'=v8))    (toggle switch 3)
 ∨
 ((v0'=¬v0)∧(v1'=v1)∧(v2'=v2)∧(v3'=¬v3)∧(v4'=¬v4)∧
  (v5'=v5)∧(v6'=¬v6)∧(v7'=v7)∧(v8'=v8))    (toggle switch 4)
 ∨
 ((v0'=v0)∧(v1'=¬v1)∧(v2'=v2)∧(v3'=¬v3)∧(v4'=¬v4)∧
  (v5'=¬v5)∧(v6'=v6)∧(v7'=¬v7)∧(v8'=v8))   (toggle switch 5)
 ∨
 ((v0'=v0)∧(v1'=v1)∧(v2'=¬v2)∧(v3'=v3)∧(v4'=¬v4)∧
  (v5'=¬v5)∧(v6'=v6)∧(v7'=v7)∧(v8'=¬v8))   (toggle switch 6)
 ∨
 ((v0'=v0)∧(v1'=v1)∧(v2'=v2)∧(v3'=¬v3)∧(v4'=v4)∧
  (v5'=v5)∧(v6'=¬v6)∧(v7'=¬v7)∧(v8'=v8))   (toggle switch 7)
 ∨
 ((v0'=v0)∧(v1'=v1)∧(v2'=v2)∧(v3'=v3)∧(v4'=¬v4)∧
  (v5'=v5)∧(v6'=¬v6)∧(v7'=¬v7)∧(v8'=¬v8))  (toggle switch 8)
 ∨
 ((v0'=v0)∧(v1'=v1)∧(v2'=v2)∧(v3'=v3)∧(v4'=v4)∧
  (v5'=¬v5)∧(v6'=v6)∧(v7'=¬v7)∧(v8'=¬v8))  (toggle switch 9)
```

This transition relation is very straightforward to write down for someone who knows what they are doing. However, if a candidate shows the he/she can construct the relation, but doesn't give all the details then I will give good marks.

(*b*) You are given the problem of getting from an initial state in which even numbered switches are on and odd numbered switches are off, to a final state in which all the switches are off.

Write down predicates on your state space that characterises the initial [2 marks] and final [2 marks] states.

Predicates `Init`, `Final` characterising the initial and final states, respectively are defined by:

```
Init(v0,v1,v2,v3,v4,v5,v6,v7,v8) =
 ¬v0 ∧ v1 ∧ ¬v2 ∧ v3 ∧ ¬v4 ∧ v5 ∧ ¬v6 ∧ v7 ∧ ¬v8

Final(v0,v1,v2,v3,v4,v5,v6,v7,v8) =
 ¬v0 ∧ ¬v1 ∧ ¬v2 ∧ ¬v3 ∧ ¬v4 ∧ ¬v5 ∧ ¬v6 ∧ ¬v7 ∧ ¬v8
```

I'll give one mark for a good try and two marks for a complete and correct definition.

(*c*) Explain how you might use a model checker to find a solution to the problem. [6 marks]

Model checkers can usually find counter-examples to properties, and sequences of transitions from an initial state to a counter-example state. Thus we could use a model checker to find a trace to a counter-example to the property that `¬Final(v0,v1,v2,v3,v4,v5,v6,v7,v8)`.

The answer above is a bit minimal. I would give 4 marks for something like this. To get all 6 marks I'd want a bit more discussion about how counterexamples are found.