PROJECT 3:

Project with NuSMV

Nikolay Nikolov

Problem 2

Tasks:

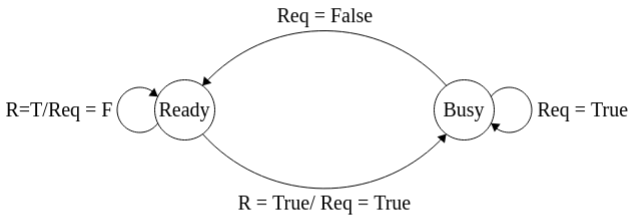
1) Read the tutorial of Nusmv 2.5 and learn the syntax, draw the state diagram of above code.

Verify formula

```
NuSMV > read_model -i short.smv
NuSMV > flatten_hierarchy
NuSMV > encode_variables
NuSMV > build_model
NuSMV > check_ctlspec
-- specification AG (request -> AF state = busy)  is true
NuSMV > check_ctlspec -p "AG(request ->AX(state=busy))
ignoring unbalanced quote ...
-- specification AG (request -> AX state = busy)  is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
  -- Loop starts here
  -> State: 1.1 <-
     request = FALSE
     state = ready
  -> State: 1.2 <-
     request = TRUE
     state = busy
  -> State: 1.3 <-
     request = FALSE
     state = ready
NuSMV >
*******  Simulation Starting From State 1.4    ********
NuSMV > show_traces -v
    <!-- ########################## Trace number: 1 ########################## -->
Trace Description: Simulation Trace
Trace Type: Simulation
  -> State: 1.1 <-
     request = FALSE
     state = ready
  -> State: 1.2 <-
     request = TRUE
     state = ready
  -> State: 1.3 <-
     request = TRUE
     state = busy
```

```
-> State: 1.4 <-
  request = FALSE
  state = ready
-> State: 1.5 <-
  request = TRUE
  state = ready
-> State: 1.6 <-
  request = FALSE
  state = busy
-> State: 1.7 <-
  request = FALSE
  state = ready
-> State: 1.8 <-
  request = TRUE
  state = ready
-> State: 1.9 <-
  request = TRUE
  state = busy
-> State: 1.10 <-
  request = FALSE
  state = busy
-> State: 1.11 <-
  request = TRUE
  state = busy
-> State: 1.12 <-
  request = FALSE
  state = ready
```

State Diagram



Come up with 3 more CTL properties and check against the model.

1)

```
NuSMV > check_ctlspec -p "AG(request -> EF(state=busy))"
-- specification AG (request -> EF state = busy)  is true
```

2)

```
NuSMV > check_ctlspec -p "EF(request -> AG(state=busy))"
-- specification EF (request -> AG state = busy)  is true
```

3)

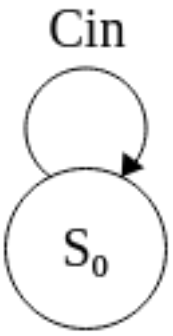
```
NuSMV > check_ctlspec -p "EF(request -> AG(state=busy))"
-- specification EF (request -> AG state = busy)  is true
```

4)

```
NuSMV > check_ctlspec -p "AG(request -> EF(state=ready))"
-- specification AG (request -> EF state = ready)  is true
```

Problem 3:

State Diagram



Trace

```
NuSMV > read_model -i counter.smv
NuSMV > flatten_hierarchy
NuSMV > encode_variables
NuSMV > build_model
NuSMV > pick_state -i

***************  AVAILABLE STATES  *************

  ================ State ================
  0) ------------------------------
  bit0.carry_out = FALSE
  bit0.value = FALSE
  bit1.carry_out = FALSE
  bit1.value = FALSE
  bit2.carry_out = FALSE
  bit2.value = FALSE

********  Simulation Starting From State 1.4  ********
NuSMV > show_traces -v
    <!-- ######################## Trace number: 1 ######################## -->
Trace Description: Simulation Trace
Trace Type: Simulation
  -> State: 1.1 <-
    bit0.value = FALSE
    bit1.value = FALSE
    bit2.value = FALSE
```

```
    bit0.carry_out = FALSE
    bit1.carry_out = FALSE
    bit2.carry_out = FALSE
-> State: 1.2 <-
    bit0.value = TRUE
    bit1.value = FALSE
    bit2.value = FALSE
    bit0.carry_out = TRUE
    bit1.carry_out = FALSE
    bit2.carry_out = FALSE
-> State: 1.3 <-
    bit0.value = FALSE
    bit1.value = TRUE
    bit2.value = FALSE
    bit0.carry_out = FALSE
    bit1.carry_out = FALSE
    bit2.carry_out = FALSE
-> State: 1.4 <-
    bit0.value = TRUE
    bit1.value = TRUE
    bit2.value = FALSE
    bit0.carry_out = TRUE
    bit1.carry_out = TRUE
    bit2.carry_out = FALSE
-> State: 1.5 <-
    bit0.value = FALSE
    bit1.value = FALSE
    bit2.value = TRUE
    bit0.carry_out = FALSE
    bit1.carry_out = FALSE
    bit2.carry_out = FALSE
-> State: 1.6 <-
    bit0.value = TRUE
    bit1.value = FALSE
    bit2.value = TRUE
    bit0.carry_out = TRUE
    bit1.carry_out = FALSE
    bit2.carry_out = FALSE
-> State: 1.7 <-
    bit0.value = FALSE
    bit1.value = TRUE
    bit2.value = TRUE
    bit0.carry_out = FALSE
    bit1.carry_out = FALSE
    bit2.carry_out = FALSE
-> State: 1.8 <-
```

```
bit0.value = TRUE
bit1.value = TRUE
bit2.value = TRUE
bit0.carry_out = TRUE
bit1.carry_out = TRUE
bit2.carry_out = TRUE
```

2) Come up with 3 more CTL property and check against above model.

1)

```
NuSMV > check_ctlspec -p "EF(bit1.value=TRUE)"
-- specification EF bit1.value = TRUE  is true
```

2)

```
NuSMV > check_ctlspec -p "EF(bit2.value=TRUE)"
-- specification EF bit2.value = TRUE  is true
```

3)

```
NuSMV > check_ctlspec -p "EF(bit0.value=TRUE)"
-- specification EF bit0.value = TRUE  is true
```

4)

```
NuSMV > check_ctlspec -p "EF(bit0.value=TRUE)"
-- specification EF bit0.value = TRUE  is true
```

5)

```
NuSMV > check_ctlspec -p "EX(bit0.value=TRUE)"
-- specification EX bit0.value = TRUE  is true
```

6)

```
NuSMV > check_ctlspec -p "EF(bit0.carry_out=FALSE)"
-- specification EF bit0.carry_out = FALSE  is true
```

7)

```
NuSMV > check_ctlspec -p "EF(bit0.value=FALSE)"
-- specification EF bit0.value = FALSE  is true
```

8)

```
NuSMV > check_ctlspec -p "AX(bit0.carry_out=TRUE)"
-- specification AX bit0.carry_out = TRUE  is true
```

Problem 4:

Tasks:

1) Understand the design and draw the transition diagram for the design

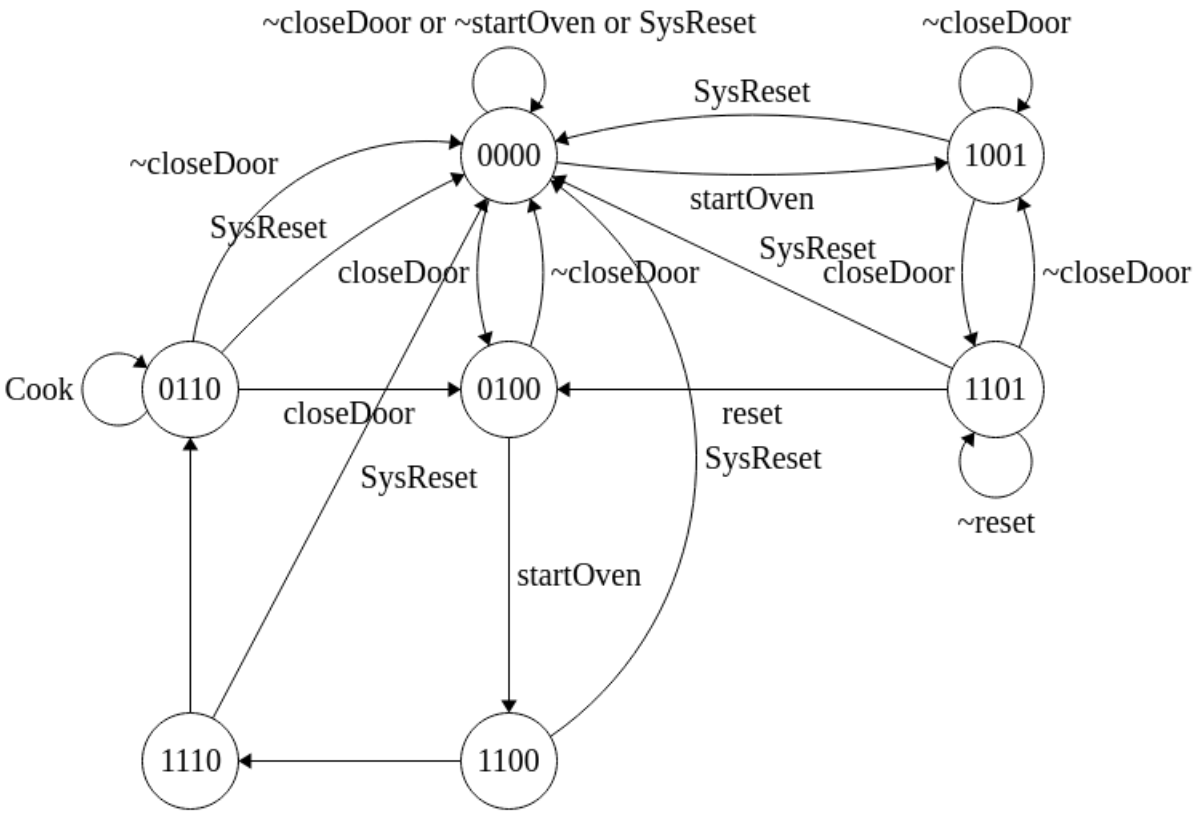


Fig. 1: State Diagram for Microwave

2) Propose 5 new CTL properties. Describe them in English first, then in CTL forms. Your properties should be different from those proposed by other students.

1. If a Start signal occurs then there is eventually a closeDoor signal
2. If a Start signal occurs then there is eventually a not closeDoor signal
3. If eventually a Heat signal occurs then there is eventually a not Heat signal

4. For all paths Heat signal will not occur until Start occurs
5. The signal closedDoor holds infinitely often
6. The signal Heat holds infinitely often

3) Create Verilog test benches and run simulation to verify your properties, justify your test result. (show the test result in script form instead of wave form)

4) Model the design in NuSMV

5) Prove/disprove them by the NuSMV package.

6) Explain why the property is true or false and if the test result meets your expectation.

7) Compare the model checking to simulation, and discuss the advantages and disadvantages of the above two verification methods.