Portland State University

Project 2

ECE582

Model Checking by NuSMV

Erik Rhodes Jordan Fluth

August 12, 2014

1 Combinational Lock

The combinational lock was modeled in NuSMV, using the operational parameters defined in the project description, in order to verify the properties below. We hypothesized that the five properties below were true for the combinational lock system. The results of the NuSMV simulation indicate that our properties were in fact true.

1.1 Added properties

Property 1: AG((open * up * down) $\rightarrow$ (open))
Description: If the lock is open and it is not interrupted, it can remain open indefinitely.

Property 2: AG(state=1 * open * down * up * position=21) $\rightarrow$ (state=2)
Description: A transition to state two can only happen under the required circumstances.

Property 3: AG((open * up * down) $\vee$ (open * down * up)) $\rightarrow$ AX(open)
Description: If the lock is open and it is interrupted the the lock will close.

Property 4: AG(EF(open))
Description: It is always possible for the lock to be closed.

Property 5: AG(down * position=15) $\rightarrow$ (open)
Description: The lock will not open by counting down to 15.

1.2 Results

Listing 1: NuSMV Output

```
-- specification (AG ((open & !up) & !down) -> open)  is true
-- specification (AG ((((state = left_12 & !open) & down) & !up) & position = 21)
   -> state = right_21)  is true
-- specification (AG (((open & up) & !down) | ((open & down) & !up)) -> AX !open)
    is true
-- specification AG (EF !open)  is true
-- specification (AG (down & position = 15) -> !open)  is true
```

1.3 Proofs

The properties mentioned above are explained in figure 1.

1. In state three the lock is open and if neither up, nor down is asserted then the lock continues to stay in state three and remain open.

2. In order for a transition to state two, you must have been in state one, the position of the count must be at 21, and you must have reached 21 by counting down.

3. In state three the lock is open, the only way to transition from state three, and thus close the lock, is by moving the position count up or down. If either of these happen there will be a transition to state zero and the lock will be closed.

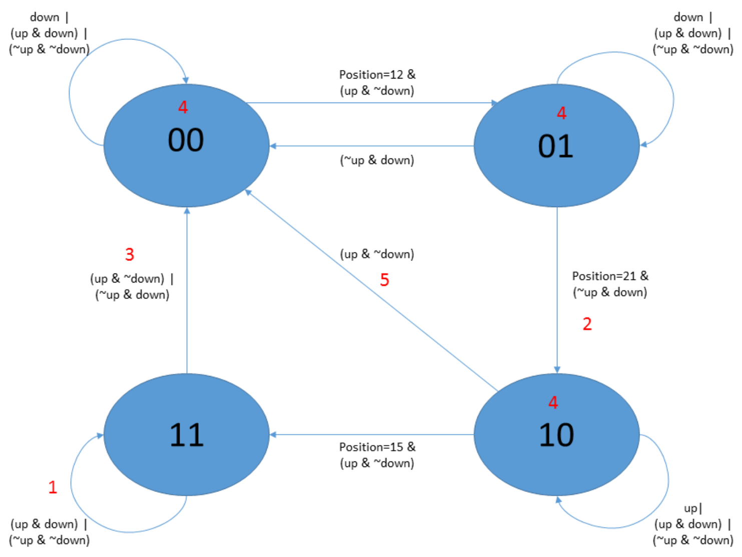

Figure 1: Diagram showing Lock States

4. In states zero, one, and two the lock is closed. State three is the only state in which the lock is open. Since there is a transition out of state three that is possible then there exist the possibility for the lock to be closed.
5. If in state two and counting up occurs there will be a transition to state zero. Therefore the lock will not be opened.

2 Kripke Structures

2.1 Structure 1

2.1.1 Added properties

Property 1: AG(state=3) → AX(state=4)
Description: From state three, the next transition must be to state four.

Property 2: AG(state=4) → AX(state=2)
Description: From state four, the next transition must be to state two.

Property 3: AG(state=5) → EG(state=5)
Description: From state five, it is possible to remain in state five indefinitely.

Property 4: AG(state=1) → EG(state=5)
Description: From state one, the next state cannot be five.

Property 5: AG(state=2 — state=5) → AX(state=3)
Description: State three is reachable by state two or state five.

2.1.2 Results

Listing 2: NuSMV Kripke Diagram Output

```
-- specification (AG state = 3 -> AX state = 4)  is true
-- specification (AG state = 4 -> AX state = 2)  is true
-- specification (AG state = 5 -> EG state = 5)  is true
-- specification (AG state = 1 -> !(EG state = 5))  is true
-- specification (AG (state = 2 | state = 5) -> AX state = 3)  is true
```

2.1.3 Proofs

The properties mentioned above are explained in figure 2.

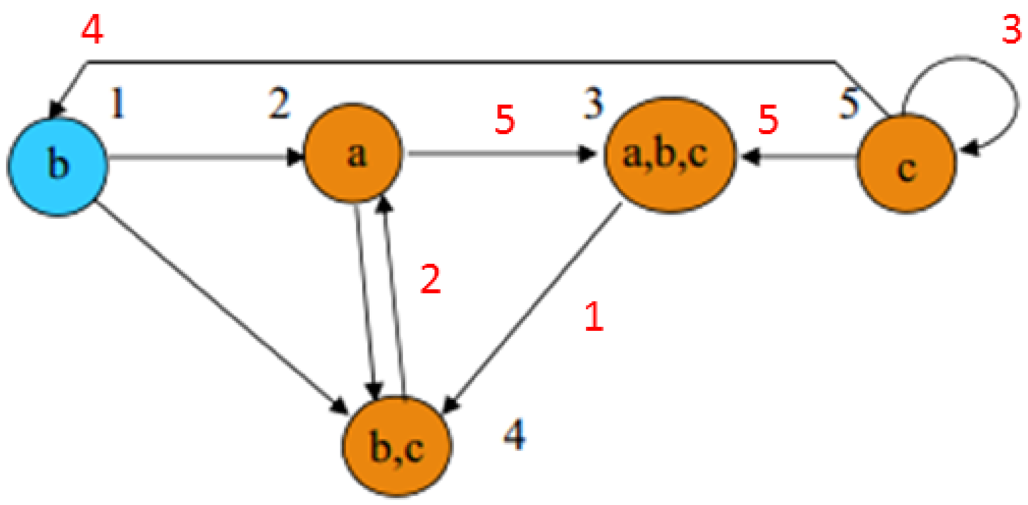

Figure 2: State Diagram Explaining Results

1. If in state three there is only one possible transition. When in state three the next state will be state four.
2. If in state four there is only one possible transition. When in state four the next state will be state two.
3. If in state five one possible transition loops back to state five, therefore once you get in state five it is always possible to remain in state five.
4. If in state one, there is no transition to state five. Therefore, there is no path that leads from state one, to state five as the next state.
5. If in state two, or state five, there is a possible transition to state three.

2.2 Figure 2

2.2.1 Added properties

Property 1: AG(state=6) → AX(state=7)
Description: From state six, the next transition must be to state seven.

Property 2: AG(state=7) → AX(state=4)
Description: From state three, the next transition must be to state four.

Property 3: AG(state=3 * door) → AX(state=1)
Description: If in state three and the door is open then the next state, is state one.

Property 4: AG(state=5 * reset) → AX(state=3)
Description: If in state five and a reset is signaled then the next state, is state three.

Property 5: AG(state=4 * cook) → AX(state=4)
Description: If in state four and it is cooking then it will remain in state four.

2.2.2 Results

Listing 3: NuSMV Kripke Diagram Output

```
-- specification (AG state = 6 -> AX state = 7)  is true
-- specification (AG state = 7 -> AX state = 4)  is true
-- specification (AG (state = 3 & open_door) -> AX state = 1)  is true
-- specification (AG (state = 5 & reset) -> AX state = 3)  is true
-- specification (AG (state = 4 & cook) -> AX state = 4)  is true
```

2.2.3 Proofs

The properties mentioned above are explained in figure 3.

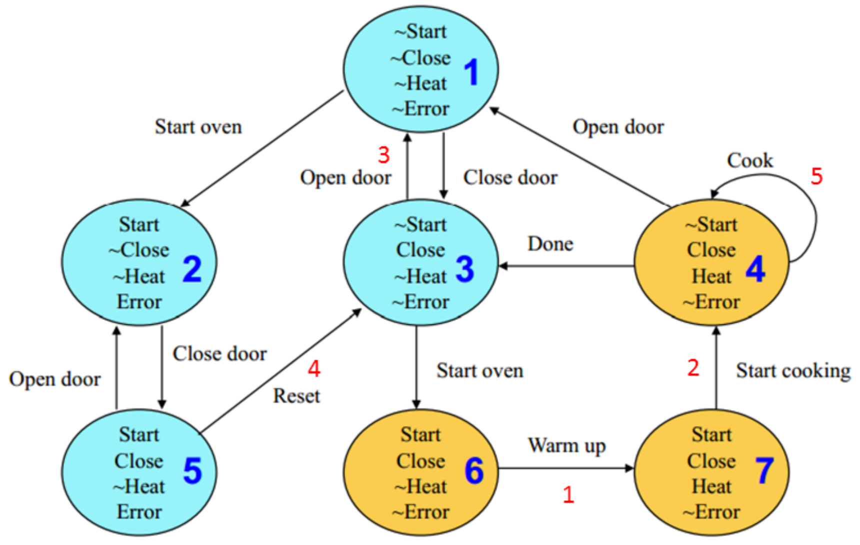
~Start ~Close ~Heat ~Error 1
Start ~Close ~Heat Error 2
~Start Close ~Heat ~Error 3
~Start Close Heat ~Error 4
Start Close ~Heat Error 5
Start Close ~Heat ~Error 6
Start Close Heat ~Error 7
Start oven
Open door
3
Open door
Close door
Cook
5
Done
Close door
Open door
4
Reset
Start oven
2
Start cooking
Warm up
1

Figure 3: Oven State Diagram

1. If in state six there is only one possible transition. When in state six, the warm up signal is asserted, and the next state will be state seven.
2. If in state seven there is only one possible transition. When in state seven, the start cooking signal is asserted, and the next state will be state four.

3. If in state three and the door open signal is asserted, then the next state will be state one.
4. If in state five and the reset signal is asserted, then the next state will be state three.
5. If in state four and the cooking signal is asserted, then the next state will be state four.

3 Equivalence Checking

We performed equivalence checking by model checking on the sequential design from homework 2.

3.1 Results

The results from the command line after running `nuSMV -int p3.smv` are:

Listing 4: NuSMV Interactive Output

```
NuSMV > go
NuSMV > pick_state
NuSMV > print_reachable_states
######################################################################
system diameter: 4
reachable states: 9 (2^3.16993) out of 64 (2^6)
######################################################################
```

When running the simulation normally, we see that `AG(out=0)` is false:

Listing 5: NuSMV Interactive Output

```
-- specification AG !out  is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
-> State: 1.1 <-
  state = S0
  x = FALSE
  out = FALSE
-> State: 1.2 <-
  state = S10
  out = TRUE
```