15-820-a
Assignment 3
Using SMV

Due March 5th, 2003

1 Informal Description

Following is a description of an elevator controller. In this exercise you will specify, implement, and check this controller using the SMV model checker.

The elevator spans three floors. In each floor there is a button that calls the elevator. Inside the elevator there are three buttons, one for each floor. The controller should include a main module, and a floor module. There should be one instance of the elevator module, called el, and three instances of the floor module, called `floor1`, `floor2`, and `floor3`.

The main module should have the following signals:

- `lift` - This signal holds the current location of the elevator, and can have one of the following values: {`at1`, `at2`, `at3`, `bet12`, `bet23`}, where `ati` means the lift is at floor number i, and `betij` means the lift is between floors i and j.
- `doors` - describes the status of the elevator doors, gets values in the domain {`open`, `closed`}.
- `moving` - a Boolean variables that is true when the elevator is moving, and false otherwise.
- `direction` - This signal tells us whether the elevator is on its way up or down. It takes values from the domain {`up`, `down`}.
- `Sensor` - This Boolean signal describes the elevator sensor that senses when people enter or exit the elevator.

The floor module should have the following signals:

- `floor_button` - This signal shows the status of the button on the floor. It takes on one of the values {`on`, `blink`, `off`} - on means the button is lit, blink means the button is blinking, and off means the button is not lit.
- `elev_button` - This signal shows the status of the button that requests this floor from within the elevator. When a person requests floor1 from within the elevator, the signal `floor1.elev_button` becomes true.

2 Part A - The Specification

Each of the following natural language specifications describes a property of the elevator controller. Translate each of these into a CTL formula (notice - the formula should be in CTL and not CTL*). Since this is a specification, it should be implementation independent. Specifically, the formulas may not refer to signals that are not defined above.

1. When the `floor_button` signal of any floor is on or blinking, it will stay that way until the doors open on the right floor.
2. If `floor_button` is not off, it will blink when the elevator is moving and be on when it is not.
3. When the door opens at any floor it stays open for at least three time units.
4. If the door is open, it must stay open until the sensor has been false for at least two time units. The door will close one time unit after this happens.
5. When the door is open, eventually the sensor will be false for two consecutive time units.
6. If the elevator is requested at a given floor, it will eventually reach that floor.
7. When the elevator doors are open the elevator must not move (i.e., will not change location) and both buttons of that floor are off.
8. The elevator must not move at the first time unit after the doors have closed.
9. It takes the elevator two or three time units to move between floors (i.e., after one time unit of moving it will be between floors, and within one or two extra time units it will be in the next floor).

10. People enter the elevator (making sensor true) only when the doors are open.

    The following specifications determine which direction the elevator moves when the doors close.

11. If there is no request to another floor, the elevator should not move.
12. The elevator cannot change direction between floors.
13. If the elevator is on the second floor and there are requests for both the first and the third floor, the elevator will continue its current direction (if it came from the bottom floor it will go to the top and vice versa.
14. If the elevator is in a requested floor it will not leave this floor before the doors open.

    Liveness specifications:

15. It is possible that there are requests for the top and bottom floors at the same time and the elevator chose to go to the top floor first.
16. It is possible that there are requests for the top and bottom floors at the same time and the elevator chose to go to the bottom floor first.

3 Part B - The Implementation

Implement the controller in the SMV language. Make sure to define the elevator signals in the main module, and to define instances of the floor module called `floor1`, `floor2`, and `floor3`. The implementation should adhere to all of the specifications above (both numbered and un-numbered). When the behavior is not fully specified you may choose a behavior that makes sense to you.

Check your implementation using SMV. Use fairness to assume that the elevator does not get stuck in a single floor forever.

4 Client-Server Application from Class

Karen presented a client server application during class (slides 20-22). There is an error in the implementation of the server module.

1. Write a CTL specification that describes a desired behavior of the client server example, and show how it fails on the given implementation.

2. Fix the implementation and use SMV to prove the corrected implementation.

5 Submission

Submit the solutions in SMV format. Please label (comment) all formulae clearly.