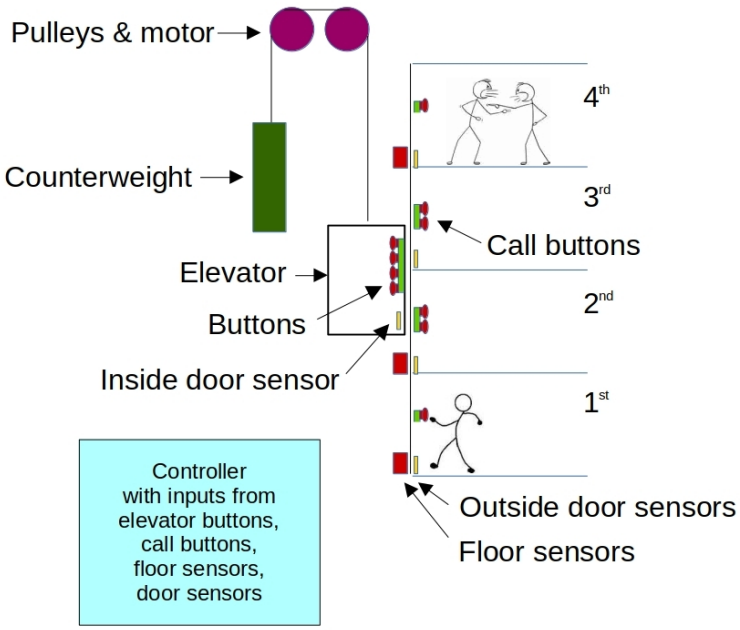
Lab: Elevator Model

An elevator services four floors numbered 1 through 4. There is a (outer) door at each floor, there are two call-buttons for up/down on floors 2 and 3 and one call-button on floors 1 and 4, all call-buttons have lights that indicate the button has been pressed. In the elevator there is a (inner) door, four floor-select buttons (one per floor), and four lights indicating floors selected by passengers in the elevator. There is a sensor, called a floor sensor, on each floor that sends a signal to the elevator controller when the elevator is aligned with that floor or a different signal that it is no longer aligned. There is an outside door sensor on every floor that sends a signal to the controller when the outer floor door is opened or closed. There is a door sensor on the elevator that sends a signal when the elevator door is opened or closed. The controller opens/closes doors, activates and de-activates lights, and activates the motor sitting atop pulleys to cause the elevator to ascend, descend, and stop. The controller also remembers button presses. See the figure below.



The system must have these properties:

1. A floor door is never open if the elevator is not present at that floor.
2. The elevator door is open only when stopped at a floor and when the outside door of that floor is also open.
3. The elevator door is always closed when in motion. The elevator does not begin to move until all doors are closed.
4. A floor requested by a waiting person will be served sometime - that is, a person cannot cancel a call.

5. When the top (penthouse) floor is requested (either from within the elevator or by a waiting party), that floor is served immediately - that is, the elevator does not stop on the way there.
6. Otherwise, if the elevator is descending and in use, it serves floors that have been called in the order they are reached by the elevator until there are no more waiting calls on any floor beneath the elevator's current position.
7. If the elevator is ascending and in use, it serves floors that have been called in the order they are reached by the elevator until there are no more waiting calls on any floor above the elevator's current position.
8. The elevator always returns to the 1st floor when not in use.
9. The elevator never reverses direction while it is in motion.
10. The elevator door and floor doors always open and close together.
11. Buttons inside the elevator cannot be pushed unless someone is in the elevator.

Outputs from the controller:

1. Signals elevator door and floor door simultaneously to open or close.
2. Signals elevator motor to accend, descend, or stop.
3. Uses floor sensors to signal the motor to stop the elevator at a floor.
4. Signals motor to prevent elevator from continuing below 1st floor and above 4th floor.
5. Signals call-button and elevator button lights to turn off when elevator lands on floors.

Exercise 1:

Construct system state in Cryptol. It starts off like this:

```
type State = { cb : [4][12], // call-buttons, 1/floor, 4 total, 0: no call, 1: up, 2: down, 3: both
               cbl : [4][12], // call-button lights, 0: both off, 1: up on, 2: down on, 3: both on
                 ...
               } ■
```

Exercise 2:

Construct an initial `State`. Say the call-button and elevator button lights are all off, no one is in the elevator, the elevator is motionless on the first floor, no one is calling for the elevator, and the elevator doors and first floor doors are all open. The initial State might start out like this:

```
initState : State
initState = {  cb = [0,0,0,0],  // no call buttons pressed
              cbl = [0,0,0,0],  // all call button lights are off
              ods = [0,1,1,1],  // outside floor door is open on 1st floor only
               fs = [1,0,0,0],  // elevator is on 1st floor
                    ...
            }  ■
```

Exercise 3:

Identify legal `States` and create legal transitions between `States`. For example, from `initState` a person may enter the elevator and press the 4th floor button. The next `State` could have the elevator door and 1st floor doors closed, the elevator in motion ascending, the elevator 4th floor light on, all others still off, the 1st floor sensor still sensing the elevator (it has just begun to move), all outside and inside door sensors are off (all doors are closed). The next `State` could also have the 4th floor down button pressed, all doors closed, the elevator is in motion ascending, the 1st floor sensor still sensing the elevator and so on. Create a function that takes two `States` as input and returns `True` if the second `State` is a possible

next `State` to the first `State`. Note that numbers of the `State` parameters must be legal (cannot go higher than 4th floor or lower than first floor, for example). ■

Exercise 4:
Formulate properties that express required features of the elevator such as the elevator never reverses direction while it is in motion. ■

Exercise 5:
Prove those properties to be true. ■

Note: `solutions.pdf` contains all of the above for the puzzle of the fox, chicken, corn and a farmer who wants to cross a river with a two object capacity rowboat under restrictions that the fox must not be left alone with the chicken and the chicken must not be left alone with the corn. The solution to this problem follows that one although it is much more complicated.