```
──────────────────────── MODULE Elevator ────────────────────────
EXTENDS Integers

CONSTANT NumFloors

VARIABLES
     Current position of the elevator car
    position,
     State of the down button for a given floor
    down,
     State of the up button for a given floor
    up,
     Drop off requesetd for a given floor
    destinations

Floors ≜ 0 . . NumFloors − 1

vars ≜ ⟨position, down, up, destinations⟩

TypeOK ≜ ∧ position ≥ 0
         ∧ position < NumFloors
         ∧ down ∈ [Floors → BOOLEAN ]
         ∧ up ∈ [Floors → BOOLEAN ]
         ∧ destinations ∈ [Floors → BOOLEAN ]

To start the car is on the ground floor, there are no up or down calls, and no floor is selected
Init ≜ ∧ position     = 0
       ∧ down = [f ∈ Floors ↦ FALSE]
       ∧ up     = [f ∈ Floors ↦ FALSE]
       ∧ destinations = [f ∈ Floors ↦ FALSE]


As long as the car is not already on the floor, and there is either an up or down call, the car can
move directly to that floor
MoveToFloor(f) ≜ ∧ position ≠ f
                 ∧
                    ∨ up[f] = TRUE
                    ∨ down[f] = TRUE
                 ∧ position′ = f
                 ∧ UNCHANGED ⟨up, down, destinations⟩

A user going down calls the elevator, as long as the car is not already on that floor, and the car
has not already been called, down[f] becomes true
DownCall(f) ≜ ∧ f ≠ 0
              ∧ down[f] = FALSE
```

$$\begin{array}{l}
\land\ down' = [down\ \textsc{except}\ ![f] = \textsc{true}]\\
\land\ \textsc{unchanged}\ \langle up,\ position,\ destinations \rangle
\end{array}$$

Up Call is defined similar to *DownCall*

$$UpCall(f)\ \triangleq\ \begin{array}[t]{l}
\land\ f \neq NumFloors - 1\\
\land\ up[f] = \textsc{false}\\
\land\ up' = [up\ \textsc{except}\ ![f] = \textsc{true}]\\
\land\ \textsc{unchanged}\ \langle down,\ position,\ destinations \rangle
\end{array}$$

The elevator may pickup a passenger going either direction provided the car is on that floor and there is a passenger waiting. Their destination floor is set to true in destinations

$$PickupGoingUp(f,\ destination)\ \triangleq\ \begin{array}[t]{l}
\land\ position = f\\
\land\ up[f] = \textsc{true}\\
\land\ up' = [up\ \textsc{except}\ ![f] = \textsc{false}]\\
\land\ destinations' = [destinations\ \textsc{except}\ ![destination] = \textsc{true}]\\
\land\ \textsc{unchanged}\ \langle down,\ position \rangle
\end{array}$$

$$PickupGoingDown(f,\ destination)\ \triangleq\ \begin{array}[t]{l}
\land\ position = f\\
\land\ down[f] = \textsc{true}\\
\land\ down' = [down\ \textsc{except}\ ![f] = \textsc{false}]\\
\land\ destinations' = [destinations\ \textsc{except}\ ![destination] = \textsc{true}]\\
\land\ \textsc{unchanged}\ \langle up,\ position \rangle
\end{array}$$

When the elevator is on a given floor, and that floor is in destinations, destianations for that floor moves to false to indicate passengers have been dropped of

$$Dropoff(f)\ \triangleq\ \begin{array}[t]{l}
\land\ position = f\\
\land\ destinations[f] = \textsc{true}\\
\land\ destinations' = [destinations\ \textsc{except}\ ![f] = \textsc{false}]\\
\land\ \textsc{unchanged}\ \langle position,\ up,\ down \rangle
\end{array}$$

Next state transition is: The elevator car may move to a floor, be called by a passenger going up or down, and pickup or drop off passenges

$$Next\ \triangleq\ \begin{array}[t]{l}
\lor\ \exists f \in Floors : MoveToFloor(f)\\
\lor\ \exists f \in Floors : DownCall(f)\\
\lor\ \exists f \in Floors : UpCall(f)\\
\lor\ \exists f \in Floors,\ dest \in Floors : PickupGoingUp(f,\ dest)\\
\lor\ \exists f \in Floors,\ dest \in Floors : PickupGoingDown(f,\ dest)\\
\lor\ \exists f \in Floors : Dropoff(f)
\end{array}$$

This temporal formula for liveness states that if an up call occurs on a given floor, the passenger must evetually be picked up, which is indicated by the up call being cleared

$$Liveness\ \triangleq\ \begin{array}[t]{l}
\land\ \forall f \in Floors : (up[f] = \textsc{true}) \leadsto (up[f] = \textsc{false})\\
\land\ \forall f \in Floors : (destinations[f] = \textsc{true}) \leadsto (destinations[f] = \textsc{false})
\end{array}$$

$$
\begin{aligned}
\mathit{Fairness} \triangleq\ & \land \forall f \in \mathit{Floors},\ \mathit{dest} \in \mathit{Floors} : \mathrm{SF}_{\mathit{vars}}(\mathit{PickupGoingUp}(f,\ \mathit{dest})) \\
& \land \forall f \in \mathit{Floors},\ \mathit{dest} \in \mathit{Floors} : \mathrm{SF}_{\mathit{vars}}(\mathit{PickupGoingDown}(f,\ \mathit{dest})) \\
& \land \forall f \in \mathit{Floors} : \mathrm{SF}_{\mathit{vars}}(\mathit{Dropoff}(f)) \\
& \land \forall f \in \mathit{Floors} : \mathrm{WF}_{\mathit{vars}}(\mathit{MoveToFloor}(f)) \\
& \land \forall f \in \mathit{Floors} : \mathrm{WF}_{\mathit{vars}}(\mathit{UpCall}(f)) \\
& \land \forall f \in \mathit{Floors} : \mathrm{WF}_{\mathit{vars}}(\mathit{DownCall}(f))
\end{aligned}
$$

$$\mathit{Spec} \triangleq \mathit{Init} \land \Box[\mathit{Next}]_{\mathit{vars}}$$

$$\mathit{FairSpec} \triangleq \mathit{Spec} \land \mathit{Fairness}$$

* Modification History
* Last modified *Mon* May 21 21:10:20 *PDT* 2018 by *jennmat*
* Created *Thu* May 10 21:35:52 *PDT* 2018 by *jennmat*