Building Management System

Dr Alun Moon

3rd February 2021

1 Requirements

To track people as they enter and leave a building.

2 Specification

MODULE *building*

Sample solution for first TLA+ exercise

CONSTANT

People — we're dealing with people here; this is the set of all people

VARIABLE

register, — Set of registered users

in, — Set of people in the building

out — Set of people out of the building

$TypeOK \triangleq$ — type invarient

$\land \quad register \subseteq People$ — Everyone on the register is a person

$\land \quad register = in \cup out$ — everyones location is known

$\land \quad in \cap out = \{\}$ — noone can be both in and out of the building

---

$Init \triangleq$

$\land\ register = \{\}$ — Initially no-one is registered

$\land\ in \quad = \{\}$ — no-one is inside

$\land\ out \quad = \{\}$ — no-one is outside

$Register(p) \triangleq$

$\land \quad p \in People \setminus register$ — p is a person and not registered

$\land\ register' = register \cup \{p\}$ — add p to register

$\land\ out' = out \cup \{p\}$ — p is outside

$\land\ in' = in$ — must keep set of those inside the same

$Enter(p) \triangleq$

$\land\ p \in out$ — p is outside the building
$\land\ in' = in \cup \{p\}$ — add p to the inside set
$\land\ out' = out \setminus \{p\}$ — remove p from the outside set
$\land\ register' = register$ — register is unchanged

$Leave(p) \triangleq$
$\land\ p \in in$ — p is in the building
$\land\ in' = in \setminus \{p\}$ — remove p from the inside set
$\land\ out' = out \cup \{p\}$ — add p to the outside set
$\land\ register' = register$ — resigter is unchanged

$Next \triangleq$
$\exists\, p \in People :$ — There is a person who can either
$\lor\ Register(p)$ — be registered, or
$\lor\ Enter(p)$ — enter the building, or
$\lor\ Leave(p)$ — leave the building

Modification History
Last modified Wed Oct 02 10:31:48 BST 2019 by alun
Last modified Tue Sep 10 12:27:57 BST 2019 by cgam1
Created Mon Sep 24 11:53:39 BST 2018 by cgam1

3 Model

What is the Model The model defines the constant *People*.
People $\leftarrow$ {"Alun", "Neil", "David", "Michael"}

What is the Behaviour spec? The behaviour specification is given by an *Initial predicate and next-state relation*

Init *Init*

Next *Next*

Invariants The invariants checked are :

TypeOK	The type invariant from the specification
$\forall p \in register : p \in People$	every registered person is in People.
$register \subseteq People$	register is a subset of People
$\forall\, p \in out : p \in People$	Everyone outside the building is a person (see next invariant)
$out \subseteq People$	out is a subset of people (says the same thing as the last invariant)
$\forall\, p \in in : p \in register$	everyone in the building is registered
$in \subseteq register$	in is a subset of register

4 Results

A summary of the numbers of states found by the model checking is shown below.

4.1 Statistics

States found for model as a whole

States Found	325
Distinct States	81

Number of next states found for the actions is:

Action	**States found**
Init (line 18)	1
Register (line 23)	108
Enter (line 29)	108
Leave (line 35)	108

5 Discussion

The (simple) building model has three state variables; the register of users, the list of people inside the building, and the list of those outside the building. The model has the set of people the specification applies to.

There is a redundancy in the state variables, and consistency is enforced by the type invariant $register = in \cup out$

The Next action can be interpreted as follows.

> There is a person, who can either; be registered, or can enter the building, or can Leave the building.

$$
\begin{array}{ll}
Next \triangleq & \\
\quad \exists\, p \in People : & \text{There is a person who can either} \\
\qquad \lor\ Register(p) & \quad \text{be registered, or} \\
\qquad \lor\ Enter(p) & \quad \text{enter the building, or} \\
\qquad \lor\ Leave(p) & \text{leave the building}
\end{array}
$$