

1 Address Book — v1

1.1 The Specification

```
MODULE addressbook
CONSTANTS Names, Emails
VARIABLE address

Invariant  $\triangleq$  address  $\in$  [Names  $\rightarrow$  Emails]

Init  $\triangleq$  address =  $\langle \rangle$   A different ideom for an empty function...

Add(name, email)  $\triangleq$ 
    address' = [n  $\in$  DOMAIN address  $\cup$  {name}  $\mapsto$ 
        IF n  $\in$  DOMAIN address
        THEN address[n]
        ELSE email
    ]

Remove(name)  $\triangleq$ 
    address' = [n  $\in$  DOMAIN address  $\setminus$  {name}  $\mapsto$  address[n]]

Find(name)  $\triangleq$ 
     $\wedge$  name  $\in$  DOMAIN address
     $\wedge$  UNCHANGED address

Next  $\triangleq$ 
     $\exists$  n  $\in$  Names, e  $\in$  Emails :
         $\vee$  Add(n, e)
         $\vee$  Remove(n)
         $\vee$  Find(n)
```

Modification History

Last modified Sat Feb 13 10:50:44 GMT 2021 by alunm

Created Wed Feb 10 21:49:46 GMT 2021 by alunm

2 The Model

2.1 Model Overview

The Behaviour specification is an *Initial-predicate and next-state relation*

Initial Predicate *Init*

Next-state relation *Next*

The Model values assigned to declared constants

Set of Names is set to

$$Names \leftarrow \{ "a", "b" \}$$

Set of email addresses is set to

$$Emails \leftarrow \{ "1", "2" \}$$

2.2 Checks and verifications

No invariants were checked.

2.3 Results

A summary of the results

Statistics a summaries of the actions and number of states found.

States found 97

Distinct states 9

Action	Location	States Found	Distinct states
<i>Init</i>	Line 7	1	1
<i>Add</i>	Line 9	36	6
<i>Remove</i>	Line 16	36	2
<i>Find</i>	Line 19	24	0

2.4 Discussion

2.4.1 Model description

The state of the system is . A function that maps from names to addresses.

The initial conditions are an empty function, which can be also written as an empty sequence.

The Next relation is that a name and address can be added or removed from the function, or an address found.

2.4.2 Comments

Using a function we run into some of its limits. With the invariant stated, the domain is a set of names; an initial value of an empty function does not have its domain in the set described by the invariant.



Using a function to model the look-up does work and is the right thing. The problem with this case is that the initial condition of an empty function is not strictly in the set of functions stated in the invariant. The problem here is stating the right invariant.

2.4.3 Interpretation of results

The number of distinct states found for the *Remove* operation is initially surprising, however it is correct, the state of the system after an entry has been removed is the same as the state before the entry was added.