

## Homework V Sample Solution

There are numerous ways to solve this problem, so your solution might be quite different than this one.

The basic types are unchanged so the first difference is the state space where an additional variable is incorporated to identify the listed phones, including an invariant condition pertaining to it.

PhoneDB
members: $\mathbb{P}$ Person
listed: $\mathbb{P}$ Phone
telephones: Person $\square$ Phone
dom telephones $\square$ members
listed $\square$ ran telephones

It might be desirable to also change the initial state schema. However, this is actually not required -- since telephones =  $\emptyset$  and listed  $\square$  ran telephones, it can be inferred that listed =  $\emptyset$ .

Hence we proceed to the specifications of the new operations.

UnList
$\square$ PhoneDB
people?: $\mathbb{P}$ Person
number?: Phone
number? $\square$ listed
people? = telephones $\sim$ ({number?})
listed' = listed $\setminus$ {number?}
telephones' = telephones
members' = members

Since there are obviously exceptional conditions for the UnList operation, we add a schema to treat them.

UnListFailure
$\square$ PhoneDB people?: $\mathbb{P}$ Person number?: Phone rep!: Report
<hr/> number? $\square$ listed $\vee$ people $\neq$ telephones $\sim$ ( $\{\{number?\}\}$ ) rep! = UnList_Error

The completed operation is then defined by a schema expression in the usual way.

DoUnList = $\wedge$ = UnList  $\wedge$  Success  $\vee$  UnListFailure

Now we go on to the second new operation.

ReList
$\square$ PhoneDB people?: $\mathbb{P}$ Person number?: Phone
<hr/> number? $\square$ ran telephones $\wedge$ number? $\square$ listed people? = telephones $\sim$ ( $\{\{number?\}\}$ ) listed' = listed $\square$ {number?} telephones' = telephones members' = members

Again we need an exceptions schema to put together the complete operation specification.

ReListFailure
$\square$ PhoneDB people?: $\mathbb{P}$ Person number?: Phone rep!: Report
<hr/> number? $\square$ listed $\vee$ number? $\square$ ran telephones $\vee$ people $\neq$ telephones $\sim$ ( $\{\{number?\}\}$ ) rep! = UnList_Error

DoReList = $\wedge$ = ReList  $\wedge$  Success  $\vee$  ReListFailure

In addition, we need to consider changes that are necessitated to other operation schemas to reflect and support this new feature. There turn out to be several that are warranted.

First of all, the AddEntry operation must be updated for the new state space.

```

AddEntry _____
□PhoneDB
name?: Person
newnumber?: Phone
_____
name? □ members
name? ↦ newnumber? □ telephones
telephones' = telephones □ {name? ↦ newnumber?}
members' = members
listed' = listed □ {name? ↦ newnumber?}

```

This leaves the FindPhones and FindNames operations to be modified to honor refusing to reveal information about unlisted phones. First, the normal FindPhones operation makes sure all phones are listed.

```

FindPhones _____
□PhoneDB
name?: Person
numbers!: P Phone
_____
name? □ dom telephones
telephones( {name?} ) □ listed
numbers! = telephones( {name?} )

```

Since this introduces an additional exception (i.e., the presence of an unlisted phone), we add a schema to treat it, plus revise the schema expression for the completed operation. This approach reports the listed numbers, and indicates the existence of an unlisted number.

```

UnListedPhone _____
□PhoneDB
name?: Person
numbers!: P Phone
rep! : Report
_____
name? □ dom telephones
telephones( {name?} ) \ listed ≠ {}
numbers! = listed □ telephones( {name?} )
rep! = Unlisted_Phone

```

DoFindPhones =^= FindPhones  $\wedge$  Success  
                   $\vee$  UnknownName  $\vee$  UnlistedNumber

Lastly, we have similar changes for the FindNames operation.

```
FindNames _____  
| PhoneDB  
| names!: P Person  
| number?: Phone  
|  
|-----  
| number?  $\square$  listed  
| names! = telephones~({number})  
|-----
```

An exceptional case is added so that for a known but unlisted number, no names are revealed.

```
UnlistedNumber _____  
| PhoneDB  
| number?: Phone  
| rep!: Report  
|  
|-----  
| number?  $\square$  ran telephones \ listed  
| rep! = 'Unlisted_Entry'  
|-----
```

DoFindNames =^= FindNames  $\wedge$  Success  
                   $\vee$  UnknownNumber  $\vee$  UnlistedNumber

Of course, the new error messages must also be added to the Report type.

The Zans version of this specification is in the class directory.