We have completed the specification of the first phone database operation, and now continue with those remaining. For the next operation we need to introduce a new formalism. This consists of using a relation to perform mappings as we normally do with functions. For a relation R □ X □ Y, each W □ X is associated with its **relational image** R⦇W⦈ = {y□Y | □ x□W □ xRy}.

The next PhoneDB operation is one of the “lookup” operations.

FindPhones ______________________________
| □PhoneDB
| name?: Person
| numbers!: ℙ Phone
|______________________________
| name?: dom telephones
| numbers! = telephones⦇ {name?} ⦈
|____

The relational image operation allows us to establish the desired post-condition directly, and the required pre-condition is evident. When the pre-condition is not met, the exceptional outcome needs to be explicitly stated. This is handled similarly to the previous case.

UnknownName__________________
| □PhoneDB
| name?: Person
| rep!: Report
|__________________________
| name? □ dom telephones
| rep! = 'Unknown name'
|_________________________________

The pre-condition here is just the negation of that for the FindPhones operation, and the post-condition indicates the error report. Then we again use a schema-formula to define

DoFindPhones ≙ FindPhones □ Success
UnknownName

If we pause to examine the corresponding component of the Miranda animation, we find a clear reflection of the specification.

```
findPhones n (mem, tel) = disp (image tel [n])
doFindPhones (mem, tel) n           || correction added
   = write (findPhones n (mem, tel) ++ "\n\n") (phdb (mem, tel)),
      if member (domain tel) n
   = write "Unknown name\n" (phdb (mem, tel)), otherwise
image f u = [y | (x,y) <-f; member u x]

disp x = "Empty\n", if x = [ ]
       = hd x, if #x = 1
       = hd x ++ "\n" ++ disp (tl x), otherwise
```

We continue with the operation for looking up names. With the state space adopted, this leads us to the use of the relational inverse (or transpose) operation.

```
FindNames ______________________________
| □PhoneDB
| names!: P Person
| number?: Phone
|__________________________________
| number □ ran telephones
| names! = telephones~⦇{number}⦈
|_________________________________________
```

The notation in Z for relational inverse is the postfix operator '~'. For relation R □ X □ Y and each W □ Y, this is defined as R~(W) = { x | xRy and y□W}. The pre-condition for FindNames insures that the names! result will be a non-empty set.

As was done with the FindEntry operation, we complete the specification by describing error handling.

```
UnknownNumber ________________________
| □PhoneDB
| number?: Phone
| rep!: Report
|______________________________
| number □ ran telephones
| rep! = 'Unknown number'
|_______________________________________
```

Then the completed specification is

```
   DoFindNames ≙  FindNames □ Success
                    UnknownNumber
```

Again the match with the Miranda animation should be clear.

```
   findNames e (mem, tel) = disp (image (inverse tel) [e])
   doFindNames (mem, tel) e              || correction added
      = write (findNames e (mem, tel) ++ "\n\n") (phdb (mem, tel)),
           if member (range tel) e
      = write "Unknown extension\n" (phdb (mem,tel)), otherwise

   range f = [y | (x,y) <- f]

   inverse f = [(y,x) | (x,y) <- f]
```