Midterm Exam Sample Solutions

Problem 1.

(a) □x:N • □y:N • $x+y^2>x^2+y$ is true since for each x, the existence of a suitable value for y just requires y a little larger than x. For example, if $y=x+2$, then $x+y^2 = x+(x+2)^2 = x^2+5x+4 > x^2+x+2 = x^2+y$.

(b) □y:N • □x:N • $x+y^2>x^2+y$ is false. Suppose some value exists for y, say y=k, so that . □x:N • $x+k^2>x^2+k$ is true. Then □x:N • $k^2-k>x^2-x$ would be true, but this inequality is clearly only true for values of x smaller than k, not *all* values (e.g., not for x=k).

Problem 2.

Assertion (a) is a suitable invariant -- the initial conditions of K=N, X=1 and P=0 satisfy this condition, the loop ends with K=0 yielding the desired conclusion, and the condition can be proven invariant for execution of the loop body.

To see that (b) must fail, just take the values initially assigned for the variables when N=1, namely P=0, X=1, and K=1, and then $K \geq 0 \wedge X*2^N = 2^K \wedge P*X+1 = 2^{N-K}$ is true. But after the loop body executes, K=0, P=1, and X=2 so $X*2^N = 4$ while $2^K = 1$, and $P*X+1=3$ while $2^{N-K}=2$ and hence the assertion fails to provide a valid invariant.

Problem 3.

Apart from the changes indicated here, the schemes remain as they appear in Diller.

In the invariant for the state schema, *add* the condition
ran cell □ ran land = ∅,
and this pledges states where the same number is not both a cell and a land number.

For the AddCell operation properties, *add* the pre-condition
newnumber?□ ran land
to prevent adding an existing land line as a cell phone, preserving the invariant for this operation -- this creates a new exceptional case which the problem does not require to be treated.

By duality for the AddLand operation properties, *add* the pre-condition
newnumber?□ ran cell
preserving the invariant for this operation (likewise creating a new exceptional case).

Problem 4.
There are a variety of ways to express this in Z. For a solution using only sets and relational image -- relational image telephones~⦇{ph}⦈} gives all persons assigned to a phone and can be used to identify those phones whose listing includes *all* the people in the argument set (i.e., shared by all).

It would be an acceptable solution to include no pre-condition for this operation (so no exceptions). If a non-member occurred in the argument set, this would simply lead to the empty set of phones being returned -- a meaningful and reasonable result (which can occur even when only members appear in the argument). However, it is more in the style of Diller's presentation to treat the occurrence of a non-member as an exception and report accordingly, so this is included in this solution.

```
SharedPhones ________________________________
□ PhoneDB
people? : ℙ Person
ans! : ℙ Phone
______________________________
people? □ members
ans! = {ph : Phone | people?□telephones~⦇{ph}⦈}
```

Then the operation completed with the exception is

DoSharedPhones =^= SharedPhones ∧ Success ∨ NonMembers

where NonMembers is a schema just like Diller's NotMember except that its pre-condition is □(people? □ members), the negation of the pre-condition in SharedPhones.

As an alternative solution, the character of the problem is strongly suggestive of descriptions that characterize members of the desired set. This alternative uses relational image with telephones (rather than telephones~) in an axiomatic description -- a phone belongs to the answer set if and only if for all x in people, the phone is in the telephones image (i.e., shared by them all).

```
SharedPhones ________________________________
□ PhoneDB
people? : ℙ Person
ans! : ℙ Phone
ph : Phone
______________________________
 people? □ members
 ph□ans! ⇔ (□x:Person • x□people? □  ph□telephones⦇{x}⦈)
```

The first two descriptions are existentially based over either the entire collection Person or the entire collection Phone. A third solution can be given using generalized intersection. This was not discussed in class, but is covered in Diller, and takes a single argument that is a set of sets and gives their mutual intersection. This solution alternative is more direct than the first two as it requires consideration of only the set of people given by the argument, and the intersection operation assures membership for all indicated people.

```
SharedPhones ____________________________________
□ PhoneDB
people? : ℙ Person
ans! : ℙ Phone
_______________________________
people? □ members
ans! = □ {telephones⦇{x}⦈ | x□people?}
_______________________________________________
```