```
phone0: THEORY

BEGIN

 N: TYPE
 P: TYPE
 B: TYPE = [N->P]

 nonumber: P
 emptybook: B = LAMBDA(nm:N): nonumber

 FindPhone: [B,N -> P]
 FindPhoneAx: AXIOM FORALL (bk:B), (nm:N):
    FindPhone(bk,nm) = bk(nm)

 AddPhone: [B, N, P -> B]
 AddPhoneAx: AXIOM FORALL (bk:B), (nm: N), (pn:P):
    AddPhone(bk,nm,pn) = bk WITH [(nm):=pn]

 FindAdd: CONJECTURE FORALL (bk:B), (nm:N), (pn:P):
          FindPhone(AddPhone(bk,nm,pn), nm) = pn

 DelPhone: [B, N -> B]
 DelPhoneAx: AXIOM FORALL (bk:B), (nm: N):
    DelPhone(bk,nm) = bk WITH [(nm):=nonumber]

 % problem with following conjecture: DellAdd does not restore previous state
 DelAdd: CONJECTURE FORALL (bk:B), (nm:N), (pn:P):
          DelPhone(AddPhone(bk,nm,pn), nm) = bk

END phone0
```

```
phone1: THEORY

BEGIN

 N: TYPE
 P: TYPE
 B: TYPE = [N->P]

 nonumber: P
 emptybook: B = LAMBDA(nm:N): nonumber

 FindPhone: [B,N -> P] = LAMBDA (bk:B), (nm:N): bk(nm)

 AddPhone: [B, N, P -> B] = LAMBDA (bk:B), (nm: N), (pn:P):
             bk WITH [(nm):=pn]

 FindAdd: CONJECTURE FORALL (bk:B), (nm:N), (pn:P):
          FindPhone(AddPhone(bk,nm,pn), nm) = pn


END phone1
```

```
phone2: THEORY

BEGIN

 N: TYPE
 P: TYPE
 B: TYPE = [N->P]

 nonumber: P
 emptybook: B = LAMBDA(nm:N): nonumber

 FindPhone: [B,N -> P] = LAMBDA (bk:B), (nm:N): bk(nm)

 AddPhone: [B, N, P -> B] = LAMBDA (bk:B), (nm: N), (pn:P):
        bk WITH [(nm):=pn]

 FindAdd: CONJECTURE FORALL (bk:B), (nm:N), (pn:P):
        FindPhone(AddPhone(bk,nm,pn), nm) = pn

 DellPhone: [B,N -> B] = LAMBDA  (bk:B), (nm: N):
        bk WITH [(nm):= nonumber]

 DellAdd1: CONJECTURE FORALL (bk:B), (nm:N), (pn:P):
        FindPhone(bk,nm) = nonumber IMPLIES
                DellPhone(AddPhone(bk,nm,pn),nm) = bk

 DellAdd2: CONJECTURE FORALL (bk:B), (nm:N), (pn:P):
        DellPhone(AddPhone(bk,nm,pn),nm) = DellPhone(bk,nm)

 % the conjecture causes a problem: no restriction on nm in AddPhone
 KnownAdd: CONJECTURE FORALL (bk:B), (nm:N), (pn:P):
        FindPhone(AddPhone(bk,nm,pn),nm) /= n0

END phone2
```

```
phone3: THEORY

BEGIN

 N: TYPE
 P: TYPE
 B: TYPE = [N->P]


 nonumber: P
 emptybook: B = LAMBDA(nm:N): nonumber

 % Add new type for phone numbers different from nonumber

 GP: TYPE = {pn: P | pn /= nonumber}

 FindPhone: [B,N -> P] = LAMBDA (bk:B), (nm:N): bk(nm)

 % A new type: names with valid phone numbers
 Known?: [B,N -> bool] = LAMBDA (bk:B), (nm:N):
        bk(nm) /= nonumber

 AddPhone: [B, N, GP -> B] = LAMBDA (bk:B), (nm: N), (pn:GP):
        IF Known? (bk,nm) THEN bk ELSE bk WITH [(nm):=pn] ENDIF

 ChangePhone: [B, N, GP -> B] = LAMBDA (bk:B), (nm: N), (pn:GP):
        IF Known? (bk,nm) THEN bk WITH [(nm):=pn] ELSE bk ENDIF

 DellPhone: [B, N -> B] = LAMBDA (bk:B), (nm: N):
        bk WITH [(nm):= nonumber]


  FindAdd: CONJECTURE FORALL (bk:B), (nm:N), (pn:GP):
      NOT Known?(bk,nm) IMPLIES FindPhone(AddPhone(bk,nm,pn), nm) = pn

  FindChange: CONJECTURE FORALL (bk:B), (nm:N), (pn:GP):
      Known?(bk,nm) IMPLIES FindPhone(ChangePhone(bk,nm,pn), nm) = pn

  AddChange: CONJECTURE FORALL (bk:B), (nm:N), (pn0,pn1:GP):
      ChangePhone(AddPhone(bk,nm,pn0), nm, pn1) =
      AddPhone(ChangePhone(bk,nm,pn1), nm, pn1)


END phone3
```

```
phone4: THEORY

BEGIN

 N: TYPE
 P: TYPE
 B: TYPE = [N->setof[P]]


 nonumber: P
 emptybook: B = LAMBDA(nm:N): emptyset[P]

 FindPhone: [B,N -> setof[P]] = LAMBDA (bk:B), (nm:N): bk(nm)

 AddPhone: [B, N, P -> B] = LAMBDA (bk:B), (nm: N), (pn:P):
        bk WITH [(nm):=add(pn,bk(nm))]

 DellPhone: [B, N, P -> B] = LAMBDA (bk:B), (nm: N), (pn:P):
        bk WITH [(nm):= emptyset[P]]

 DellPhoneNum: [B, N, P -> B] = LAMBDA (bk:B), (nm: N), (pn:P):
        bk WITH [(nm):= remove(pn,bk(nm))]


  FindAdd: CONJECTURE FORALL (bk:B), (nm:N), (pn:P):
      member(pn, FindPhone(AddPhone(bk,nm,pn), nm))

  DellAdd: CONJECTURE FORALL (bk:B), (nm:N), (pn:P):
      DellPhoneNum(AddPhone(bk,nm,pn), nm, pn) =
        DellPhoneNum(bk,nm,pn)



END phone4
```