points are closed for T1 topologies

Johan G. F. Belinfante
2003 July 3

```
In[1]:= << goedel52.s33; << tools.m

        :Package Title: goedel52.s33       2003 July 2 at 4:55 p.m.

        It is now:  2003 Jul 6 at 23:2

        Loading Simplification Rules

        TOOLS.M                      Revised 2003 July 3

        weightlimit = 40
```

■ introduction

This notebook contains a proof that "points are closed" for a **T1** topology. Actually, the result proved is more general in that the topology axioms are not assumed to hold for the collection of sets **t**. The final result is expressed concisely without variables.

■ points are closed for a T1 topology

Even though the topology axioms are not assumed to hold, it is easiest to motivate the proof by reference to the case of a topology. Suppose that **p** is a point of a topological space with a topology **t** that satisfies the **T1** condition. The inverse of the **T1** condition is used as follows:

```
In[2]:= SubstTest[implies,
         subclass[u, v], subclass[image[inverse[u], w], image[inverse[v], w]],
         {u -> composite[Di, id[U[t]]], v -> composite[complement[inverse[E]], id[t], E],
          w -> singleton[p]}]

Out[2]= or[not[member[p, V]], not[subclass[composite[Di, id[U[t]]],
            composite[complement[inverse[E]], id[t], E]]], subclass[U[t],
           union[singleton[p], U[intersection[t, P[complement[singleton[p]]]]]]]] ==
         True
```

This result is added as a temporary simplification rule.

```
In[3]:= (% /. {p -> p_, t -> t_}) /. Equal -> SetDelayed
```

Lemma:

```
In[4]:= equal[intersection[t, P[intersection[complement[singleton[p]], U[t]]]],
          intersection[t, P[complement[singleton[p]]]]] //
         AssertTest

Out[4]= equal[intersection[t, P[complement[singleton[p]]]],
          intersection[t, P[intersection[complement[singleton[p]], U[t]]]]] ==
         True

In[5]:= intersection[t_, P[intersection[complement[singleton[p_]], U[t_]]]] :=
         intersection[t, P[complement[singleton[p]]]]
```

Lemma:

```
In[6]:= equal[intersection[complement[singleton[p]], U[t]],
          U[intersection[t, P[complement[singleton[p]]]]]] //
         AssertTest

Out[6]= equal[intersection[complement[singleton[p]], U[t]],
          U[intersection[t, P[complement[singleton[p]]]]]] == subclass[U[t],
          union[singleton[p], U[intersection[t, P[complement[singleton[p]]]]]]]

In[7]:= Map[implies[Last[%], #]&, %]

Out[7]= or[equal[intersection[complement[singleton[p]], U[t]],
            U[intersection[t, P[complement[singleton[p]]]]], not[subclass[U[t],
             union[singleton[p], U[intersection[t, P[complement[singleton[p]]]]]]]]] ==
         True

In[8]:= (% /. {p -> p_, t -> t_}) /. Equal -> SetDelayed
```

A key fact that will be used is that a set is open if it is its own interior. This fact is needed only for the (relative) complement of the singleton of a point **p**. This key fact is actually a theorem about **Uclosure**. If **t** is a topology, then **Uclosure[t] = t**, so the **Uclosure** operation would not need explicit mention in this particular case.

```
In[9]:= SubstTest[implies,
         and[member[x, V], equal[x, U[intersection[t, P[x]]]], member[x, Uclosure[t]],
         x -> dif[U[t], singleton[p]]]

Out[9]= or[member[intersection[complement[singleton[p]], U[t]], Uclosure[t]],
          not[equal[intersection[complement[singleton[p]], U[t]],
            U[intersection[t, P[complement[singleton[p]]]]]],
          not[member[intersection[complement[singleton[p]], U[t]], V]]] ==
         True

In[10]:= (% /. {p -> p_, t -> t_}) /. Equal -> SetDelayed
```

Lemma: (This is worth making permanent, so some care is taken to make it as general as possible.)

```
In[11]:= Map[or[not[member[y, z]], #]&,
          SubstTest[implies, member[z, V], member[intersection[z, x], V], z -> U[y]]]

Out[11]= or[member[intersection[x, U[y]], V], not[member[y, z]]] == True

In[12]:= or[member[intersection[x_, U[y_]], V], not[member[y_, z_]]] := True
```

The facts deduced can be combined:

```
In[13]:= Map[not, SubstTest[and, implies[p0, p1], implies[and[p1, p2], p3],
           implies[p3, p4], implies[p5, p6], implies[and[p4, p6], p7],
           not[implies[and[p0, p2, p5], p7]],
          { p0 -> member[p, U[t]], p1 -> member[p, V], p2 ->
             subclass[composite[Di, id[U[t]]], composite[complement[inverse[E]], id[t], E]],
            p3 -> subclass[U[t],
              union[singleton[p], U[intersection[t, P[complement[singleton[p]]]]]]],
            p4 -> equal[intersection[complement[singleton[p]], U[t]],
              U[intersection[t, P[complement[singleton[p]]]]]],
            p5 -> member[t, V],
            p6 -> member[intersection[complement[singleton[p]], U[t]], V],
            p7 -> member[intersection[complement[singleton[p]], U[t]], Uclosure[t]]}]]

Out[13]= or[member[intersection[complement[singleton[p]], U[t]], Uclosure[t]],
           not[member[p, U[t]]], not[member[t, V]], not[subclass[
             composite[Di, id[U[t]]], composite[complement[inverse[E]], id[t], E]]] ==
          True

In[14]:= (% /. {p -> p_, t -> t_}) /. Equal -> SetDelayed
```

Thus, points are closed:

```
In[15]:= implies[and[member[p, U[t]], member[t, T1]],
          member[dif[U[t], singleton[p]], Uclosure[t]]]

Out[15]= True
```

This can also be written as follows:

```
In[16]:= implies[and[member[p, U[t]], member[t, T1]],
          member[singleton[p], image[RC[U[t]], Uclosure[t]]]]

Out[16]= True
```

■ variable-free version

The first step is to remove the variable **p** from the theorem:

```
In[17]:= Map[equal[V, #]&, SubstTest[class, p,
            implies[and[member[p, U[t]], member[t, x]], member[singleton[p], y]],
           {x -> T1, y -> image[RC[U[t]], Uclosure[t]]}]] // Reverse

Out[17]= or[not[member[t, V]], not[
             subclass[composite[Di, id[U[t]]], composite[complement[inverse[E]], id[t], E]]],
            subclass[image[SINGLETON, U[t]], image[RC[U[t]], Uclosure[t]]]] ==
           True

In[18]:= (% /. {p -> p_, t -> t_}) /. Equal -> SetDelayed
```

To derive a variable-free statement of the theorem, one needs to introduce a definition for the class of collections of sets that satisfy the condition that points are closed:

```
In[19]:= member[t_, PointClosed] :=
          and[member[t, V], subclass[image[SINGLETON, U[t]], image[RC[U[t]], t]]]
```

The theorem can now be written as follows:

```
In[20]:= implies[member[t, T1], member[Uclosure[t], PointClosed]]

Out[20]= True
```

The final step is to eliminate the variable **t**.

```
In[21]:= Map[equal[V, #]&, SubstTest[class, t, implies[member[t, x], member[Uclosure[t], y]],
           {x -> T1, y -> PointClosed}]] // Reverse

Out[21]= subclass[image[UCLOSURE, T1], PointClosed] == True
```

This is the variable-free formulation of the theorem.

```
In[22]:= subclass[image[UCLOSURE, T1], PointClosed] := True
```

■ Normalization for PointClosed

The **Normality** test is used in the section to obtain an equation for the class **PointClosed** that would be suitable for use as a definition if one wishes to reproduce these theorems using an automated reasoning program such as **Otter**. The formula is improved by the following rewrite rule:

```
In[23]:= fix[composite[inverse[POWER], S, IMAGE[SINGLETON]]] // Renormality

Out[23]= fix[composite[inverse[POWER], S, IMAGE[SINGLETON]]] == V

In[24]:= fix[composite[inverse[POWER], S, IMAGE[SINGLETON]]] := V

In[25]:= Map[complement, PointClosed // Normality // Reverse]

Out[25]= fix[composite[complement[composite[intersection[
               composite[E, DIF], composite[inverse[BIGCUP], FIRST]], inverse[SECOND]]],
             inverse[E], IMAGE[SINGLETON], BIGCUP]] ==
           complement[PointClosed]

In[26]:= fix[composite[complement[composite[intersection[
               composite[E, DIF], composite[inverse[BIGCUP], FIRST]], inverse[SECOND]]],
             inverse[E], IMAGE[SINGLETON], BIGCUP]] :=
           complement[PointClosed]
```