relative topology

Johan G. F. Belinfante
2003 March 16

```
<< goedel52.r38; << tools.m

:Package Title: goedel52.r38          2003 March 16 at 10:50 a.m.

It is now:  2003 Mar 16 at 14:26

Loading Simplification Rules

TOOLS.M                       Revised 2002 December 27

weightlimit = 40
```

■ introduction: subspace topology (relative topology)

Let **t** be a topology, and let **x** be a subspace of **U[t]**. The subspace topology for **x** is the collection of relatively open sets:

```
class[u, exists[v, and[member[v, t], equal[u, intersection[x, v]]]]]

image[IMAGE[id[x]], t]
```

Our goal is to show that this is always a topology:

```
(assert[forall[t, implies[member[t, z], member[image[IMAGE[id[x]], t], z]]]]) /. z → TOPS

subclass[image[IMAGE[IMAGE[id[x]]], TOPS], TOPS]
```

That is, the class **TOPS** of all topologies is invariant under the relativization operation **IMAGE[IMAGE[id[x]]]**.

```
invariant[IMAGE[IMAGE[id[x]]], TOPS]

subclass[image[IMAGE[IMAGE[id[x]]], TOPS], TOPS]
```

The strategy is to deduce this fact from the following facts which have been proved previously:

```
intersection[CAPclosed, fix[UCLOSURE]]

TOPS

invariant[IMAGE[IMAGE[id[x]]], CAPclosed]

True

invariant[IMAGE[IMAGE[id[x]]], fix[UCLOSURE]]

True
```

■ derivation

The intersection of two invariant classes is invariant:

```
SubstTest[implies, invariant[u, v], subclass[image[u, intersection[v, w]], v],
 {u → IMAGE[IMAGE[id[x]]], v → CAPclosed, w → fix[UCLOSURE]}]

subclass[image[IMAGE[IMAGE[id[x]]], TOPS], CAPclosed] == True

subclass[image[IMAGE[IMAGE[id[x_]]], TOPS], CAPclosed] := True

SubstTest[implies, invariant[u, v], subclass[image[u, intersection[v, w]], v],
 {u → IMAGE[IMAGE[id[x]]], v → fix[UCLOSURE], w → CAPclosed}]

subclass[image[IMAGE[IMAGE[id[x]]], TOPS], fix[UCLOSURE]] == True

subclass[image[IMAGE[IMAGE[id[x_]]], TOPS], fix[UCLOSURE]] := True

SubstTest[subclass, u, intersection[v, w],
 {u -> image[IMAGE[IMAGE[id[x]]], TOPS], v -> CAPclosed, w → fix[UCLOSURE]}]

subclass[image[IMAGE[IMAGE[id[x]]], TOPS], TOPS] == True

subclass[image[IMAGE[IMAGE[id[x_]]], TOPS], TOPS] := True
```

■ opposite inclusion

The inclusion derived in the proceding section can be replaced by an equation. To do so, one needs to derive an inclusion going in the opposite direction:

```
Map[range, Assoc[IMAGE[IMAGE[id[x]]], id[P[P[x]]], id[TOPS]]]

image[IMAGE[IMAGE[id[x]]], intersection[TOPS, P[P[x]]]] == intersection[TOPS, P[P[x]]]

image[IMAGE[IMAGE[id[x_]]], intersection[TOPS, P[P[x_]]]] := intersection[TOPS, P[P[x]]]

SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
 {u → intersection[TOPS, P[P[x]]], v → TOPS, w → IMAGE[IMAGE[id[x]]]}]

subclass[intersection[TOPS, P[P[x]]], image[IMAGE[IMAGE[id[x]]], TOPS]] == True

subclass[intersection[TOPS, P[P[x_]]], image[IMAGE[IMAGE[id[x_]]], TOPS]] := True

SubstTest[and, subclass[u, v], subclass[v, u],
  {u -> intersection[TOPS, P[P[x]]], v -> image[IMAGE[IMAGE[id[x]]], TOPS]}] // Reverse

equal[image[IMAGE[IMAGE[id[x]]], TOPS], intersection[TOPS, P[P[x]]]] == True

image[IMAGE[IMAGE[id[x_]]], TOPS] := intersection[TOPS, P[P[x]]]
```

■ a corollary

```
SubstTest[U, image[IMAGE[IMAGE[id[x]]], z], z -> TOPS]

U[intersection[TOPS, P[P[x]]]] == P[x]

Map[VERTSECT[reify[x, #]] &, %]

composite[CORE[TOPS], POWER] == POWER

composite[CORE[TOPS], POWER] := POWER
```