topologies characterized via closed sets

Johan G. F. Belinfante
2011 February 7

```
In[1]:= SetDirectory["l:"]; << goedel.11feb04a

        :Package Title: goedel.11feb04a          2011 February 4 at 1:50 p.m.

        It is now:  2011 Feb 7 at 13:37

        Loading Simplification Rules

        TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

        weightlimit = 40
```

summary

A topology can be characterized by its class of closed sets. This is the topic of Theorem 4 on page 40 of the following reference. (Kelley dismisses it as too trivial to be given a formal proof.)

```
In[2]:= "John L. Kelley, General Topology, D. Van Nostrand Co., Inc., Princeton, 1955.";
```

Kelley's Theorem 4 says that if **x** satisfies the four conditions **Aclosure[x] = x**, **image[CUP, cart[x, x]] = x**, **U[x] ∈ x** and **0 ∈ x**, then **x** is the class of closed sets for the topology **t = image[RC[U[x]], x]**.

the function IRC = λx. image[RC[U[x]], x]

Variable-free statements about classes of closed sets can be derived by introducing the function **λx. image[RC[U[x]], x]** for which the acronym **IRC** is introduced, standing for "**i**mage under **r**elative **c**omplements."

```
In[5]:= composite[IMG, id[composite[inverse[BIGCUP], inverse[RCF]]], inverse[SECOND]] := IRC
```

Theorem. **IRC** is a function.

```
In[6]:= SubstTest[FUNCTION, composite[funpart[t],
           id[composite[inverse[BIGCUP], inverse[RCF]]], inverse[SECOND]], t → IMG] // Reverse

Out[6]= FUNCTION[IRC] == True

In[7]:= FUNCTION[IRC] := True
```

Theorem. The domain of **IRC** is the class of all sets.

```
In[8]:= IminComp[IMG,
          composite[id[composite[inverse[BIGCUP], inverse[RCF]]], inverse[SECOND]], V]

Out[8]= domain[IRC] == V

In[9]:= domain[IRC] := V
```

Theorem. An **APPLY** rule for **IRC**.

```
In[10]:= ApComp[IMG, composite[
            id[composite[inverse[BIGCUP], inverse[RCF]]], inverse[SECOND]], x] // Reverse

Out[10]= APPLY[IRC, x] == union[complement[image[V, set[x]]], image[RC[U[x]], x]]

In[11]:= APPLY[IRC, x_] := union[complement[image[V, set[x]]], image[RC[U[x]], x]]
```

Theorem. A vertical section rule.

```
In[12]:= ImageComp[IMG,
          composite[id[composite[inverse[BIGCUP], inverse[RCF]]], inverse[SECOND]], set[x]]

Out[12]= image[IRC, set[x]] == intersection[image[V, set[x]], set[image[RC[U[x]], x]]]

In[13]:= image[IRC, set[x_]] := intersection[image[V, set[x]], set[image[RC[U[x]], x]]]
```

Theorem. Inverse image rule.

```
In[14]:= (member[x, image[inverse[funpart[t]], y]] // AssertTest) /. t → IRC

Out[14]= member[x, image[inverse[IRC], y]] == and[member[x, V], member[image[RC[U[x]], x], y]]

In[15]:= member[x_, image[inverse[IRC], y_]] := and[member[x, V], member[image[RC[U[x]], x], y]]
```

introduction

A **topology** is a set **t** of sets which is closed under binary intersections and arbitrary unions. The class of all topologies is called **TOPS**. By definition, the collection of closed sets of a topology **t** is **image[RC[U[t]], t]**. The topology itself can be recovered from the class of closed sets by the same operation. This result is currently recognized by the **GOEDEL** program when the **top[t]** wrapper is used for a topology. One can remove the wrapper as follows.

Theorem.

```
In[3]:= SubstTest[implies, and[equal[x, top[t]], equal[y, image[RC[U[x]], x]]],
          equal[x, image[RC[U[y]], y]], t → x] // Reverse

Out[3]= or[equal[x, image[RC[U[y]], y]],
          not[equal[y, image[RC[U[x]], x]]], not[member[x, TOPS]]] == True

In[4]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

A variable-free formulation of this result can be derived as follows.

Theorem. **id[TOPS] ∘ inverse[IRC] ⊂ IRC**.

```
In[16]:= Map[empty[range[composite[Id, complement[#]]]] &, SubstTest[class, pair[x, y],
          implies[and[member[setpart[x], t], equal[setpart[y], APPLY[z, setpart[x]]]],
           equal[setpart[x], APPLY[z, setpart[y]]]], {t → TOPS, z → IRC}]]

Out[16]= subclass[composite[id[TOPS], inverse[IRC]], IRC] == True

In[17]:= % /. Equal → SetDelayed
```

Corollary. **id[TOPS] ∘ inverse[IRC]** is a function.

```
In[18]:= SubstTest[implies, and[subclass[u, v], FUNCTION[v]],
          FUNCTION[u], {u → composite[id[TOPS], inverse[IRC]], v → IRC}] // Reverse

Out[18]= FUNCTION[composite[id[TOPS], inverse[IRC]]] == True

In[19]:= FUNCTION[composite[id[TOPS], inverse[IRC]]] := True
```

a preliminary result

One of the hypotheses in Kelley's Theorem 4 is the class **x** of closed sets of a topology is closed under finite unions. This condition is rewritten by the **GOEDEL** program as follows.

```
In[20]:= equal[image[BIGCUP, intersection[FINITE, P[x]]], x]

Out[20]= and[member[0, x], subclass[image[CUP, cart[x, x]], x]]
```

In this section only the first part of Kelley's Theorem 4 is considered. This part says that if **x** satisfies the four conditions **Aclosure[x] = x**, **image[CUP, cart[x, x]] = x**, **U[x] ∈ x** and **0 ∈ x**, then **t = image[RC[U[x]], x]** is a topology. For this part, the condition **0 ∈ x** is not needed. Three lemmas are needed.

The first lemma deals with the issue that **Aclosure[x]** and **fix[HULL[x]]** only agree when **x** is a set.

Lemma.

```
In[21]:= Map[not, SubstTest[and, implies[and[p2, p3], p4],
           not[implies[and[p1, p2], p4]], {p1 → member[U[x], x], p2 → equal[x, Aclosure[x]],
            p3 → member[x, V], p4 → equal[x, fix[HULL[x]]]}]] // Reverse

Out[21]= or[equal[x, fix[HULL[x]]], not[equal[x, Aclosure[x]]], not[member[U[x], x]]] == True

In[22]:= or[equal[x_, fix[HULL[x_]]],
          not[equal[x_, Aclosure[x_]]], not[member[U[x_], x_]]] := True
```

The second lemma just applies the first lemma to the case at hand.

Lemma.

```
In[23]:= Map[not, SubstTest[and, implies[p1, p2], not[implies[p1, p3]],
           {p1 → and[member[U[x], x], equal[x, Aclosure[x]]], p2 -> equal[x, fix[HULL[x]]],
            p3 → member[image[RC[U[x]], x], fix[UCLOSURE]]}]] // Reverse

Out[23]= or[equal[image[RC[U[x]], x], union[image[RC[U[x]], fix[HULL[x]]], set[0]]],
          not[equal[x, Aclosure[x]]], not[member[U[x], x]]] == True

In[24]:= (% /. x → x_) /. Equal → SetDelayed
```

The third lemma rewrites relative complements of intersections of relative complements as unions.

Lemma.

```
In[25]:= Map[subclass[image[#, cart[x, x]], x] &, Assoc[RC[U[x]], RC[U[x]], CUP]]

Out[25]= subclass[image[RC[U[x]], image[CAP, cart[image[RC[U[x]], x], image[RC[U[x]], x]]]],
          x] == or[not[member[x, V]], subclass[image[CUP, cart[x, x]], x]]

In[26]:= subclass[image[RC[U[x_]],
           image[CAP, cart[image[RC[U[x_]], x_], image[RC[U[x_]], x_]]]], x_] :=
         or[not[member[x, V]], subclass[image[CUP, cart[x, x]], x]]
```

Theorem. If **x** is a collection of sets that is binary closed under unions, closed under arbitrary intersections, and holds its sum class, then the set of relative complements of members of **x** in **U[x]** is a topology.

```
In[27]:= Map[
          implies[and[equal[x, Aclosure[x]], member[U[x], x], equal[x, image[CUP, cart[x, x]]]],
            #] &, SubstTest[member, t, intersection[u, v],
          {t → image[RC[U[x]], x], u → fix[UCLOSURE], v → binclosed[CAP]}]] // Reverse

Out[27]= or[member[image[RC[U[x]], x], TOPS], not[equal[x, Aclosure[x]]],
          not[equal[x, image[CUP, cart[x, x]]]], not[member[U[x], x]]] == True

In[28]:= or[member[image[RC[U[x_]], x_], TOPS], not[equal[Aclosure[x_], x_]],
          not[equal[image[CUP, cart[x_, x_]], x_]], not[member[U[x_], x_]]] := True
```

A variable-free formulation of this result can be derived.

Theorem.

```
In[29]:= Map[equal[V, #] &,
         SubstTest[class, x, or[member[x, t], not[equal[Aclosure[x], x]], not[member[U[x], x]],
           not[equal[image[CUP, cart[x, x]], x]]], t -> image[inverse[IRC], TOPS]]]

Out[29]= subclass[image[IRC, intersection[binclosed[CUP], fix[ACLOSURE],
            fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]]], TOPS] == True

In[30]:= subclass[image[IRC, intersection[binclosed[CUP], fix[ACLOSURE],
            fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]]], TOPS] := True
```

a counter-example

The three conditions **Aclosure[x] = x**, **image[CUP, cart[x, x]] = x** and **U[x] ∈ x** do not suffice to characterize the class of closed sets for a topology. For any topology, the class of all closed sets holds the empty set.

```
In[31]:= implies[member[t, TOPS], member[0, image[RC[U[t]], t]]]

Out[31]= True
```

The three conditions on **x** in the theorem in the preceding section do not imply **0 ∈ x**.

Counterexample. The three conditions on **x** in the theorem in the preceding section do not imply **0 ∈ x**.

```
In[32]:= implies[and[equal[Aclosure[x], x], equal[image[CUP, cart[x , x]], x], member[U[x], x]],
           member[0, x]] /. x → set[set[0]]

Out[32]= False
```

Kelley's Theorem 4 says that if **x** satisfies all four conditions **Aclosure[x] = x**, **image[CUP, cart[x, x]] = x**, **U[x] ∈ x** and **0 ∈ x**, then **x** is the class of closed sets for the topology **t = image[RC[U[x]], x]**. Eliminating the variable **t** from the conclusion yields the following messy statement.

```
In[33]:= equal[x, image[RC[U[t], t]]] /. t -> image[RC[U[x]], x]

Out[33]= equal[x,
          image[RC[intersection[complement[A[x]], image[V, set[x]], U[x]], image[RC[U[x]], x]]]]
```

Observe that since **x** is a set, **image[V, set[x]] = V**.

```
In[34]:= implies[member[U[x], x], equal[image[V, set[x]], V]]

Out[34]= True
```

Observe further that since **0 ∈ x**, one has **A[x] = 0**.

```
In[35]:= implies[member[0, x], equal[0, A[x]]]

Out[35]= True
```

With these simplifications, the conclusion reduces to the statement that **x** is a set.

```
In[36]:= (equal[x, image[RC[U[t]], t]] /. t -> image[RC[U[x]], x]) /.
          {image[V, set[x]] → V, A[x] → 0}

Out[36]= member[x, V]
```

Although this does informally prove Kelley's Theorem 4, it is not provide a convenient formal statement of the theorem. This defect can be overcome by using the function **IRC**.

image[IRC, TOPS]

A set **x** belongs to the class **image[IRC, TOPS]** if it is the class of closed sets for some topology. One can use **reify** to quickly derive variable-free formulations of standard results about closed sets.

Theorem. The collection of closed sets for any topology is closed under arbitrary intersections.

```
In[37]:= Map[empty,
          SubstTest[reify, x, dif[set[top[x]], t], t -> image[inverse[IRC], fix[ACLOSURE]]]]

Out[37]= subclass[image[IRC, TOPS], fix[ACLOSURE]] == True

In[38]:= subclass[image[IRC, TOPS], fix[ACLOSURE]] := True
```

Corollary. Closed sets are closed under arbitrary intersections.

```
In[39]:= SubstTest[implies, and[member[x, u], subclass[u, v]],
           member[x, v], {u -> image[IRC, TOPS], v -> fix[ACLOSURE]}] // Reverse

Out[39]= or[equal[x, Aclosure[x]], not[member[x, image[IRC, TOPS]]]] == True

In[40]:= or[equal[x_, Aclosure[x_]], not[member[x_, image[IRC, TOPS]]]] := True
```

Theorem. The collection of closed sets for any topology is closed under binary unions.

```
In[41]:= Map[empty,
          SubstTest[reify, x, dif[set[top[x]], t], t -> image[inverse[IRC], binclosed[CUP]]]]

Out[41]= subclass[image[IRC, TOPS], binclosed[CUP]] == True

In[42]:= subclass[image[IRC, TOPS], binclosed[CUP]] := True
```

Corollary. The class of all closed sets of any topology is closed under binary unions.

```
In[43]:= SubstTest[implies, and[member[x, u], subclass[u, v]],
           member[x, v], {u -> image[IRC, TOPS], v → binclosed[CUP]}] // Reverse

Out[43]= or[not[member[x, image[IRC, TOPS]]], subclass[image[CUP, cart[x, x]], x]] == True

In[44]:= or[not[member[x_, image[IRC, TOPS]]], subclass[image[CUP, cart[x_, x_]], x_]] := True
```

Theorem. The collection of closed sets for any topology holds its sum class.

```
In[45]:= Map[empty, SubstTest[reify, x, dif[set[top[x]], t],
           t -> image[inverse[IRC], fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]]]]

Out[45]= subclass[image[IRC, TOPS], fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]] == True

In[46]:= subclass[image[IRC, TOPS], fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]] := True
```

Corollary. The union of all closed sets is closed.

```
In[47]:= SubstTest[implies, and[member[x, u], subclass[u, v]],
           member[x, v], {u -> image[IRC, TOPS],
            v → fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]}] // Reverse

Out[47]= or[member[U[x], x], not[member[x, image[IRC, TOPS]]]] == True

In[48]:= or[member[U[x_], x_], not[member[x_, image[IRC, TOPS]]]] := True
```

Theorem. For any topology, the collection of closed sets holds the empty set.

```
In[49]:= Map[empty, SubstTest[reify, x, dif[set[top[x]], t],
           t -> image[inverse[IRC], complement[P[complement[set[0]]]]]]]

Out[49]= member[0, A[image[IRC, TOPS]]] == True

In[50]:= member[0, A[image[IRC, TOPS]]] := True
```

Corollary. The empty set is closed for any topology.

```
In[51]:= SubstTest[implies, and[member[x, u], subclass[u, v]], member[x, v],
           {u -> image[IRC, TOPS], v → complement[P[complement[set[0]]]]}] // Reverse

Out[51]= or[member[0, x], not[member[x, image[IRC, TOPS]]]] == True

In[52]:= or[member[0, x_], not[member[x_, image[IRC, TOPS]]]] := True
```

results involving ADJOIN[{0}]

Although the function **IRC** is not an involution, the following result shows that it comes close.

Lemma.

```
In[53]:= composite[IRC, IRC, ADJOIN[set[0]]] // FastReifNormality

Out[53]= composite[IRC, IRC, ADJOIN[set[0]]] == ADJOIN[set[0]]

In[54]:= composite[IRC, IRC, ADJOIN[set[0]]] := ADJOIN[set[0]]
```

Theorem.

```
In[55]:= Assoc[composite[IRC, IRC], ADJOIN[set[0]], id[fix[ADJOIN[set[0]]]]]

Out[55]= composite[IRC, IRC, id[complement[P[complement[set[0]]]]]] ==
          id[complement[P[complement[set[0]]]]]

In[56]:= composite[IRC, IRC, id[complement[P[complement[set[0]]]]]] :=
          id[complement[P[complement[set[0]]]]]
```

Corollary. (A generalization of a result derived earlier.)

```
In[57]:= SubstTest[implies, subclass[composite[x, y], Id],
           FUNCTION[composite[inverse[y], id[domain[x]]]],
           {x → IRC, y → composite[IRC, id[complement[P[complement[set[0]]]]]]}] // Reverse

Out[57]= FUNCTION[composite[id[complement[P[complement[set[0]]]]], inverse[IRC]]] == True

In[58]:= FUNCTION[composite[id[complement[P[complement[set[0]]]]], inverse[IRC]]] := True
```

formal statement of Kelley's Theorem 4

Lemma. A special case of an earlier result, adding back in the redundant condition **0 ∈ x**.

```
In[60]:= SubstTest[implies, and[subclass[x, y], subclass[y, z]], subclass[x, z],
           {x -> intersection[binclosed[CUP], complement[P[complement[set[0]]]],
              fix[ACLOSURE], fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]],
            y -> intersection[binclosed[CUP], fix[ACLOSURE],
              fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]],
            z → image[inverse[IRC], TOPS]}] // Reverse

Out[60]= subclass[image[IRC, intersection[binclosed[CUP], complement[P[complement[set[0]]]],
             fix[ACLOSURE], fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]]], TOPS] == True

In[61]:= subclass[image[IRC, intersection[binclosed[CUP], complement[P[complement[set[0]]]],
             fix[ACLOSURE], fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]]], TOPS] := True
```

Lemma. A simplification rule.

```
In[62]:= ImageComp[IRC, composite[IRC, id[fix[ADJOIN[set[0]]]]], x] // Reverse

Out[62]= image[IRC, image[IRC, intersection[x, complement[P[complement[set[0]]]]]]] ==
          intersection[x, complement[P[complement[set[0]]]]]

In[63]:= image[IRC, image[IRC, intersection[x_, complement[P[complement[set[0]]]]]]] :=
          intersection[x, complement[P[complement[set[0]]]]]
```

Lemma.

```
In[64]:= SubstTest[implies, subclass[x, y],
           subclass[image[t, x], image[t, y]], {t → IRC, x -> image[IRC,
              intersection[binclosed[CUP], complement[P[complement[set[0]]]], fix[ACLOSURE],
               fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]]], y → TOPS}] // Reverse

Out[64]= subclass[intersection[binclosed[CUP],
            fix[ACLOSURE], fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]],
           union[image[IRC, TOPS], P[complement[set[0]]]]] == True

In[65]:= % /. Equal → SetDelayed
```

Theorem. Variable free statement of Kelley's Theorem 4.

```
In[66]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u → intersection[binclosed[CUP], complement[P[complement[set[0]]]], fix[ACLOSURE],
             fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]], v → image[IRC, TOPS]}]

Out[66]= equal[image[IRC, TOPS],
           intersection[binclosed[CUP], complement[P[complement[set[0]]]],
            fix[ACLOSURE], fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]]] == True

In[67]:= intersection[binclosed[CUP], complement[P[complement[set[0]]]], fix[ACLOSURE],
           fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]] := image[IRC, TOPS]
```

Corollary. Reintroducing variables into Kelley's Theorem 4.

```
In[68]:= Map[implies[#, member[x, image[IRC, TOPS]]] &,
          SubstTest[member, x, intersection[t, u, v, w],
           {t -> binclosed[CUP], u -> complement[P[complement[set[0]]]], v -> fix[ACLOSURE],
            w -> fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]}]]

Out[68]= or[member[x, image[IRC, TOPS]], not[equal[x, Aclosure[x]]], not[member[0, x]],
           not[member[U[x], x]], not[subclass[image[CUP, cart[x, x]], x]]] == True

In[69]:= or[member[x_, image[IRC, TOPS]], not[equal[x_, Aclosure[x_]]], not[member[0, x_]],
           not[member[U[x_], x_]], not[subclass[image[CUP, cart[x_, x_]], x_]]] := True
```