# closed under finite unions

Johan G. F. Belinfante
2008 January 10

In[1]:= `SetDirectory["l:"]; << goedel.08jan06a; << tools.m`

```
:Package Title: goedel.08jan06a            2008 January 6 at 10:55 p.m.

It is now:  2008 Jan 10 at 15:29

Loading Simplification Rules

TOOLS.M                                    Revised 2008 January 2

weightlimit = 40
```

## summary

In this notebook it is proved that a class is closed under finite unions if and only if it holds the empty set and is closed under binary unions. The main idea of the derivation is to use **FINITE** induction:

In[2]:= `implies[and[member[0, x], invariant[K, x]], subclass[FINITE, x]]`

Out[2]= True

## derivation

The following lemma will be used to simplify the expression that arises when **FINITE** induction is applied.

In[3]:=
```
SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
  {t → composite[inverse[E], SECOND, id[cart[P[x], V]]],
   u → inverse[K], v → inverse[S]}] // Reverse
```

Out[3]= `subclass[U[image[inverse[K], P[x]]], x] == True`

In[4]:= `subclass[U[image[inverse[K], P[x_]]], x_] := True`

Corollary.

In[5]:= `equal[intersection[image[K, complement[P[x]]], P[x]], 0]`

Out[5]= True

In[6]:= `intersection[image[K, complement[P[x_]]], P[x_]] := 0`

Another simplification rule is needed as well:

```
In[7]:= or[not[member[0, x]], not[subclass[image[CUP, cart[x, x]], x]],
          subclass[image[CUP, cart[x, union[x, set[0]]]], x]] // AssertTest

Out[7]= or[not[member[0, x]], not[subclass[image[CUP, cart[x, x]], x]],
          subclass[image[CUP, cart[x, union[x, set[0]]]], x]] == True

In[8]:= (% /. x → x_) /. Equal → SetDelayed
```

An image of the cover relation **K** is rewritten as an image of the function **CUP** using the following lemma.

```
In[9]:= Map[subclass[image[composite[#,
             composite[id[cart[V, union[set[0], range[SINGLETON]]]], inverse[FIRST]]],
            image[inverse[BIGCUP], x]], x] &, Assoc[BIGCUP, CUP, id[cart[P[x], P[x]]]]]

Out[9]= subclass[image[BIGCUP, intersection[image[K, image[inverse[BIGCUP], x]], P[x]]], x] ==
         subclass[image[CUP, cart[x, union[x, set[0]]]], x]

In[10]:= subclass[image[BIGCUP, intersection[image[K, image[inverse[BIGCUP], x_]], P[x_]]],
           x_] := subclass[image[CUP, cart[x, union[x, set[0]]]], x]
```

An application of **FINITE** induction now yields this:

```
In[11]:= SubstTest[implies, and[member[0, u], invariant[K, u]], subclass[FINITE, u],
           u -> union[complement[P[x]], image[inverse[BIGCUP], x]]] // Reverse

Out[11]= or[not[member[0, x]], not[subclass[image[CUP, cart[x, union[x, set[0]]]], x]],
           subclass[image[BIGCUP, intersection[FINITE, P[x]]], x]] == True
```

Main theorem.

```
In[13]:= Map[not, SubstTest[and, implies[and[p1, p2], p3],
            implies[and[p1, p3], p4], not[implies[and[p1, p2], p4]],
            {p1 → member[0, x], p2 -> subclass[image[CUP, cart[x, x]], x],
             p3 -> subclass[image[CUP, cart[x, union[x, set[0]]]], x],
             p4 -> subclass[image[BIGCUP, intersection[FINITE, P[x]]], x]}]] // Reverse

Out[13]= or[not[member[0, x]], not[subclass[image[CUP, cart[x, x]], x]],
           subclass[image[BIGCUP, intersection[FINITE, P[x]]], x]] == True

In[14]:= (% /. x → x_) /. Equal → SetDelayed
```

The implication in the other direction also holds, and can be combined with the main theorem to obtain the following simple rewrite rule:

```
In[15]:= equiv[subclass[image[BIGCUP, intersection[FINITE, P[x]]], x],
          and[member[0, x], subclass[image[CUP, cart[x, x]], x]]]

Out[15]= True

In[16]:= subclass[image[BIGCUP, intersection[FINITE, P[x_]]], x_] :=
          and[member[0, x], subclass[image[CUP, cart[x, x]], x]]
```

Corollary. (Variable-free restatement of the main theorem.)

```
In[17]:= allclosed[composite[BIGCUP, id[FINITE]]] // Normality

Out[17]= allclosed[composite[BIGCUP, id[FINITE]]] ==
          intersection[binclosed[CUP], complement[P[complement[set[0]]]]]

In[18]:= allclosed[composite[BIGCUP, id[FINITE]]] :=
          intersection[binclosed[CUP], complement[P[complement[set[0]]]]]
```