# finite choice

*Johan G. F. Belinfante*
*2004 December 6*

```
In[1]:= SetDirectory["i:"]; << goedel64.05b; << tools.m

        :Package Title: goedel64.05b      2004 December 5 at 11:55 p.m.

        It is now:  2004 Dec 6 at 10:26

        Loading Simplification Rules

        TOOLS.M                         Revised 2004 November 17

        weightlimit = 40
```

## summary

The axiom of choice is not needed for finite sets because of the following theorem which is derived in this notebook: If **x** is a finite collection of sets, there is a function which selects a member from each non-empty set belonging to **x**. This theorem will be derived using a version of **FINITE** induction.

## derivation

The class to which **FINITE** induction will be applied is:

```
In[2]:= class[x, member[composite[inverse[E], id[x]], s]] /. s → SELECT

Out[2]= image[inverse[IMAGE[composite[id[inverse[E]], inverse[FIRST]]]], SELECT]
```

Note that the empty set belongs to this class:

```
In[3]:= member[0,
         image[inverse[IMAGE[composite[id[inverse[E]], inverse[FIRST]]]], SELECT]]

Out[3]= True
```

The same is true for every singleton:

```
In[4]:= member[singleton[x],
         image[inverse[IMAGE[composite[id[inverse[E]], inverse[FIRST]]]], SELECT]]

Out[4]= True
```

The variable x can be eliminated from this statement:

```
In[5]:= Map[equal[V, #] &,
          SubstTest[class, x, member[singleton[x], y], y -> image[inverse[
              IMAGE[composite[id[inverse[E]], inverse[FIRST]]]], SELECT]]] // Reverse

Out[5]= subclass[range[VERTSECT[composite[id[inverse[E]], inverse[FIRST]]]], SELECT] ==
         True

In[6]:= subclass[
          range[VERTSECT[composite[id[inverse[E]], inverse[FIRST]]]], SELECT] := True
```

The remaining step is to show that the class in question is closed under binary unions. This follows from this observation:

```
In[7]:= SubstTest[implies, equal[0, X[union[u, v]]], or[equal[0, X[u]], equal[0, X[v]]],
         {u → composite[inverse[E], id[x]], v → composite[inverse[E], id[y]]}]

Out[7]= or[equal[0, X[composite[inverse[E], id[x]]]],
          equal[0, X[composite[inverse[E], id[y]]]],
          not[equal[0, X[composite[inverse[E], id[union[x, y]]]]]]] == True

In[8]:= or[equal[0, X[composite[inverse[E], id[x_]]]],
          equal[0, X[composite[inverse[E], id[y_]]]],
          not[equal[0, X[composite[inverse[E], id[union[x_, y_]]]]]]] := True
```

The variables **x** and **y** can be eliminated as follows:

```
In[9]:= Map[equal[0, composite[Id, complement[#]]] &, SubstTest[class, pair[x, y],
           implies[and[member[x, z], member[y, z]], member[union[x, y], z]],
           z -> image[inverse[IMAGE[composite[id[inverse[E]], inverse[FIRST]]]],
             SELECT]]] // Reverse

Out[9]= subclass[
          image[IMAGE[composite[id[inverse[E]], inverse[FIRST]]], image[CUP, cart[
             image[inverse[IMAGE[composite[id[inverse[E]], inverse[FIRST]]]], SELECT],
             image[inverse[IMAGE[composite[id[inverse[E]], inverse[FIRST]]]],
              SELECT]]]], SELECT] == True

In[10]:= % /. Equal → SetDelayed
```

The final step is to use a version of **FINITE** induction:

```
In[11]:= SubstTest[implies, and[member[0, x], subclass[range[SINGLETON], x],
           subclass[image[CUP, cart[x, x]], x]], subclass[FINITE, x], x ->
           image[inverse[IMAGE[composite[id[inverse[E]], inverse[FIRST]]]], SELECT]]

Out[11]= subclass[image[IMAGE[composite[id[inverse[E]], inverse[FIRST]]], FINITE],
           SELECT] == True

In[12]:= subclass[image[IMAGE[composite[id[inverse[E]], inverse[FIRST]]], FINITE],
           SELECT] := True
```

This result becomes more transparent when one reintroduces a variable:

```
In[13]:= SubstTest[implies, and[member[x, y], subclass[y, z]],
          member[x, z], {y -> FINITE, z ->
            image[inverse[IMAGE[composite[id[inverse[E]], inverse[FIRST]]]], SELECT]}]

Out[13]= or[not[equal[0, X[composite[inverse[E], id[x]]]]],
           not[member[x, FINITE]]] == True

In[14]:= or[not[equal[0, X[composite[inverse[E], id[x_]]]]],
           not[member[x_, FINITE]]] := True
```

---

## comment

The following weaker statement was derived in an earlier notebook:

```
In[15]:= subclass[FINITE, SELECT]

Out[15]= True
```

From this result one can derive a special case of the theorem derived in this notebook. If **x** is a finite collection of finite sets, then **composite[inverse[E], id[x]]** is finite, and the fact **subclass[FINITE, SELECT]** would imply that the class of choice functions is not empty. The result obtained in this notebook is stronger in that it applies also to finite collections whose members may be infinite.