# finite total orders are well-orderings

Johan G. F. Belinfante
2005 October 17

```
In[1]:= SetDirectory["l:"]; << goedel74.17a; << tools.m

:Package Title: goedel74.17a            2005 October 17 at 12:25 noon

It is now:  2005 Oct 17 at 14:0

Loading Simplification Rules

TOOLS.M                        Revised 2005 October 17

weightlimit = 40
```

## summary

A finite total order relation and its inverse are both wellorderings. The derivation of these facts in this notebook uses the natural map to reduce the general case to the special case that the totat order relation is a restriction of the subset relation **S**. The relation **restrict[S, x, x]** is a total order if **x** is a clique of **union[S, inverse[S]]**.

```
In[2]:= member[restrict[S, x, x], TO]

Out[2]= and[member[x, V], subclass[cart[x, x], union[S, inverse[S]]]]
```

The relation **restrict[S, x, x]** is a wellordering if every nonempty subset of **x** has a least member.

```
In[3]:= member[restrict[S, x, x], WO]

Out[3]= and[member[x, V], subclass[P[x], union[fix[composite[E, BIGCAP]], set[0]]]]
```

The use of the compound wrapper **to[fin[x]]** for finite total orderings simplifies the reasoning.

## removed rule

The following rewrite rule causes some problems and will be removed:

```
In[4]:= composite[id[x], VERTSECT[y]]

Out[4]= composite[VERTSECT[y], id[image[inverse[VERTSECT[y]], x]]]

In[5]:= composite[id[x_], VERTSECT[y_]] =.
```

## finiteness lemma

Finiteness lemma.

```
In[6]:= SubstTest[member, composite[Id, y], FINITE, y -> composite[id[x], S, id[x]]]

Out[6]= member[composite[id[x], S, id[x]], FINITE] == member[x, FINITE]

In[7]:= member[composite[id[x_], S, id[x_]], FINITE] := member[x, FINITE]
```

## lemma

In this section it is shown that any finite total ordering which is a restriction of the subclass relation is a wellordering.

```
In[8]:= SubstTest[implies, and[subclass[u, v], subclass[v, z]], subclass[u, z],
         {u → cart[y, y], v → cart[x, x]}]

Out[8]= or[not[subclass[y, x]], not[subclass[cart[x, x], z]], subclass[cart[y, y], z]] == True

In[9]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed

In[10]:= Map[not, SubstTest[and, implies[and[p1, p3], p4], implies[and[p2, p3], p5],
           implies[and[p4, p5], p6], not[implies[and[p1, p2, p3], p6]],
           {p1 → member[x, FINITE], p2 → subclass[cart[x, x], union[S, inverse[S]]],
            p3 → subclass[y, x], p4 → member[y, FINITE], p5 →
             subclass[cart[y, y], union[S, inverse[S]]], p6 → or[equal[0, y], member[A[y], y]]}]]

Out[10]= or[equal[0, y], member[A[y], y], not[member[x, FINITE]],
           not[subclass[y, x]], not[subclass[cart[x, x], union[S, inverse[S]]]]] == True

In[11]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Removing the variable **y** yields:

```
In[12]:= Map[equal[V, #] &, SubstTest[class, y, or[equal[0, y], member[A[y], y],
             not[member[x, u]], not[subclass[y, x]], not[subclass[v, w]]],
           {u → FINITE, v → cart[x, x], w → union[S, inverse[S]]}]] // Reverse

Out[12]= or[not[member[x, FINITE]], not[subclass[cart[x, x], union[S, inverse[S]]]],
           subclass[P[x], union[fix[composite[E, BIGCAP]], set[0]]]] == True

In[13]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Restatement.

```
In[14]:= implies[and[equal[y, composite[id[u], S, id[u]]],
           member[y, TO], member[y, FINITE]], member[y, WO]]

Out[14]= True
```

The remaining task is to show that this holds also for any finite total ordering, not just for restrictions of **S**.

## the natural map

A temporary abbreviation is introduced for the natural map.

```
In[15]:= h[x_] := composite[VERTSECT[to[fin[x]]], id[fix[to[fin[x]]]]]
```

Since the natural map is one-to-one, one has:

```
In[16]:= SubstTest[TOTALORDER, composite[oopart[u], to[v], inverse[oopart[u]]],
          {u → h[x], v → fin[x]}]

Out[16]= TOTALORDER[
           composite[VERTSECT[to[fin[x]]], to[fin[x]], inverse[VERTSECT[to[fin[x]]]]]] == True

In[17]:= TOTALORDER[composite[VERTSECT[to[fin[x_]]],
            to[fin[x_]], inverse[VERTSECT[to[fin[x_]]]]]] := True
```

The natural map transforms any total order to a restriction of **inverse[S]**.

```
In[18]:= Map[TOTALORDER, ImageComp[cross[h[x], h[x]], inverse[cross[h[x], h[x]]], inverse[S]]]

Out[18]= TOTALORDER[composite[id[image[VERTSECT[to[fin[x]]], fix[to[fin[x]]]]],
            inverse[S], id[image[VERTSECT[to[fin[x]]], fix[to[fin[x]]]]]]] == True

In[19]:= (% /. x → x_) /. Equal → SetDelayed
```

The inverse of a total order is also a total order.

```
In[20]:= SubstTest[TOTALORDER, inverse[z], z -> restrict[S, range[h[x]], range[h[x]]]] // Reverse

Out[20]= subclass[cart[image[VERTSECT[to[fin[x]]], fix[to[fin[x]]]],
            image[VERTSECT[to[fin[x]]], fix[to[fin[x]]]]], union[S, inverse[S]]] == True

In[21]:= (% /. x → x_) /. Equal → SetDelayed
```

The range of the natural map for a finite total order is finite.

```
In[22]:= SubstTest[implies, member[v, FINITE], member[image[funpart[u], v], FINITE],
          {u -> VERTSECT[to[fin[x]]], v -> fix[to[fin[x]]]}]

Out[22]= member[image[VERTSECT[to[fin[x]]], fix[to[fin[x]]]], FINITE] == True

In[23]:= (% /. x → x_) /. Equal → SetDelayed
```

The general case can now be reduced to the theorem derived for special case of restrictions of **S**.

```
In[24]:= SubstTest[implies,
           and[equal[y, composite[id[u], S, id[u]]], member[y, TO], member[y, FINITE]],
           member[y, WO], u → range[h[x]]] /. y → restrict[S, range[h[x]], range[h[x]]]

Out[24]= subclass[P[image[VERTSECT[to[fin[x]]], fix[to[fin[x]]]]],
           union[fix[composite[E, BIGCAP]], set[0]]] == True

In[25]:= (% /. x → x_) /. Equal → SetDelayed
```

Main result, with wrappers.

```
In[26]:= SubstTest[implies, WELLORDER[w],
          WELLORDER[composite[oopart[u], w, inverse[oopart[u]]]],
          {w → restrict[S, range[h[x]], range[h[x]]], u → inverse[h[x]]}]

Out[26]= WELLORDER[inverse[to[fin[x]]]] == True

In[27]:= WELLORDER[inverse[to[fin[x_]]]] := True

In[28]:= SubstTest[implies, equal[x, to[fin[y]]], WELLORDER[inverse[x]], y → x]

Out[28]= or[not[member[x, FINITE]], not[TOTALORDER[x]], WELLORDER[inverse[x]]] == True

In[29]:= or[not[member[x_, FINITE]], not[TOTALORDER[x_]], WELLORDER[inverse[x_]]] := True

In[30]:= SubstTest[implies, and[member[y, FINITE], TOTALORDER[y]],
          WELLORDER[inverse[y]], y → inverse[to[fin[x]]]]

Out[30]= WELLORDER[to[fin[x]]] == True

In[31]:= WELLORDER[to[fin[x_]]] := True

In[32]:= SubstTest[implies, equal[x, to[fin[y]]], WELLORDER[x], y → x]

Out[32]= or[not[member[x, FINITE]], not[TOTALORDER[x]], WELLORDER[x]] == True

In[33]:= or[not[member[x_, FINITE]], not[TOTALORDER[x_]], WELLORDER[x_]] := True
```

## variable-free formulation

```
In[35]:= Map[equal[V, #] &, SubstTest[class, x, implies[member[x, u], member[x, v]],
            {u -> intersection[FINITE, TO], v → WO}]] // Reverse

Out[35]= subclass[intersection[FINITE, TO], WO] == True

In[36]:= subclass[intersection[FINITE, TO], WO] := True
```