total orders and chains

Johan G. F. Belinfante
2007 August 1

```
In[1]:= (* SetDirectory["l:"]; *) << goedel95.27b; << tools.m

        :Package Title: goedel95.27b      2007 July 27 at 4:55 p.m.

        It is now:  2007 Aug 1 at 13:55

        Loading Simplification Rules

        TOOLS.M                                        Revised 2007 June 25

        weightlimit = 40
```

summary

A **vertical section** of the total order **to[x]** is a class of the form

```
In[2]:= class[v, member[pair[u, v], t]] /. t → to[x]

Out[2]= image[to[x], set[u]]
```

The class of all (small) vertical sections is

```
In[3]:= class[v, exists[u, and[member[u, fix[t]], equal[v, image[t, set[u]]]]]] /. t → to[x]

Out[3]= image[VERTSECT[to[x]], fix[to[x]]]
```

In this notebook it is shown that the class of all vertical sections of a total order is a chain with respect to inclusion. A succinct variable−free restatement is derived that connects the class **TO** of all (small) total orders with the class **chains[S]** of all (small) nests of sets.

derivation

Lemma. (Specialization of a general result that holds for all partial orders to the case of total orders.)

```
In[4]:= SubstTest[composite, VERTSECT[po[t]],
          po[t], inverse[VERTSECT[po[t]]], t → to[x]] // Reverse

Out[4]= composite[VERTSECT[to[x]], to[x], inverse[VERTSECT[to[x]]]] ==
         composite[id[image[VERTSECT[to[x]], fix[to[x]]]],
          inverse[S], id[image[VERTSECT[to[x]], fix[to[x]]]]]

In[5]:= composite[VERTSECT[to[x_]], to[x_], inverse[VERTSECT[to[x_]]]] :=
         composite[id[image[VERTSECT[to[x]], fix[to[x]]]],
          inverse[S], id[image[VERTSECT[to[x]], fix[to[x]]]]]
```

Corollary. (The same result for the inverse relation.)

```
In[6]:= composite[VERTSECT[to[x]], inverse[to[x]], inverse[VERTSECT[to[x]]]] // DoubleInverse

Out[6]= composite[VERTSECT[to[x]], inverse[to[x]], inverse[VERTSECT[to[x]]]] == composite[
          id[image[VERTSECT[to[x]], fix[to[x]]]], S, id[image[VERTSECT[to[x]], fix[to[x]]]]]

In[7]:= composite[VERTSECT[to[x_]], inverse[to[x_]], inverse[VERTSECT[to[x_]]]] := composite[
          id[image[VERTSECT[to[x]], fix[to[x]]]], S, id[image[VERTSECT[to[x]], fix[to[x]]]]]
```

Theorem. The vertical sections of a total order form a chain with respect to inclusion.

```
In[8]:= Map[subclass[#, union[S, inverse[S]]] &, SubstTest[composite, VERTSECT[to[x]],
           union[u, v], inverse[VERTSECT[to[x]]], {u → to[x], v → inverse[to[x]]}]] // Reverse

Out[8]= subclass[cart[image[VERTSECT[to[x]], fix[to[x]]], image[VERTSECT[to[x]], fix[to[x]]]],
          union[S, inverse[S]]] == True

In[9]:= subclass[cart[image[VERTSECT[to[x_]], fix[to[x_]]],
           image[VERTSECT[to[x_]], fix[to[x_]]]], union[S, inverse[S]]] := True
```

Corollary. Restatement without wrappers.

```
In[10]:= SubstTest[implies, equal[t, to[x]],
           subclass[cartsq[image[VERTSECT[t], fix[t]]], union[S, inverse[S]]], x → t] // Reverse

Out[10]= or[not[TOTALORDER[t]],
           subclass[cart[image[VERTSECT[t], fix[t]], image[VERTSECT[t], fix[t]]],
            union[S, inverse[S]]]] == True

In[11]:= or[not[TOTALORDER[t_]],
           subclass[cart[image[VERTSECT[t_], fix[t_]], image[VERTSECT[t_], fix[t_]]],
            union[S, inverse[S]]]] := True
```

Lemma. A similar restatement, less transparent, but more suitable for the task of eliminating variables.

```
In[12]:= SubstTest[implies, equal[t, to[setpart[x]]], member[t,
            image[inverse[VS], image[inverse[IMAGE[SECOND]], chains[S]]]], x → t] // Reverse

Out[12]= or[and[member[domain[thinpart[t]], V], member[image[VERTSECT[t], domain[t]], V],
            subclass[cart[image[VERTSECT[t], domain[t]], image[VERTSECT[t], domain[t]]],
             union[S, inverse[S]]]], not[member[t, V]], not[TOTALORDER[t]]] == True

In[13]:= (% /. t → t_) /. Equal → SetDelayed
```

Theorem. (A succinct variable−free restatement.)

```
In[14]:= Map[equal[V, #] &, SubstTest[class, t, implies[member[t, u], member[t, v]],
           {u → TO, v -> image[inverse[VS], image[inverse[IMAGE[SECOND]], chains[S]]]}]]

Out[14]= subclass[image[IMAGE[SECOND], image[VS, TO]], chains[S]] == True

In[15]:= subclass[image[IMAGE[SECOND], image[VS, TO]], chains[S]] := True
```

comment

An **initial segment** of a total order ≤ is a class of the form **{x | x < y}**. Vertical sections are of the form **{x| y ≤ x}**. Thus, an initial segment is the complement of a vertical section. For the usual total ordering of the class of all ordinals, the initial segments are sets, while the vertical sections are proper classes. In this section it is shown that for this reason, the chain considered in the above is empty for the usual ordering of the ordinals.

```
In[32]:= Map[not, SubstTest[implies, and[subclass[u, v], member[v, V]], member[u, V],
            {u -> dif[OMEGA, ord[x]], v -> intersection[OMEGA, image[S, set[ord[x]]]]}]] //
          Reverse

Out[32]= member[intersection[OMEGA, image[S, set[ord[x]]]], V] == False

In[33]:= member[intersection[OMEGA, image[S, set[ord[x_]]]], V] := False

In[35]:= SubstTest[implies, equal[x, ord[t]],
           not[member[intersection[OMEGA, image[S, set[x]]], V]], t → x] // Reverse

Out[35]= or[not[member[x, OMEGA]], not[member[intersection[OMEGA, image[S, set[x]]], V]]] == True

In[36]:= or[not[member[x_, OMEGA]],
           not[member[intersection[OMEGA, image[S, set[x_]]], V]]] := True
```

Lemma.

```
In[44]:= equal[intersection[OMEGA, complement[P[OMEGA]]], 0]

Out[44]= True

In[46]:= intersection[OMEGA, complement[P[OMEGA]]] := 0
```

Lemma.

```
In[47]:= domain[composite[VERTSECT[composite[id[OMEGA], S]], id[OMEGA]]] // Normality // Reverse

Out[47]= intersection[OMEGA, fix[composite[inverse[S],
             BIGCAP, id[P[OMEGA]], S, VERTSECT[composite[id[OMEGA], S]]]]] ==
          intersection[OMEGA, domain[VERTSECT[composite[id[OMEGA], S]]]]

In[48]:= % /. Equal → SetDelayed
```

Lemma.

```
In[49]:= Map[equal[V, #] &, SubstTest[class, x,
           or[not[member[x, w]], not[member[intersection[w, image[S, set[x]]], V]]], w → OMEGA]]

Out[49]= equal[0, intersection[OMEGA, domain[VERTSECT[composite[id[OMEGA], S]]]]] == True

In[51]:= intersection[OMEGA, domain[VERTSECT[composite[id[OMEGA], S]]]] := 0
```

Theorem.

```
In[52]:= SubstTest[composite, x, id[domain[x]],
          x → composite[VERTSECT[composite[id[OMEGA], S]], id[OMEGA]]]

Out[52]= composite[VERTSECT[composite[id[OMEGA], S]], id[OMEGA]] == 0

In[53]:= composite[VERTSECT[composite[id[OMEGA], S]], id[OMEGA]] := 0
```

Corollary.

```
In[55]:= ImageComp[VERTSECT[composite[id[OMEGA], S]], id[OMEGA], V] // Reverse

Out[55]= image[VERTSECT[composite[id[OMEGA], S]], OMEGA] == 0

In[56]:= image[VERTSECT[composite[id[OMEGA], S]], OMEGA] := 0
```

The chain **image[VERTSECT[t], fix[t]]** is therefore empty for the relation **restrict[S,OMEGA,OMEGA]**.

```
In[57]:= image[VERTSECT[t], fix[t]] /. t → restrict[S, OMEGA, OMEGA]

Out[57]= 0
```