Zermelo’s theorems about towers

Johan G. F. Belinfante
2007 October 4

```
In[1]:= (* SetDirectory["l:"]; *) << goedel98.04a; << tools.m

:Package Title: goedel98.04a          2007 October 4 at 1:30 p.m.

It is now:  2007 Oct 5 at 9:5

Loading Simplification Rules

TOOLS.M                                    Revised 2007 September 19

weightlimit = 40
```

summary

Rewrite rules are derived about invariant subsets that are closed under unions of chains. Halmos attributes these theorems to Zermelo, who used these results to show that Zorn’s lemma follows from the axiom of choice.

```
In[2]:= "Paul R. Halmos, Naive Set Theory, D. Van Nostrand Company, Inc., 1960.  (see page 63)";
```

temporary definitions

Temporary definition.   A class **t** is an **x**–tower if it is invariant under **x** and **t** is closed under unions of chains.

```
In[3]:= twr[x_, t_] := and[equal[t, Uchains[t]], subclass[image[x, t], t]]
```

Temporary definition. The class of all sets that are **x**–towers.

```
In[4]:= towers[x_] := intersection[fix[UCHAINS], invar[x]]
```

the intersection of the class of all towers

Theorem. The intersection of the class of all towers of **x** is invariant under **x**.

```
In[5]:= SubstTest[implies, subclass[t, invar[x]],
          invariant[x, A[t]], t → intersection[fix[UCHAINS], invar[x]]] // Reverse

Out[5]= subclass[image[x, A[intersection[fix[UCHAINS], invar[x]]]],
          A[intersection[fix[UCHAINS], invar[x]]]] == True

In[6]:= subclass[image[x_, A[intersection[fix[UCHAINS], invar[x_]]]],
          A[intersection[fix[UCHAINS], invar[x_]]]] := True
```

Lemma. (This can be improved upon.)

```
In[7]:= SubstTest[implies, equal[t, fix[HULL[t]]],
           or[empty[t], member[A[t], t]], t → towers[x]] // Reverse // MapNotNot

Out[7]= or[equal[0, intersection[fix[UCHAINS], invar[x]]],
          equal[A[intersection[fix[UCHAINS], invar[x]]],
           Uchains[A[intersection[fix[UCHAINS], invar[x]]]]]] == True
```

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

Theorem. The intersection of the class of all towers is closed under unions of chains.

```
In[9]:= SubstTest[and, implies[p, q], or[p, q],
         {p -> equal[0, towers[x]], q -> equal[A[towers[x]], Uchains[A[towers[x]]]]}]

Out[9]= equal[A[intersection[fix[UCHAINS], invar[x]]],
          Uchains[A[intersection[fix[UCHAINS], invar[x]]]]] == True

In[10]:= Uchains[A[intersection[fix[UCHAINS], invar[x_]]]] := A[intersection[fix[UCHAINS], invar[x]]]
```

the least tower

Lemma. (This temporary result will be improved upon in the sequel, eliminating the sethood hypothesis.)

```
In[11]:= SubstTest[implies, member[w, t], subclass[A[t], w], t → towers[x]] // Reverse

Out[11]= or[not[equal[w, Uchains[w]]], not[member[w, V]], not[subclass[image[x, w], w]],
          subclass[A[intersection[fix[UCHAINS], invar[x]]], w]] == True

In[12]:= (% /. {w → w_, x → x_}) /. Equal → SetDelayed
```

Lemma.

```
In[13]:= SubstTest[implies, and[twr[x, u], twr[x, v]],
          invariant[x, intersection[u, v]], v → A[towers[x]]] // Reverse

Out[13]= or[not[equal[u, Uchains[u]]], not[subclass[image[x, u], u]],
          subclass[image[x, intersection[u, A[intersection[fix[UCHAINS], invar[x]]]]],
           A[intersection[fix[UCHAINS], invar[x]]]]] == True

In[14]:= (% /. {u → u_, x → x_}) /. Equal → SetDelayed
```

Lemma.

```
In[15]:= SubstTest[implies, and[twr[x, u], twr[x, v]],
          equal[Uchains[intersection[u, v]], intersection[u, v]], v → A[towers[x]]] // Reverse

Out[15]= or[equal[intersection[u, A[intersection[fix[UCHAINS], invar[x]]]],
           Uchains[intersection[u, A[intersection[fix[UCHAINS], invar[x]]]]]],
          not[equal[u, Uchains[u]]], not[subclass[image[x, u], u]]] == True

In[16]:= (% /. {u → u_, x → x_}) /. Equal → SetDelayed
```

Lemma.

```
In[17]:= implies[and[not[empty[towers[x]]], equal[w, Uchains[w]], subclass[image[x, w], w]],
          member[intersection[w, A[towers[x]]], towers[x]]] // NotNotTest

Out[17]= or[and[equal[intersection[w, A[intersection[fix[UCHAINS], invar[x]]]],
            Uchains[intersection[w, A[intersection[fix[UCHAINS], invar[x]]]]]],
           member[intersection[w, A[intersection[fix[UCHAINS], invar[x]]]], V],
           subclass[image[x, intersection[w, A[intersection[fix[UCHAINS], invar[x]]]]],
            A[intersection[fix[UCHAINS], invar[x]]]]],
          equal[0, intersection[fix[UCHAINS], invar[x]]], not[equal[w, Uchains[w]]],
          not[subclass[image[x, w], w]]] == True

In[18]:= (% /. {w → w_, x → x_}) /. Equal → SetDelayed
```

Lemma.

In[19]:=
```
SubstTest[implies, member[t, towers[x]],
  subclass[A[towers[x]], t], t → intersection[w, A[towers[x]]]] // Reverse
```

Out[19]=
```
or[not[equal[intersection[w, A[intersection[fix[UCHAINS], invar[x]]]],
    Uchains[intersection[w, A[intersection[fix[UCHAINS], invar[x]]]]]]],
  not[member[intersection[w, A[intersection[fix[UCHAINS], invar[x]]]], V]],
  not[subclass[image[x, intersection[w, A[intersection[fix[UCHAINS], invar[x]]]]], w]],
  not[subclass[image[x, intersection[w, A[intersection[fix[UCHAINS], invar[x]]]]],
    A[intersection[fix[UCHAINS], invar[x]]]]],
  subclass[A[intersection[fix[UCHAINS], invar[x]]], w]] == True
```

In[20]:=
```
(% /. {w → w_, x → x_}) /. Equal → SetDelayed
```

Theorem. If towers exist, then the least tower is contained in every invariant subclass which is closed under unions of chains.

In[21]:=
```
Map[not, SubstTest[and, implies[and[p1, p2], p3],
   implies[p3, p4], not[implies[and[p1, p2], p4]], {p1 → not[empty[towers[x]]],
    p2 → and[equal[w, Uchains[w]], subclass[image[x, w], w]],
    p3 -> member[intersection[w, A[towers[x]]], towers[x]],
    p4 → subclass[A[towers[x]], intersection[w, A[towers[x]]]]}]] // Reverse
```

Out[21]=
```
or[equal[0, intersection[fix[UCHAINS], invar[x]]], not[equal[w, Uchains[w]]],
  not[subclass[image[x, w], w]], subclass[A[intersection[fix[UCHAINS], invar[x]]], w]] == True
```

In[22]:=
```
or[equal[0, intersection[fix[UCHAINS], invar[x_]]],
  not[equal[w_, Uchains[w_]]], not[subclass[image[x_, w_], w_]],
  subclass[A[intersection[fix[UCHAINS], invar[x_]]], w_]] := True
```

the greatest spinal element of the least tower

Lemma.

In[23]:=
```
SubstTest[implies, member[t, V],
  member[U[spine[S, t]], Uchains[t]], t → A[towers[x]]] // Reverse
```

Out[23]=
```
or[equal[0, intersection[fix[UCHAINS], invar[x]]],
  member[U[spine[S, A[intersection[fix[UCHAINS], invar[x]]]]],
   A[intersection[fix[UCHAINS], invar[x]]]]] == True
```

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

Lemma.

In[25]:=
```
spine[S, V] // Normality
```

Out[25]=
```
spine[S, V] == set[0]
```

In[26]:=
```
spine[S, V] := set[0]
```

Theorem about the greatest spinal element of the minimal tower.

In[27]:=
```
SubstTest[and, implies[p, q], or[p, q],
 {p -> equal[0, towers[x]], q -> member[U[spine[S, A[towers[x]]]], spine[S, A[towers[x]]]]}]
```

Out[27]=
```
member[U[spine[S, A[intersection[fix[UCHAINS], invar[x]]]]],
  A[intersection[fix[UCHAINS], invar[x]]]] == True
```

In[28]:=
```
member[U[spine[S, A[intersection[fix[UCHAINS], invar[x_]]]]],
  A[intersection[fix[UCHAINS], invar[x_]]]] := True
```

existence of towers

Theorem. If the domain of a unary operation is closed under unions of chains, then the intersection of the class of its towers is a subset of its domain.

In[29]:=
```
Map[implies[member[x, UNOPS], #] &, SubstTest[implies, member[u, v], subclass[A[v], u],
    {u → domain[x], v → intersection[fix[UCHAINS], invar[x]]}]] // Reverse // MapNotNot
```

Out[29]=
```
or[not[equal[domain[x], Uchains[domain[x]]]], not[member[x, UNOPS]],
  subclass[A[intersection[fix[UCHAINS], invar[x]]], domain[x]]] == True
```

In[30]:=
```
or[not[equal[domain[x_], Uchains[domain[x_]]]], not[member[x_, UNOPS]],
  subclass[A[intersection[fix[UCHAINS], invar[x_]]], domain[x_]]] := True
```

Lemma.

In[31]:=
```
SubstTest[implies, and[subclass[t, y], member[y, z]], member[t, V], t → A[x]] // Reverse
```

Out[31]=
```
or[not[equal[0, x]], not[member[y, z]], not[subclass[A[x], y]]] == True
```

In[32]:=
```
or[not[equal[0, x_]], not[member[y_, z_]], not[subclass[A[x_], y_]]] := True
```

Corollary. If the domain of a unary operation is closed under unions of chains, then it has towers.

In[33]:=
```
Map[not, SubstTest[and, implies[and[p1, p2], p3],
    implies[p1, p4], implies[and[p3, p4], p5], not[implies[and[p1, p2], p5]],
    {p1 → member[x, UNOPS], p2 -> equal[domain[x], Uchains[domain[x]]],
     p3 -> subclass[A[intersection[fix[UCHAINS], invar[x]]], domain[x]],
     p4 → member[domain[x], V], p5 → not[empty[intersection[fix[UCHAINS], invar[x]]]]}]] //
  Reverse
```

Out[33]=
```
or[not[equal[0, intersection[fix[UCHAINS], invar[x]]]],
  not[equal[domain[x], Uchains[domain[x]]]], not[member[x, UNOPS]]] == True
```

In[34]:=
```
or[not[equal[0, intersection[fix[UCHAINS], invar[x_]]]],
  not[equal[domain[x_], Uchains[domain[x_]]]], not[member[x_, UNOPS]]] := True
```