axch <= Zorn's lemma

Johan G. F. Belinfante
2007 November 15

summary

In this notebook it is shown that the axiom of choice is equivalent to two particular versions of Zorn's lemma. Both versions are concerned only with collections of sets partially ordered by inclusion, not with more general partial orderings. The more special of the two versions of Zorn's lemma considered in this notebook states that, the axiom of choice is equivalent to the statement that if a set $x$ of sets is closed under unions of chains, then $x$ has a maximal element. The more general version says that, the axiom of choice is equivalent to the statement that it is not possible for every chain contained in a set $x$ to have a proper upper bound in $x$. The derivations of both versions are based on a basic version of Zorn's lemma derived earlier using Zermelo's theorem about covering operations. In fact, the only formal difference between the variable-free formulation of the special version of Zorn's lemma derived here and that of the basic version derived earlier is that the proper subset relation $PS$ replaces the cover relation $K$.

introduction

The condition that a set $x$ be subvariant under the inverse of the proper subset relation $PS$ is equivalent to the statement that $x$ does not have a maximal element.

```
In[2]:= assert[
    forall[y, implies[member[y, x], exists[z, and[member[pair[y, z], PS], member[z, x]]]]]
```

```
Out[2]= subclass[x, image[inverse[PS], x]]
```

The class of all such sets enters into the statement of the special version of Zorn's lemma.

```
In[3]:= class[x, subclass[x, image[inverse[PS], x]]]
```

```
```
Lemma.

In \[4\] := \text{ImageComp[inverse[BIGCUP], inverse[S], x]}

Out[4] = image[inverse[S], image[POWER, x]] = image[inverse[BIGCUP], image[inverse[S], x]]

In[5] := image[inverse[S], image[POWER, x_]] := image[inverse[BIGCUP], image[inverse[S], x]]

The condition that every chain in \(x\) have an upper bound can be written without quantifiers as follows:

In[6] := assert[forall[t, implies[and[subclass[t, x], member[t, chains[S]]],
   exists[u, and[member[u, x], forall[v, implies[member[v, t], subclass[v, u]]]]]]]]

Out[6] = subclass[Uchains[x], image[inverse[S], x]]

The class of sets satisfying this chain condition is:

In[7] := class[x, subclass[Uchains[x], image[inverse[S], x]]]


Similar results hold for the condition that every chain in \(x\) have a proper upper bound, but with the subset relation \(S\) replaced by the proper subset relation \(PS\).

---

**chain conditions**

Theorem. The class \(\text{fix}[UChains]\) of sets closed under unions of chains is a subclass of the class of sets with the property that every chain has an upper bound.

In[8] := Map[equal[V, # &], complement[dif[fix[UChains],
   fix[composite[inverse[IMAGE[inverse[S]]]], S, UChains]]]] // Renormality


Lemma.

In[10] := Map[empty[composite[Id, complement[#]] &,
   complement[dif[composite[inverse[S], IMAGE[inverse[PS]]],
   composite[inverse[S], IMAGE[inverse[S]]]]]] // VSNormality


Theorem.
Assert Test

Reverse:

3

Assert Test

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

Lemma.

Lemma. If every chain in x has a proper upper bound, then x has no maximal elements.

Conversely, if every chain has an upper bound, and there are no maximal elements, then every chain has a proper upper bound.

Theorem. (A rewrite rule combining the two preceding results.)
covering a chain

Lemma. If $y$ contains every member of $x$, then the $x$ is comparable to all member of $x$.

Technical lemma to prepare for the next lemma.

Lemma. If $u$ is an upper bound for a chain $t$ of sets, then $\text{union}[t, \text{set}[u]]$ is also a chain of sets which covers $t$.

Lemma. (If $u$ is a proper class, its singleton is empty.)
Lemma. (Improvement of a prior lemma, removing a redundant sethood literal.)

In[32]:= SubstTest[and, implies[p, q], or[p, q], {p → member[u, V],
q → or[notsubclass[cart[t, t], union[S, inverse[S]]], notsubclass[U[t], u],
subclass[cart[union[t, set[u]], union[t, set[u]], union[S, inverse[S]]]]]]

Out[32]= or[notsubclass[cart[t, t], union[S, inverse[S]]], notsubclass[U[t], u],
subclass[cart[union[t, set[u]], union[t, set[u]], union[S, inverse[S]]]]] = True

In[33]:= (% /. {t → t_, u → u_}) /. Equal → SetDelayed

Lemma.

In[34]:= SubstTest[implies, and[member[pair[v, u]], composite[Id, t]], member[v, x]],
member[u, image[t, x]], t → inverse[K] // Reverse

Out[34]= or[member[u, image[inverse[K], x]],
not[member[v, x]], not[member[pair[u, v], K]]] = True

In[35]:= (% /. {u → u_, v → v_, x → x_}) /. Equal → SetDelayed

Application of the above lemma: the case that v covers a chain t in x.

In[36]:= SubstTest[implies,
and[member[pair[t, v], K], member[v, y]], member[t, image[inverse[K], y]],
{v → union[t, set[u]], y → intersection[P[x], chains[S]]}] // Reverse

Out[36]= or[member[t, image[inverse[K], intersection[chains[S], P[x]]]],
member[u, t], not[member[t, V]], not[member[u, x]], notsubclass[t, x],
notsubclass[cart[union[t, set[u]], union[t, set[u]], union[S, inverse[S]]]]] = True

In[37]:= (% /. {t → t_, u → u_, x → x_}) /. Equal → SetDelayed

Theorem. Covering a chain: if t is a chain of sets in x, and u is a proper upper bound for t, then t can be covered by some chain in x. Comment: this derivation takes about a minute: be patient.

In[38]:= Map[not, SubstTest[and, implies[and[p4, p5, p6],
implies[and[p2, p4], p7], implies[and[p1, p2, p3, p6, p7], p8],
not[implies[and[p1, p2, p3, p4, p5], p8]], {p1 → andsubclass[t, x], member[t, V],
p2 → subclass[cart[t, t], union[S, inverse[S]]],
p3 → member[u, x], p4 → andsubclass[U[t], u], member[u, V],
p5 → not[equal[u, U[t]]], p6 → not[member[u, t]],
p7 → subclass[cart[union[t, set[u]], union[t, set[u]], union[S, inverse[S]]],
p8 → member[t, image[inverse[K], intersection[chains[S], P[x]]]]}] // Reverse

Out[38]= or[equal[u, U[t]], member[t, image[inverse[K], intersection[chains[S], P[x]]]],
not[member[t, V]], not[member[u, x]], notsubclass[t, x],
notsubclass[cart[t, t], union[S, inverse[S]]], notsubclass[U[t], u]] = True

In[39]:= (% /. {t → t_, u → u_, x → x_}) /. Equal → SetDelayed

The following lemma is needed to clean up the result obtained when the variables t and u in the preceding theorem are eliminated.
The special case of Zorn's lemma

Corollary. If a class $x$ is closed under unions of chains and has no maximal elements, then the class of chains in $x$ is subvariant under $\text{inverse}[K]$. 

$$\text{Map}[\text{not}, \text{SubstTest}[\text{and}, \text{implies}[\text{and}[\text{p1, p2}], \text{p3}], \text{implies}[\text{p3, p4}], \text{not}[\text{implies}[\text{and}[\text{p1, p2}], \text{p4}]], \{\text{p1 -> equal[Uchains}[x], x], \text{p2 -> subvariant[\text{inverse}[PS], x], p3 -> subclass[Uchains}[x], \text{image[\text{inverse}[PS], x]], \text{p4 -> subvariant[\text{inverse}[K], \text{intersection}[\text{chains}[S], \text{P}[x]]]]}]]) ] // \text{Reverse}$$

$$\text{or}[\text{not}[\text{equal}[x, \text{Uchains}[x]]], \text{not}[\text{subclass}[x, \text{image[\text{inverse}[PS], x]]]], \text{subclass}[\text{intersection}[\text{chains}[S], \text{P[x]]]], \text{image[\text{inverse}[K], \text{intersection}[\text{chains}[S], \text{P[x]]]]}] := \text{True}$$
In[47]:= or[not[equal[x_, Uchains[x_]]],
    not[subclass[x_, image[inverse[PS], x_]]], subclass[intersection[chains[S], P[x_]],
    image[inverse[K], intersection[chains[S], P[x_]]]] := True

Lemma.

In[48]:= SubstTest[implies, and[equal[Uchains[u], u], equal[Uchains[v], v]],
    equal[Uchains[intersection[u, v]], intersection[u, v]],
    {u -> chains[x], v -> P[y]}]

Out[48]= True = equal[intersection[chains[x], P[y]], Uchains[intersection[chains[x], P[y]]]]

In[49]:= Uchains[intersection[chains[x_], P[y_]]] := intersection[chains[x], P[y]]

Lemma.

In[50]:= SubstTest[implies, disjoint[u, v],
    not[and[member[t, u], member[t, v]]], {t -> intersection[chains[S], P[x]],
    u -> fix[UCHAINS], v -> subvar[inverse[K]]}] // Reverse

Out[50]= or[not[axch], not[member[intersection[chains[S], P[x]], V]],
    not[subclass[intersection[chains[S], P[x]],
    image[inverse[K], intersection[chains[S], P[x]]]]] = True

In[51]:= (% /. x -> x_) /. Equal -> SetDelayed

Theorem. Special case of Zorn’s lemma: the axiom of choice implies that if a set is closed under unions of chains, then it has a maximal element.

In[52]:= Map[implies/member[x, y], not[[]]] &,
    SubstTest[and, implies[p1, p2], implies[p0, not[p2]], not[implies[p0, not[p1]]],
    {p0 -> axch, p1 -> member[x, intersection[fix[UCHAINS], subvar[inverse[PS]]]]},
    p2 -> member[intersection[chains[S], P[x]], subvar[inverse[K]]]]] // Reverse

Out[52]= or[not[axch], not[equal[x, Uchains[x]]],
    not[member[x, y]], not[subclass[x, image[inverse[PS], x]]]] = True

In[53]:= or[not[axch], not[equal[x_, Uchains[x_]]],
    not[member[x_, y_]], not[subclass[x_, image[inverse[PS], x_]]]] := True

Eliminating the variable x yields:

In[54]:= Map[equal[V, []] &, SubstTest[class, x, implies[equal[s, V], not[member[x, t]]],
    {s -> SELECT, t -> intersection[fix[UCHAINS], subvar[inverse[PS]]]]}]

Out[54]= or[equal[0, intersection[fix[UCHAINS], subvar[inverse[PS]]]], not[axch]] = True

In[55]:= % /. Equal -> SetDelayed

Lemma.
the more general version of Zorn's lemma

Lemma.

Theorem. The more general form of Zorn's lemma: the axiom of choice implies that it is not possible for every chain in a set to have a proper upper bound.

Theorem. The special case of Zorn's lemma can be combined with the converse to obtain a simple rewrite rule.
Eliminating the variable $x$ yields:

```plaintext
In[66]:= Map[equal[V, #] &,
    SubstTest[class, x, implies[and[member[x, V], equal[s, V]], not[member[x, t]]],
    {s -> SELECT, t -> fix[composite[inverse[IMAGE[inverse[PS]]], S, UCHAINS]]}]]
```

```plaintext
Out[66]= or[equal[0, fix[composite[inverse[IMAGE[inverse[PS]]], S, UCHAINS]]], not[axch]] = True
```

Converse Theorem. The more general version of Zorn's lemma implies the axiom of choice.

```plaintext
In[68]:= SubstTest[implies, and[subclass[u, v], disjoint[v, w]], disjoint[u, w],
    {u -> fix[UCHAINS], v -> fix[composite[inverse[IMAGE[inverse[S]]], S, UCHAINS]]],
    w -> subvar[inverse[PS]]}] // Reverse
```

```plaintext
Out[68]= or[axch, not[equal[0, fix[composite[inverse[IMAGE[inverse[PS]]], S, UCHAINS]]]]] = True
```

Theorem. A rewrite rule for the more general formulation of Zorn's lemma.

```plaintext
In[70]:= equiv[equal[0, fix[composite[inverse[IMAGE[inverse[PS]]], S, UCHAINS]]], axch]
```

```plaintext
Out[70]= True
```

```plaintext
In[71]:= equal[0, fix[composite[inverse[IMAGE[inverse[PS]]], S, UCHAINS]]] := axch
```