a variable-free form of Zorn's lemma for posets

Johan G. F. Belinfante
2008 July 8

```
In[1]:= SetDirectory["l:"]; << goedel.08jul06a; << tools.m

        :Package Title: goedel.08jul06a                 2008 July 6 at 11:05 p.m.

        It is now:  2008 Jul 8 at 16:45

        Loading Simplification Rules

        TOOLS.M                                    Revised 2008 July 5

        weightlimit = 40
```

summary

A variable-free statement of Zorn's lemma for partially ordered sets is derived. The statement uses the function **UBD** which takes any set **x** to the set **domain[UB[x]]** of all sets which have upper bounds with respect to **x**.

chain condition

Theorem. (Membership rule for the class of sets satisfying the chain condition in Zorn's lemma.)

```
In[2]:= member[x, fix[composite[inverse[UBD], S, CHAINS]]] // AssertTest

Out[2]= member[x, fix[composite[inverse[UBD], S, CHAINS]]] ==
         and[member[x, V], subclass[chains[x], domain[UB[x]]]]

In[3]:= member[x_, fix[composite[inverse[UBD], S, CHAINS]]] :=
         and[member[x, V], subclass[chains[x], domain[UB[x]]]]
```

Zorn's lemma

Theorem. (Variable-free statement of Zorn's lemmma.)

```
In[4]:= Map[equal[V, #] &,
         SubstTest[class, x, implies[and[equal[V, u], member[x, v]], empty[x]],
          {u → SELECT, v -> intersection[PO, image[inverse[FUNPART], set[0]],
             fix[composite[inverse[UBD], S, CHAINS]]]}]]

Out[4]= or[not[axch], subclass[intersection[PO, fix[composite[inverse[UBD], S, CHAINS]],
            image[inverse[FUNPART], set[0]]], set[0]]] == True

In[5]:= % /. Equal → SetDelayed
```

The remainder of this notebook is concerned with the converse of Zorn's lemma. To derive it, one needs to specialize Zorn's lemma to the class **image[IMAGE[id[S]], image[CART,Id]]** of restrictions of the subset relation **S**.

Technical Lemma. (simplification rule)

```
In[15]:= SubstTest[range, composite[inverse[funpart[t]], funpart[t]],
           t → composite[IMAGE[id[S]], CART, DUP]] // Reverse

Out[15]= fix[image[inverse[CART],
            image[inverse[IMAGE[id[S]]], image[IMAGE[id[S]], image[CART, Id]]]]] == V

In[16]:= % /. Equal → SetDelayed
```

Technical Lemma. (simplification rule)

```
In[17]:= equal[0, intersection[fix[image[inverse[CART], image[inverse[IMAGE[id[S]]],
               intersection[complement[set[0]], fix[composite[inverse[UBD], S, CHAINS]],
                image[IMAGE[id[S]], image[CART, Id]]]]]], subvar[inverse[PS]]]] // AssertTest

Out[17]= equal[0, intersection[fix[image[inverse[CART], image[inverse[IMAGE[id[S]]],
               intersection[complement[set[0]], fix[composite[inverse[UBD], S, CHAINS]],
                image[IMAGE[id[S]], image[CART, Id]]]]]], subvar[inverse[PS]]]] == axch

In[18]:= % /. Equal → SetDelayed
```

Lemma. (The special case of Zorn's lemma implies the axiom of choice.)

```
In[20]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
          {t → composite[inverse[DUP], inverse[CART], inverse[IMAGE[id[S]]]],
           u → intersection[complement[set[0]], fix[composite[inverse[UBD], S, CHAINS]],
             image[IMAGE[id[S]], image[CART, Id]]],
           v → complement[image[inverse[FUNPART], set[0]]]}] // Reverse

Out[20]= or[axch, not[
            subclass[intersection[fix[composite[inverse[UBD], S, CHAINS]], image[IMAGE[id[S]],
               image[CART, Id]]], image[inverse[FUNPART], set[0]]], set[0]]]] == True

In[21]:= % /. Equal → SetDelayed
```

Lemma. (The general case implies the special case, and therefore implies **axch**.)

```
In[23]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], not[implies[p1, p3]],
            {p1 → subclass[intersection[PO, fix[composite[inverse[UBD], S, CHAINS]],
               image[inverse[FUNPART], set[0]]], set[0]], p2 → subclass[intersection[
               fix[composite[inverse[UBD], S, CHAINS]], image[IMAGE[id[S]], image[CART, Id]],
               image[inverse[FUNPART], set[0]]], set[0]], p3 → axch}]] // Reverse

Out[23]= or[axch, not[subclass[intersection[PO, fix[composite[inverse[UBD], S, CHAINS]],
              image[inverse[FUNPART], set[0]]], set[0]]]] == True

In[24]:= % /. Equal → SetDelayed
```

Main Theorem.   Zorn's lemma for posets is equivalent to the axiom of choice.

```
In[25]:= equiv[subclass[intersection[PO, fix[composite[inverse[UBD], S, CHAINS]],
            image[inverse[FUNPART], set[0]]], set[0]], axch]

Out[25]= True

In[26]:= subclass[intersection[PO, fix[composite[inverse[UBD], S, CHAINS]],
            image[inverse[FUNPART], set[0]]], set[0]] := axch
```

Comment. The class **image[inverse[FUNPART], set[0]]** that appears here is that class of all x which are not single-valued at any point.

```
In[27]:= member[x, image[inverse[FUNPART], set[0]]]

Out[27]= and[equal[0, funpart[x]], member[x, V]]
```

For a partially ordered set, this statement is equivalent to saying that its fixed point set has no maximal element. The statement of the main theorem says that the axiom of choice is equivalent to the statement that the only (small) partial order relation satisfying the condition that all chains have upper bounds and whose fixed point set has no maximal element is the empty relation.