PO ⊂ image[inverse[S], TO]

Johan G. F. Belinfante
2007 December 30

\texttt{In[1]:=} \texttt{SetDirectory["l:"]; \texttt{\textbackslash<goedel.07dec29a; \texttt{\textbackslash}} tools.m}

\texttt{:Package Title: goedel.07dec29a} \hspace{1cm} 2007 December 29 at 7:00 p.m.

\texttt{It is now: 2007 Dec 30 at 3:52}

\texttt{Loading Simplification Rules}

\texttt{TOOLS.M} \hspace{1cm} Revised 2007 December 29

\texttt{weightlimit = 40}

\textbf{introduction}

One application of Zorn's lemma is the Szpilrajn theorem, which says that any partial order can be extended to a total order. For this application, the class to which Zorn's lemma is applied is not the class PO of all partial orders, but rather the class of all partial orders contained in a fixed cartesian square.


To prove this theorem, one needs to show that a maximal partial order on a given set is a total order. If \( x \) is a partial order and if an ordered pair \( \langle u,v \rangle \) does not belong to \( x \), then one can extend the partial order \( x \) one by adjoining to \( x \) the cartesian product of \texttt{image[inverse[x], set[u]]} and \texttt{image[x, set[v]]}. This extension is proper if the ordered pair \( \langle v,u \rangle \) also does not belong to \( x \).

\textbf{adjoining an identity}

A partial order cannot be maximal if it can be extended by adjoining an identity. In this section it is shown that if a partial order \( x \) is maximal in the cartesian square \texttt{cart[y, y]}, then \( y = \texttt{fix[x]} \).

Lemma. We begin by wrapping \( x \) with \texttt{po}. Note that the desired equation \( y = \texttt{fix[po[x]]} \) is here replaced by an equivalent inclusion. The use of the \texttt{po} wrapper causes the inclusion of the partial order in \texttt{cart[y, y]} to be rewritten as \texttt{subclass[fix[po[x]],y]}. 

adjoining a cartesian product

A partial order cannot be maximal if it can be extended by adjoining a cartesian product. In this section it is shown that if a partial order $x$ is maximal in the cartesian square $cart[y, y]$, then $x$ is a total order. One needs to consider separately the transitive, reflexive and antisymmetric properties for extensions of a partial order by a (certain type of) cartesian product. For convenience only the case $y = \text{fix}[x]$ is considered for the initial lemmas, thereby reducing the number of free variables.

Lemma. For the transitive property, the following suffices.

In[9]:= \text{SubstTest}[\text{subclass}, \text{composite}[t, t], t, \\
\text{t} \to \text{union}[\text{po}[x], \text{cart}[\text{image}[\text{inverse}[\text{po}[x]]], \text{u}], \text{image}[\text{po}[x], \text{v}]]] \\
Out[9]= \text{TRANSITIVE}[\text{union}[\text{cart}[\text{image}[\text{inverse}[\text{po}[x]]], \text{u}], \text{image}[\text{po}[x], \text{v}]], \text{po}[x]]] = \text{True}
In[10]:=(% // {x -> x_, u -> u_, v -> v_})/. Equal -> SetDelayed

Lemma.

In[12]:= SubstTest[REFLEXIVE, union[id[x], rfx[t]], t -> po[y]] // Reverse

Out[12]= REFLEXIVE[union[id[x], po[y]]] = True

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

Lemma. For the reflexive property, the following suffices.

In[14]:= SubstTest[subclass, t, cartsq[fix[t]],
   t -> union[cart[image[inverse[po[x]], y], image[po[x], z]], po[x]]]

Out[14]= REFLEXIVE[union[cart[image[inverse[po[x]], y], image[po[x], z]], po[x]]] = True

In[15]:=(% // {x -> x_, y -> y_, z -> z_})/. Equal -> SetDelayed

For the antisymmetric property, two lemmas are needed. This is the first:

In[17]:= SubstTest[implies, empty[t], subclass[t, Id], t -> composite[
   id[image[inverse[po[x]], set[z]]], po[x], id[image[po[x], set[y]]]]] // Reverse

Out[17]= or[member.pair[y, z], po[x]], subclass[composite[
   id[image[inverse[po[x]], set[z]]], po[x], id[image[po[x], set[y]]]], Id] = True

In[18]:=(% // {x -> x_, y -> y_, z -> z_})/. Equal -> SetDelayed

The second lemma is similar, but involves inverse[po[x]] instead of po[x].

In[19]:= SubstTest[implies, empty[t], subclass[t, Id], t -> composite[id[image[po[x], set[y]]],
   inverse[po[x]], id[image[inverse[po[x], set[z]]]]]] // Reverse

Out[19]= or[member.pair[y, z], po[x]], subclass[composite[id[image[po[x], set[y]]],
   inverse[po[x]], id[image[inverse[po[x], set[z]]]], Id] = True

In[20]:=(% // {x -> x_, y -> y_, z -> z_})/. Equal -> SetDelayed

Theorem. Combining the above lemmas, one obtains this result: if pair[y, z] does not belong to po[x], then one can extend po[x] by adjoining a cartesian product.

In[21]:= Map[implies[not[member.pair[y, z], po[x]]], #] &,
   SubstTest[and, REFLEXIVE[t], ANTISYMMETRIC[t], TRANSITIVE[t],
   t -> union[cart[image[inverse[po[x]], set[z]], image[po[x], set[y]]], po[x]]]] // MapNotNot

Out[21]= or[member.pair[y, z], po[x]], PARTIALORDER[union[cart[image[inverse[po[x]], set[z]], image[po[x], set[y]]], po[x]]] = True

In[22]:=(% // {x -> x_, y -> y_, z -> z_})/. Equal -> SetDelayed

The extension is not proper unless the reverse ordered pair also does not belong to po[x].
The following lemma introduces the class \texttt{image[\texttt{inverse}]},\texttt{intersection[\texttt{PO},\texttt{P[cart]}\texttt{[y, y]]]}]. Both \texttt{po} and \texttt{setpart} wrappers are used here.

If neither \texttt{pair[y,z]} nor \texttt{pair[z,y]} belongs to \texttt{po[x]}, and \texttt{y} = \texttt{fix[po[x]]}, then \texttt{po[x]} is not a maximal element of \texttt{intersection[PO, P[cart[y, y]]]}. 

Next the variables \(y\) and \(z\) are eliminated. For convenience, the `setpart` wrapper is reintroduced here.

Removing both the `po` and `setpart` wrappers yields:

This result is now combined with that of the preceding section.
Finally, we eliminate the variable $x$.

```plaintext
In[49]:= Map[equality[v, #] & , SubstTest[Class, x, or[member[x, u], not[member[x, v]], 
not[member[y, v]], not[subclass[x, cart[y, y]]], member[x, w]], 
{u -> image[inverse[PS], intersection[PO, P[cart[y, y]]]], v -> PO, w -> TO}]
```

```plaintext
Out[49]= or[not[member[y, V]], subclass[intersection[PO, P[cart[y, y]]], 
union[TO, image[inverse[PS], intersection[PO, P[cart[y, y]]]]]] = True
```

```plaintext
In[50]:= (% /. y -> y_) /. Equal -> SetDelayed
```

---

**Applying Zorn's Lemma**

A strong version of Zorn's lemma is applied to the intersection of $PO$ with the power set of a cartesian square, using `setpart` wrappers.

```plaintext
In[51]:= SubstTest[implies, and[axch, member[t, V], subclass[Uchains[t], image[inverse[S], t]]], 
subclass[t, image[inverse[S], intersection[t, complement[setpart[Porous]]], t]], 
t -> intersection[PO, P[cartsq[setpart[y]]]]] // Reverse
```

```plaintext
Out[51]= or[not[axch], subclass[intersection[PO, P[cart[setpart[y]]], setpart[y]]], 
image[inverse[S], intersection[PO, setpart[y]], intersection[PO, P[cart[setpart[y]]], setpart[y]]]] = True
```

```plaintext
In[52]:= (% /. y -> y_) /. Equal -> SetDelayed
```

The `setpart` wrapper is now removed:

```plaintext
In[53]:= SubstTest[implies, equal[x, setpart[y]], 
or[not[axch], subclass[intersection[PO, P[cart[x, x]]], image[inverse[S]], 
intersection[PO, complement[setpart[Porous]], intersection[PO, P[cart[x, x]]]], 
P[cart[x, x]]]]], y -> x] // Reverse
```

```plaintext
Out[53]= or[not[axch], not[member[x, V]], 
subclass[intersection[PO, P[cart[x, x]]], image[inverse[S]], 
intersection[PO, complement[setpart[Porous]], intersection[PO, P[cart[x, x]]]], 
P[cart[x, x]]]]] = True
```

```plaintext
In[54]:= (% /. x -> x_) /. Equal -> SetDelayed
```

The following lemma introduces the class `TO` of total orders.
Szpilrajn Theorem. The axiom of choice implies that any partial order can be extended to a total order.
\[\text{In[63]}:=\ \text{Map[equal[V, #]}\&,
\text{SubstTest[class, x, or[not[equal[t, V]], not[member[x, u]], member[x, v]],
\{t \rightarrow \text{SELECT}, u \rightarrow \text{PO}, v \rightarrow \text{image[inverse[S], TO]}\}]}
\]

\[\text{Out[63]}=\ \text{or[not[axch], subclass[PO, image[inverse[S], TO]]]} \Rightarrow \text{True}\]

\[\text{In[64]}:=\ \text{or[not[axch], subclass[PO, image[inverse[S], TO]]]} := \text{True}\]