bijective transforms of partial orders

Johan G. F. Belinfante
2005 January 17

sumary

By renaming the elements of a partially ordered set one obtains another. For reflexive
and transitive relations a more general result holds. The reflexive and transitive prop-
erties are preserved by transformations using arbitrary functions, not just bijections:

\[
\text{In}[2] := \text{imlies}\left[\text{and}[\text{FUNCTION}[x], \text{REFLEXIVE}[y]], \\
\qquad \text{REFLEXIVE}[\text{composite}[\text{inverse}[x], y, x]]\right] \\
\text{Out}[2] = \text{True}
\]

\[
\text{In}[3] := \text{imlies}\left[\text{and}[\text{FUNCTION}[x], \text{TRANSITIVE}[y]], \\
\qquad \text{TRANSITIVE}[\text{composite}[\text{inverse}[x], y, x]]\right] \\
\text{Out}[3] = \text{True}
\]

The antisymmetric property, on the other hand, need not be preserved under such trans-
formations unless \(x\) is one-to-one, and the same is true for partial ordering relations.
The results about partial orders will be derived from corresponding facts about reflex-
ive, antisymmetric and transitive relations. Some lemmas concerning reflexive and
antisymmetric relations are needed to do this. At the end of the notebook a simple
counterexample is presented which shows that one cannot omit the one-to-one hypothe-
sis on \(x\) for the theorems derived about antisymmetric relations and partial orderings.
Lemma about RFX

In this section a needed result is derived for the class \textbf{RFX} of reflexive relations. The main ideas are just that the class \textbf{P[Id]} of identity functions a subclass of the class \textbf{BIJ} of bijections, and the latter is a subclass of the class \textbf{image[INVERSE, FUNS]} of inverse functions. The technique is to use the monotonicity of imaging with the following function:

\begin{verbatim}
In[4]:= abstract[x, image[IMG, cart[map[CROSS, id[x]], RFX]]]
Out[4]= composite[IMG, id[cart[V, RFX]], inverse[FIRST], CROSS, DUP]
\end{verbatim}

This yields two inclusions in opposite directions:

\begin{verbatim}
In[5]:= SubstTest[implies, subclass[u, v],
    subclass[image[w, u], image[w, v]], {u \rightarrow P[Id], v \rightarrow BIJ,
    w \rightarrow composite[IMG, id[cart[V, RFX]], inverse[FIRST], CROSS, DUP]}]
In[6]:= % /. Equal \rightarrow SetDelayed
\end{verbatim}

\begin{verbatim}
In[7]:= SubstTest[implies, subclass[u, v],
    subclass[image[w, u], image[w, v]], {u \rightarrow BIJ, v \rightarrow image[INVERSE, FUNS],
    w \rightarrow composite[IMG, id[cart[V, RFX]], inverse[FIRST], CROSS, DUP]}]
In[8]:= % /. Equal \rightarrow SetDelayed
\end{verbatim}

The two inclusions are combined into an equation:

\begin{verbatim}
In[9]:= SubstTest[and, subclass[u, v], subclass[v, u],
    {u \rightarrow image[IMG, cart[map[CROSS, id[BIJ]], RFX]], v \rightarrow RFX}]
Out[9]= True = equal[RFX, image[IMG, cart[map[CROSS, id[BIJ]], RFX]]]
\end{verbatim}

\begin{verbatim}
In[10]:= image[IMG, cart[map[CROSS, id[BIJ]], RFX]] := RFX
\end{verbatim}

Restrictions of antisymmetric relations

Restrictions of antisymmetric relations are studied in this section. A double-negation lemma prepares for the removal of variables.
\textbf{In [11]}:
\begin{verbatim}
(implies[member[y, z], member[restrict[y, x, x], z]] /\ z \rightarrow \text{ANTISYM}) /\ NotNotTest
\end{verbatim}
\textbf{Out [11]}:
\begin{verbatim}
or[and[member[intersection[x, image[y, x]], V],
    member[intersection[x, image[inverse[y], x]], V],
    subclass[composite[id[x], intersection[y, inverse[y]], id[x]], Id]],
  not[member[y, V]], not[subclass[y, cart[V, V]]],
  not[subclass[intersection[y, inverse[y]], Id]]] = True
\end{verbatim}
\textbf{In [12]}:
\begin{verbatim}
(\% /\ (x \rightarrow \_\_, y \rightarrow \_\_)) /\ \text{Equal} \rightarrow \text{SetDelayed}
\end{verbatim}

The variable $y$ is eliminated first.

\textbf{In [13]}:
\begin{verbatim}
Map[equal[V, \#] \&, SubstTest[class, y,
    implies[member[y, z], member[restrict[y, x, x], z]], z \rightarrow \text{ANTISYM}]] /\ Reverse
\end{verbatim}
\textbf{Out [13]}:
\begin{verbatim}
subclass[image[IMAGE[id[cart[x, x]]], ANTIMY], ANTIMY] = True
\end{verbatim}
\textbf{In [14]}:
\begin{verbatim}
subclass[image[IMAGE[id[cart[\_\_, \_\_]]], ANTIMY], ANTIMY] := True
\end{verbatim}

Reification is used to remove the variable $x$.

\textbf{In [15]}:
\begin{verbatim}
Map[equal[0, \#] \&, SubstTest[reify, x,
    dif[image[IMAGE[id[cart[x, x]]], y], y], y \rightarrow \text{ANTISYM}]] /\ Reverse
\end{verbatim}
\textbf{Out [15]}:
\begin{verbatim}
subclass[image[IMG, cart[image[IMAGE[DUP], image[CART, Id]]], ANTIMY]],
    ANTIMY] = True
\end{verbatim}
\textbf{In [16]}:
\begin{verbatim}
(\% /\ \text{Equal} \rightarrow \text{SetDelayed}
\end{verbatim}

The result can be rewritten in a simpler form:

\textbf{In [17]}:
\begin{verbatim}
Map[subclass[\#, ANTIMY] \&,
    ImageComp[IMG, cross[IDP, Id], cart[image[CART, Id], ANTIMY]]]
\end{verbatim}
\textbf{Out [17]}:
\begin{verbatim}
subclass[image[CAP, cart[image[CART, Id], ANTIMY]], ANTIMY] = True
\end{verbatim}
\textbf{In [18]}:
\begin{verbatim}
(\% /\ \text{Equal} \rightarrow \text{SetDelayed}
\end{verbatim}

The symmetry of $\text{CAP}$ allows one to flip the arguments of $\text{cart}$.

\textbf{In [19]}:
\begin{verbatim}
Map[subclass[\#, ANTIMY] \&,
    ImageComp[CAP, SWAP, cart[image[CART, Id], ANTIMY]]] /\ Reverse
\end{verbatim}
\textbf{Out [19]}:
\begin{verbatim}
subclass[image[CAP, cart[ANTIMY, image[CART, Id]]], ANTIMY] = True
\end{verbatim}
\textbf{In [20]}:
\begin{verbatim}
(\% /\ \text{Equal} \rightarrow \text{SetDelayed}
\end{verbatim}

The reverse inclusion also holds:
The two inclusions are combined into an equation.

There is a companion result with the `cart` arguments interchanged:

lemma about ANTISYM

In this section transforms of antisymmetric relations under bijections are studied. In the first step the `ooopart` wrapper is used to construct a generic bijection.
The `oo part` wrapper is removed and replaced with a pair of `FUNCTION` literals.

A sethood lemma is needed. This rewrite rule appears to be generally useful, and can be made permanent.

Again a double-negation lemma is needed to prepare for the removal of variables:

All the variables can now be removed at once:
In[37]:= \[\text{Map[equal[0, composite[complement[#], id[cart[V, V]]]] \&,}
\]
\[\text{SubstTest[class, pair[pair[x, y], z],}
\]
\[\text{implies[and[member[pair[pair[x, x], z], w], member[x, u], member[y, v]],}
\]
\[\text{member[Image[inverse[z], y], v]],}
\]
\[\text{(u \rightarrow BIJ, v \rightarrow ANTSYM, w \rightarrow CROSS)]] // Reverse}
\]
Out[37]= subclass[Image[IMG, cart[Image[CROSS, id[BIJ]], ANTSYM]], ANTSYM] = True

In[38]:= % /. Equal \[\rightarrow\] SetDelayed

In the reverse direction, a lemma is needed:

In[39]:= \[\text{ImageComp[IMG, cross[IDP, Id], cart[Image[CART, Id], ANTSYM]] // Reverse}
\]
Out[39]= Image[IMG, cart[Image[IMAGE[DUP], image[CART, Id]], ANTSYM]] = ANTSYM

In[40]:= % /. Equal \[\rightarrow\] SetDelayed

The inclusion in the opposite direction follows:

In[41]:= \[\text{SubstTest[implies, subclass[u, v],}
\]
\[\text{subclass[Image[w, u], Image[w, v]], (u \rightarrow P[Id], v \rightarrow BIJ,}
\]
\[\text{w \rightarrow composite[IMG, id[cart[V, ANTSYM]], inverse[FIRST], CROSS, DUP])]
\]
Out[41]= subclass[ANTSYM, Image[IMG, cart[Image[CROSS, id[BIJ]], ANTSYM]]] = True

In[42]:= % /. Equal \[\rightarrow\] SetDelayed

The two inclusions are combined into an equation:

In[43]:= \[\text{SubstTest[and, subclass[u, v], subclass[v, u], (u \rightarrow ANTSYM,}
\]
\[\text{v \rightarrow Image[IMG, cart[Image[CROSS, id[BIJ]], ANTSYM]])] // Reverse}
\]
Out[43]= equal[ANTSYM, Image[IMG, cart[Image[CROSS, id[BIJ]], ANTSYM]]] = True

In[44]:= Image[IMG, cart[Image[CROSS, id[BIJ]], ANTSYM]] := ANTSYM

corollary for partial order

Lemma.

In[45]:= \[\text{ImageComp[IMG, cross[IDP, Id], cart[Image[CART, Id], PO]] // Reverse}
\]
Out[45]= Image[IMG, cart[Image[IMAGE[DUP], Image[CART, Id]], PO]] = PO

In[46]:= % /. Equal \[\rightarrow\] SetDelayed
The reverse inclusion holds for reflexive, antisymmetric and transitive relations, and therefore for partial order relations:

\[
\text{In \[49\]} = \begin{aligned}
\text{SubstTest} & \equiv \text{subclass, image[w, \text{intersection}[x, y, z]],}
\text{intersection}[\text{image}[w, x], \text{image}[w, y], \text{image}[w, z]],
\{w & \mapsto \text{composite[IMG, id[cart[image[CROSS, id[BIJ]], V]], inverse[SECOND]],}
\text{x & \mapsto RFX, y & \mapsto ANTSYM, z & \mapsto TRV]\}
\end{aligned}
\]

\[
\text{Out\[49\]} = \begin{aligned}
\text{subclass}[PO, \text{image}[\text{image[CROSS, id[BIJ]], PO]]] = \text{True}
\end{aligned}
\]

These two inclusions are combined into an equation:

\[
\text{In \[50\]} = \begin{aligned}
\text{SubstTest} & \equiv \text{and, subclass[u, v], subclass[v, u],}
\{u & \mapsto \text{image[IMG, cart[image[CROSS, id[BIJ]], PO]], v & \mapsto PO}\}
\end{aligned}
\]

\[
\text{Out\[50\]} = \begin{aligned}
\text{True} = \text{equal}[PO, \text{image}[\text{image[CROSS, id[BIJ]], PO]]]
\end{aligned}
\]

\[
\text{In\[52\]} = \begin{aligned}
\text{image[IMG, cart[image[CROSS, id[BIJ]], PO]]} = \text{PO}
\end{aligned}
\]

---

a counterexample

In this section, an explicit counterexample is given to show that one cannot replace the \textbf{ONEONE} hypothesis by \textbf{FUNCTION}. The function in the counterexample is the constant function which takes \textbf{0} and \textbf{1} to \textbf{0}.

\[
\text{In \[53\]} = \begin{aligned}
\text{FUNCTION}[\text{cart[successor[set[0]]], set[0]]}
\end{aligned}
\]

\[
\text{Out\[53\]} = \text{True}
\]

The antisymmetric relation in this counterexample is the less-than-or-equal relation for the set of numbers \{0, 1\}.

\[
\text{In \[54\]} = \begin{aligned}
\text{ANTISYMMETRIC[composite[id[successor[set[0]]]], S, id[successor[set[0]]]]}
\end{aligned}
\]

\[
\text{Out\[54\]} = \text{True}
\]
The transformed relation is a cartesian square, which is not antisymmetric.

\[
\text{In}[55]:= \text{In}\{\text{composite}\{\text{inverse}[x], y, x]\}\.\{x \rightarrow \text{cart}\{\text{succ}\{\text{set}[0]\}, \text{set}[0]\},
\text{y} \rightarrow \text{composite}\{\text{id}\{\text{succ}\{\text{set}[0]\}\}, S, \text{id}\{\text{succ}\{\text{set}[0]\}\}\}\}\}
\]

\[
\text{Out}[55]= \text{cart}\{\text{succ}\{\text{set}[0]\}, \text{succ}\{\text{set}[0]\}\}
\]

\[
\text{In}[56]:= \text{cart}\{\text{succ}\{\text{set}[0]\}, \text{succ}\{\text{set}[0]\}\} // \text{ANTISYMMETRIC}
\]

\[
\text{Out}[56]= \text{False}
\]

A cartesian square \text{cart}[x, x] is a partial order only when \text{x} has less than 2 members.

\[
\text{In}[57]:= \text{PARTIALORDER}\{\text{cart}\{x, x\]\} // \text{AssertTest}
\]

\[
\text{Out}[57]= \text{PARTIALORDER}\{\text{cart}\{x, x\]\} = \text{or}\{\text{equal}\{0, x\}, \text{member}\{x, \text{range}\{\text{SINGLETON}\}\}\}\}
\]

\[
\text{In}[58]:= \text{PARTIALORDER}\{\text{cart}\{x_, x_\]\} := \text{or}\{\text{equal}\{0, x\}, \text{member}\{x, \text{range}\{\text{SINGLETON}\}\}\}\}
\]

It follows that the same counterexample works for partial order.

\[
\text{In}[59]:= \text{imply}\{\text{and}\{\text{FUNCTION}\{x\}, \text{PARTIALORDER}\{y\}\}, \text{PARTIALORDER}\{\text{composite}\{\text{inverse}[x], y, x\]\}\}\}\.\{x \rightarrow \text{cart}\{\text{succ}\{\text{set}[0]\}, \text{set}[0]\},
\text{y} \rightarrow \text{composite}\{\text{id}\{\text{succ}\{\text{set}[0]\}\}, S, \text{id}\{\text{succ}\{\text{set}[0]\}\}\}\}\}
\]

\[
\text{Out}[59]= \text{False}
\]