XS and axch

Johan G. F. Belinfante
2004 November 18

```
In[1]:= SetDirectory["i:"]; << goedel63.17b; << tools.m

        :Package Title: goedel63.17b        2004 November 17 at 5:30 p.m.

        It is now:  2004 Nov 18 at 14:31

        Loading Simplification Rules

        TOOLS.M                       Revised 2004 November 17

        weightlimit = 40
```

summary

The function **XS = lambda[x, X[x]]** cannot be one-to-one because the cross-sections of **x** are the same as those of **composite[Id, x]**. If the axiom of choice holds, then any relation can be recovered from the set of its cross-sections:

```
In[2]:= U[X[x]]

Out[2]= composite[x, id[image[V, X[x]]]]
```

Hence the restriction of **XS** to the class of relations is one-to-one in this case. In this notebook, the converse is derived, yielding several statements about **XS** equivalent to **axch**.

easy direction: axch => FUNCTION[composite[id[P[cart[V,V]]], inverse[XS]]

In this section it is shown that if **axch** holds then **composite[XS, id[P[cart[V,V]]]]** is one-to-one. The following technical lemma is needed:

```
In[3]:= SubstTest[implies, equal[0, x],
          member[union[intersection[x, P[cart[V, V]]], singleton[0]],
           range[SINGLETON]], x → complement[SELECT]]

Out[3]= or[
          member[union[intersection[complement[SELECT], P[cart[V, V]]], singleton[0]],
           range[SINGLETON]], not[axch]] == True

In[4]:= % /. Equal → SetDelayed
```

This implies that any relation can be recovered from its cross-sections when **axch** holds.

```
In[5]:= Map[implies[axch, subclass[#, Id]] &,
          composite[BIGCUP, XS, id[P[cart[V, V]]]] // VSNormality]

Out[5]= or[not[axch], subclass[composite[BIGCUP, XS, id[P[cart[V, V]]]], Id]] == True

In[6]:= % /. Equal → SetDelayed
```

The following lemma allows this fact to be exploited.

```
In[7]:= SubstTest[implies,
          and[subclass[composite[x, y], Id], subclass[range[y], domain[x]]],
          FUNCTION[inverse[y]], {x → BIGCUP, y -> composite[XS, id[P[cart[V, V]]]]}]

Out[7]= or[FUNCTION[composite[id[P[cart[V, V]]], inverse[XS]]],
          not[subclass[composite[BIGCUP, XS, id[P[cart[V, V]]]], Id]]] == True

In[8]:= % /. Equal → SetDelayed
```

This yields the easy half of an equivalence theorem.

```
In[9]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], not[implies[p1, p3]],
           {p1 → axch, p2 -> subclass[composite[BIGCUP, XS, id[P[cart[V, V]]]], Id],
            p3 → FUNCTION[composite[id[P[cart[V, V]]], inverse[XS]]]}]]

Out[9]= or[FUNCTION[composite[id[P[cart[V, V]]], inverse[XS]]], not[axch]] == True

In[10]:= % /. Equal → SetDelayed
```

thickness result for RESTRICT

To lay the groundwork for the reverse implication, a thickness result for the relation **inverse[RESTRICT]** is derived in this section. The basic idea is that if **x** admits a cross-section, then so do all its restrictions.

```
In[11]:= Map[implies[member[x, y], equal[V, #]] &,
           SubstTest[class, y, implies[and[member[x, w], member[y, z]], member[y, w]],
            {w → SELECT, z → RS[x]}]] // Reverse

Out[11]= or[equal[0, X[x]], not[member[x, y]], subclass[RS[x], SELECT]] == True

In[12]:= or[equal[0, X[x_]], not[member[x_, y_]], subclass[RS[x_], SELECT]] := True
```

The variable-free statement of this fact is that **SELECT** is invariant under the relation **RESTRICT**, or equivalently, the class **complement[SELECT]** is invariant under the relation **inverse[RESTRICT]**. A lemma helps to produce a clean result.

```
In[13]:= subclass[x, union[cart[complement[domain[x]], V], composite[Id, x]]] //
          AssertTest

Out[13]= subclass[x, union[cart[complement[domain[x]], V], composite[Id, x]]] ==
          subclass[x, cart[V, V]]

In[14]:= subclass[x_, union[cart[complement[domain[x_]], V], composite[Id, x_]]] :=
          subclass[x, cart[V, V]]
```

The following computation resembles one done recently in a notebook about the set-hood of intervals.

```
In[15]:= Map[member[intersection[P[cart[V, V]], #], V] &,
          image[inverse[RESTRICT], singleton[x]] // Renormality]

Out[15]= member[
           intersection[image[inverse[image[inverse[COMPOSE], singleton[x]]], P[Id]],
            P[cart[V, V]]], V] == or[not[member[x, V]], not[subclass[x, cart[V, V]]]]

In[16]:= member[
           intersection[image[inverse[image[inverse[COMPOSE], singleton[x_]]], P[Id]],
            P[cart[V, V]]], V] := or[not[member[x, V]], not[subclass[x, cart[V, V]]]]
```

The main result of this section is this thickness result for **inverse[RESTRICT]**.

```
In[17]:= domain[VERTSECT[composite[id[P[cart[V, V]]], inverse[RESTRICT]]]] // Normality

Out[17]= domain[VERTSECT[composite[id[P[cart[V, V]]], inverse[RESTRICT]]]] ==
          complement[P[cart[V, V]]]

In[18]:= domain[VERTSECT[composite[id[P[cart[V, V]]], inverse[RESTRICT]]]] :=
          complement[P[cart[V, V]]]
```

a strengthening of an earlier theorem

The axiom of choice is equivalent to the statement every set has a cross-section. A stronger statement is that **axch** is equivalent to the statement that the class of sets which admit no cross-sections is a set. In this section this result is strengthened further, replacing sets by relations. The main idea is to use this fact:

```
In[19]:= invariant[composite[id[P[cart[V, V]]], inverse[RESTRICT]],
          dif[P[cart[V, V]], SELECT]]

Out[19]= True

In[20]:= SubstTest[implies, and[subclass[u, v], member[v, V]], member[u, V],
          {u → intersection[P[cart[V, V]],
             complement[lb[image[inverse[COMPOSE], SELECT], P[Id]]]],
           v → dif[P[cart[V, V]], SELECT]}]

Out[20]= or[member[intersection[complement[
             lb[image[inverse[COMPOSE], SELECT], P[Id]]], P[cart[V, V]]], V],
           not[member[intersection[complement[SELECT], P[cart[V, V]]], V]]] == True

In[21]:= % /. Equal → SetDelayed

In[22]:= Map[implies[member[intersection[complement[SELECT], P[cart[V, V]]], V], #] &,
          SubstTest[member, x, domain[IMAGE[y]], {x → dif[P[cart[V, V]], SELECT],
            y → composite[id[P[cart[V, V]]], inverse[RESTRICT]]}]]

Out[22]= or[axch, not[member[intersection[complement[SELECT], P[cart[V, V]]], V]]] ==
          True

In[23]:= % /. Equal → SetDelayed
```

The reverse implication also holds, yielding a statement equivalent to the axiom of choice. It says that **axch** is equivalent to the statement that the class of relations admitting no cross-sections is a set.

```
In[24]:= equiv[member[intersection[complement[SELECT], P[cart[V, V]]], V], axch]

Out[24]= True

In[25]:= member[intersection[complement[SELECT], P[cart[V, V]]], V] := axch
```

main result

If the restriction of **XS** to relations were one-to-one, there could be at most one relation in **complement[SELECT]**:

```
In[26]:= member[0, image[XS, P[cart[V, V]]]] // AssertTest

Out[26]= member[0, image[XS, P[cart[V, V]]]] == not[axch]

In[27]:= member[0, image[XS, P[cart[V, V]]]] := not[axch]
```

Whether or not the axiom of choice holds, there cannot be just a single relation that fails to admit a cross-section. Either there are none, or else very many.

```
In[28]:= member[intersection[complement[SELECT], P[cart[V, V]]],
           range[SINGLETON]] // AssertTest

Out[28]= member[intersection[complement[SELECT], P[cart[V, V]]], range[SINGLETON]] ==
          False

In[29]:= member[intersection[complement[SELECT], P[cart[V, V]]],
           range[SINGLETON]] := False
```

Corollary.

```
In[30]:= member[0, domain[funpart[composite[id[P[cart[V, V]]], inverse[XS]]]]] //
          AssertTest

Out[30]= member[0, domain[funpart[composite[id[P[cart[V, V]]], inverse[XS]]]]] == False

In[31]:= member[0, domain[funpart[composite[id[P[cart[V, V]]], inverse[XS]]]]] := False
```

It follows that if the restriction of the function **XS** to the class of relations is one-to-one, then the axiom of choice holds.

```
In[32]:= SubstTest[implies, and[member[x, domain[z]], FUNCTION[z]],
          member[image[z, singleton[x]], range[SINGLETON]],
          {x → 0, z -> composite[id[P[cart[V, V]]], inverse[XS]]}]

Out[32]= or[axch, not[FUNCTION[composite[id[P[cart[V, V]]], inverse[XS]]]]] == True

In[33]:= % /. Equal → SetDelayed
```

The final result is a new equivalent of the axiom of choice.

```
In[34]:= equiv[FUNCTION[composite[id[P[cart[V, V]]], inverse[XS]]], axch]

Out[34]= True

In[35]:= FUNCTION[composite[id[P[cart[V, V]]], inverse[XS]]] := axch
```

Corollary.

```
In[36]:= SubstTest[equal, composite[y, inverse[y]],
          id[range[y]], y -> composite[id[P[cart[V, V]]], inverse[XS]]]

Out[36]= equal[composite[id[P[cart[V, V]]], inverse[XS], XS, id[P[cart[V, V]]]],
           id[P[cart[V, V]]]] == axch

In[37]:= equal[composite[id[P[cart[V, V]]], inverse[XS], XS, id[P[cart[V, V]]]],
           id[P[cart[V, V]]]] := axch
```

sets with empty domain

Any set whose domain is empty admits the empty set as a cross-section.

```
In[38]:= SubstTest[implies, member[u, v], not[empty[v]], {u → 0, v → X[x]}]

Out[38]= or[not[equal[0, domain[x]]], not[equal[0, X[x]]]] == True

In[39]:= or[not[equal[0, domain[x_]]], not[equal[0, X[x_]]]] := True
```

The class of sets with empty domain is contained in the class **SELECT**.

```
In[40]:= Map[equal[V, #] &,
           SubstTest[class, x, implies[and[member[x, V], equal[0, domain[x]]],
             member[x, y]], y → SELECT]] // Reverse

Out[40]= subclass[P[complement[cart[V, V]]], SELECT] == True

In[41]:= subclass[P[complement[cart[V, V]]], SELECT] := True
```

The axiom of choice is therefore equivalent to the statement that every set with a non-empty domain admits a cross-section.

```
In[42]:= Map[equal[0, U[#]] &, ImageComp[IMAGE[id[cart[V, V]]],
            inverse[IMAGE[id[cart[V, V]]]], complement[SELECT]]] // Reverse

Out[42]= equal[0, domain[U[complement[SELECT]]]] == axch

In[43]:= equal[0, domain[U[complement[SELECT]]]] := axch
```

This result is applied in the next section to obtain yet another statement that is equivalent to the axiom of choice.

another equivalent of axch

Lemma.

```
In[44]:= equal[IMAGE[id[cart[V, V]]],
           union[cart[union[complement[x], P[complement[cart[V, V]]]], singleton[0]],
            composite[IMAGE[id[cart[V, V]]], id[x]]]] // AssertTest

Out[44]= equal[IMAGE[id[cart[V, V]]],
           union[cart[union[complement[x], P[complement[cart[V, V]]]], singleton[0]],
            composite[IMAGE[id[cart[V, V]]], id[x]]]] ==
          equal[0, domain[U[complement[x]]]]

In[45]:= equal[IMAGE[id[cart[V, V]]],
           union[cart[union[complement[x_], P[complement[cart[V, V]]]], singleton[0]],
            composite[IMAGE[id[cart[V, V]]], id[x_]]]] :=
          equal[0, domain[U[complement[x]]]]
```

This yields another statement equivalent to the axiom of choice.

```
In[46]:= Map[equal[IMAGE[id[cart[V, V]]], #] &, composite[BIGCUP, XS] // VSNormality]

Out[46]= equal[composite[BIGCUP, XS], IMAGE[id[cart[V, V]]]] == axch

In[47]:= equal[composite[BIGCUP, XS], IMAGE[id[cart[V, V]]]] := axch
```