axch <=> disjoint[fix[UCHAINS], subvar[inverse[K]]]

Johan G. F. Belinfante
2007 November 13

```
In[1]:= SetDirectory["l:"]; << goedel99.13a; << tools.m

        :Package Title: goedel99.13a         2007 November 13 at 2:20 p.m.

        It is now:  2007 Nov 13 at 15:49

        Loading Simplification Rules

        TOOLS.M                                        Revised 2007 September 19

        weightlimit = 40
```

summary

In this notebook it is shown that the axiom of choice is equivalent to the statement that a collection of sets cannot simultaneously be closed under unions of chains and have the property that to each set in the collection one can add a single point to obtain a bigger set in the collection. This theorem may be viewed as an especially simple case of Zorn's lemma. More general versions of Zorn's lemma could be derived as corollaries of this basic result, but this will not be done here. The theorem derived in this notebook is expressed as a rewrite rule for a simple variable-free equation. This formulation is especially convenient for automated reasoning because pattern matching is facilitated.

domains of cross-sections

Lemma.

```
In[2]:= Map[empty, dif[X[x], image[inverse[IMAGE[FIRST]], set[domain[x]]]] // Normality]

Out[2]= subclass[image[IMAGE[FIRST], X[x]], set[domain[x]]] == True

In[3]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem.

```
In[4]:= equal[image[IMAGE[FIRST], X[x]], intersection[set[domain[x]], image[V, X[x]]]] // assert

Out[4]= True

In[5]:= image[IMAGE[FIRST], X[x_]] := intersection[image[V, X[x]], set[domain[x]]]
```

Corollary.

```
In[6]:= Map[VERTSECT, SubstTest[reify, x, image[t, X[x]], t -> IMAGE[FIRST]]]

Out[6]= composite[IMAGE[IMAGE[FIRST]], XS] == union[
          cart[complement[SELECT], set[0]], composite[SINGLETON, IMAGE[FIRST], id[SELECT]]]

In[7]:= composite[IMAGE[IMAGE[FIRST]], XS] := union[
          cart[complement[SELECT], set[0]], composite[SINGLETON, IMAGE[FIRST], id[SELECT]]]
```

Comment. The class **SELECT** is the class of all sets with cross-sections. The axiom of choice is equivalent to the equation **SELECT = V**.

UNOPS

Lemma.

```
In[8]:= equal[image[DORA, X[x]], cart[set[domain[x]], image[IMAGE[SECOND], X[x]]]] // assert

Out[8]= True

In[9]:= image[DORA, X[x_]] := cart[set[domain[x]], image[IMAGE[SECOND], X[x]]]
```

Theorem.

```
In[10]:= SubstTest[subclass, X[x], intersection[u, v],
           {u → FUNS, v -> image[inverse[DORA], inverse[S]]}] // Reverse

Out[10]= subclass[X[x], UNOPS] ==
          or[equal[0, X[x]], not[member[domain[x], V]], subclass[range[x], domain[x]]]

In[11]:= subclass[X[x_], UNOPS] :=
          or[equal[0, X[x]], not[member[domain[x], V]], subclass[range[x], domain[x]]]
```

Zermelo's theorem about covering operations

Theorem. (Negative form of Zermelo's theorem about covering operations: the domain of a unary operation contained in the covering relation **K** cannot be closed under unions of chains.)

```
In[12]:= and[equal[domain[x], Uchains[domain[x]]],
           member[x, UNOPS], subclass[x, K]] // NotNotTest

Out[12]= and[equal[domain[x], Uchains[domain[x]]], member[x, UNOPS], subclass[x, K]] == False

In[13]:= and[equal[domain[x_], Uchains[domain[x_]]], member[x_, UNOPS], subclass[x_, K]] := False
```

A variable-free formulation of Zermelo's theorem is:

```
In[14]:= intersection[image[inverse[IMAGE[FIRST]], fix[UCHAINS]], UNOPS, P[K]]

Out[14]= 0
```

The following corollary of Zermelo's theeorem is immediate. It will be cleaned up shortly.

```
In[15]:= Map[implies[and[member[t, subvar[inverse[K]]], #],
           equal[X[composite[id[t], K, id[t]]], 0]] &,
         SubstTest[subclass, X[composite[id[t], K, id[t]]], intersection[u, v, w],
          {u → P[K], v -> UNOPS, w -> image[inverse[IMAGE[FIRST]], fix[UCHAINS]]}]] // MapNotNot

Out[15]= or[equal[0, X[composite[id[t], K, id[t]]]],
           not[equal[t, Uchains[intersection[t, image[inverse[K], t]]]]], not[member[t, V]],
           not[subclass[intersection[t, image[K, t]], image[inverse[K], t]]], not[subclass[
            Uchains[intersection[t, image[inverse[K], t]]], image[inverse[K], t]]]] == True

In[16]:= (% /. t → t_) /. Equal → SetDelayed
```

Lemma. A simplification rule needed to clean up the above result.

```
In[17]:= SubstTest[implies, and[equal[Uchains[t], t], equal[s, t]],
           equal[Uchains[s], t], s -> intersection[t, image[inverse[K], t]]] // Reverse

Out[17]= or[equal[t, Uchains[intersection[t, image[inverse[K], t]]]],
           not[equal[t, Uchains[t]]], not[subclass[t, image[inverse[K], t]]]] == True

In[18]:= (% /. t → t_) /. Equal → SetDelayed
```

Lemma. Another simplification rule.

```
In[19]:= SubstTest[implies, and[equal[s, t], equal[t, Uchains[t]],
            subclass[t, image[inverse[K], t]]], subclass[Uchains[s], image[inverse[K], t]],
           s → intersection[t, image[inverse[K], t]]] // Reverse

Out[19]= or[not[equal[t, Uchains[t]]], not[subclass[t, image[inverse[K], t]]], subclass[
            Uchains[intersection[t, image[inverse[K], t]]], image[inverse[K], t]]] == True

In[20]:= (% /. t → t_) /. Equal → SetDelayed
```

Theorem. If a set **x** is subvariant under **inverse[K]** and closed under unions of chains, then the restriction of **K** to **x** cannot have cross-sections.

```
In[21]:= Map[implies[member[x, y], not[#]] &,
           SubstTest[and, implies[p1, p2], implies[p1, p3], implies[p1, p4], implies[p1, p5],
            implies[and[p2, p3, p4, p5], p6], not[implies[p1, p6]],
            {p1 → and[equal[Uchains[x], x], member[x, subvar[inverse[K]]]],
             p2 -> equal[x, Uchains[intersection[x, image[inverse[K], x]]]], p3 -> member[x, V],
             p4 -> subclass[intersection[x, image[K, x]], image[inverse[K], x]], p5 ->
              subclass[Uchains[intersection[x, image[inverse[K], x]]], image[inverse[K], x]],
             p6 -> equal[0, X[composite[id[x], K, id[x]]]]}]] // Reverse

Out[21]= or[equal[0, X[composite[id[x], K, id[x]]]], not[equal[x, Uchains[x]]],
           not[member[x, y]], not[subclass[x, image[inverse[K], x]]]] == True

In[22]:= or[equal[0, X[composite[id[x_], K, id[x_]]]], not[equal[x_, Uchains[x_]]],
           not[member[x_, y_]], not[subclass[x_, image[inverse[K], x_]]]] := True
```

Eliminating the variable **x** yields this variable-free corollary.

```
In[23]:= Map[equal[V, #] &, SubstTest[class, t, implies[member[t, u], not[member[t, v]]],
           {u -> intersection[fix[UCHAINS], subvar[inverse[K]]],
            v -> fix[image[inverse[CART], image[inverse[IMAGE[id[K]]], SELECT]]]}]]

Out[23]= equal[0, intersection[fix[UCHAINS],
            fix[image[inverse[CART], image[inverse[IMAGE[id[K]]], SELECT]]],
            subvar[inverse[K]]]] == True

In[24]:= intersection[fix[UCHAINS],
           fix[image[inverse[CART], image[inverse[IMAGE[id[K]]], SELECT]]],
           subvar[inverse[K]]] := 0
```

Observation.

```
In[25]:= abstract[x, intersection[fix[UCHAINS],
           fix[image[inverse[CART], image[inverse[IMAGE[id[K]]], x]]], subvar[inverse[K]]]]

Out[25]= composite[id[intersection[fix[UCHAINS], subvar[inverse[K]]]],
          inverse[DUP], inverse[CART], inverse[IMAGE[id[K]]]]
```

Corollary. The axiom of choice implies that the classes **fix[UCHAINS]** and **subvar[inverse[K]]** are disjoint.

```
In[26]:= SubstTest[implies, equal[u, v], equal[image[t, u], image[t, v]],
           {t -> composite[id[intersection[fix[UCHAINS], subvar[inverse[K]]]], inverse[DUP],
              inverse[CART], inverse[IMAGE[id[K]]]], u → SELECT, v → V}] // Reverse

Out[26]= or[equal[0, intersection[fix[UCHAINS], subvar[inverse[K]]], not[axch]] == True

In[27]:= % /. Equal → SetDelayed
```

---

converse

Lemma.

```
In[28]:= SubstTest[empty, intersection[t, complement[image[inverse[K], t]]],
          t → intersection[FUNS, P[x]]]

Out[28]= subclass[intersection[FUNS, P[x]],
           image[inverse[K], intersection[FUNS, P[x]]]] == equal[0, X[x]]

In[29]:= subclass[intersection[FUNS, P[x_]],
           image[inverse[K], intersection[FUNS, P[x_]]]] := equal[0, X[x]]
```

Eliminating the variable x yields the desired converse statement: If **fix[UCHAINS]** and **subvar[inverse[K]]** are disjoint, then the axiom of choice is true.

```
In[30]:= Map[equal[V, class[x, #]] &,
           SubstTest[implies, empty[v], not[member[u, v]], {u -> intersection[FUNS, P[x]],
             v → intersection[fix[UCHAINS], subvar[inverse[K]]]}]] // Reverse

Out[30]= or[axch, not[equal[0, intersection[fix[UCHAINS], subvar[inverse[K]]]]]] == True

In[31]:= % /. Equal → SetDelayed
```

The theorem and its converse can be combined into a simple rewrite rule:

```
In[32]:= equiv[equal[0, intersection[fix[UCHAINS], subvar[inverse[K]]]], axch]

Out[32]= True

In[33]:= equal[0, intersection[fix[UCHAINS], subvar[inverse[K]]]] := axch
```