UCLOSURE and IMAGE[IMAGE[x]]

Johan G. F. Belinfante
2003 April 21

```
In[1]:= << goedel52.r53; << tools.m

        :Package Title: goedel52.r53          2003 April 19 at 11:55 p.m

        It is now:  2003 Apr 21 at 13:15

        Loading Simplification Rules

        TOOLS.M                      Revised 2003 April 1

        weightlimit = 40
```

■ an open question

This notebook presents an open question in Gödel's class theory. An equation is derived that holds for all sets. The question is whether this equation remaims true when the set is replaced by a proper class. There are many other questions of this type. It would be desirable to have a general method for extending (when possible) results for sets to proper classes. In particular, one might expect this to be the case when one can "approximate" the class by a very large set. Any class **x** is the union of the class **P[x]** of its subsets. One can regard this as a possible interpretation of the intuitive idea that the class is approximated by large subsets:

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

Out[2]= x
```

Some comments on notation are in order. The class **U[x]** here is the union of all elements of the class **x**.

```
In[7]:= class[z, exists[y, and[member[z, y], member[y, x]]]]

Out[7]= U[x]
```

The power class **P[x]** is the class of all subsets of **x**.

```
In[8]:= member[y, P[x]]

Out[8]= and[member[y, V], subclass[y, x]]
```

Similar approximations hold for various (but not all) constructors. For example, the inverse of a class is the union of the inverses of its subsets:

```
In[5]:= U[image[IMAGE[SWAP], P[y]]]

Out[5]= inverse[y]
```

The function **IMAGE[SWAP]** here takes sets to their inverses:

```
In[6]:=  lambda[x, inverse[x]]

Out[6]=  IMAGE[SWAP]
```

Another example is:

```
In[9]:=  U[image[IMAGE[inverse[S]], P[x]]]

Out[9]=  image[inverse[S], x]
```

Here **S** is the subclass relation, and

```
In[10]:=  lambda[x, image[inverse[S], x]]

Out[10]=  IMAGE[inverse[S]]
```

More generally, the function **IMAGE[x]** is defined by

```
In[17]:=  lambda[y, image[x, y]]

Out[17]=  IMAGE[x]
```

■ background of the problem

On page 94 of Kelley's book on topology it is stated that the inverse images under a function of the open sets for a topology on the range of a function yields a topology on the domain of the function. This is of course only true when the domain of the function is a set. Topologies satisfy two conditions: closure under binary intersection and under arbitrary unions. Thus the set **TOPS** of all topologies can be written as the intersection of two classes:

```
In[11]:=  intersection[CAPclosed, fix[UCLOSURE]]

Out[11]=  TOPS
```

The same issue can be raised for each of these two separate classes. Here we focus on invariance under forming arbitrary unions.

```
In[12]:=  class[t, forall[x, implies[subclass[x, t], member[U[x], t]]]]

Out[12]=  fix[UCLOSURE]
```

The function **UCLOSURE** is

```
In[13]:=  lambda[x, image[BIGCUP, P[x]]]

Out[13]=  UCLOSURE
```

where **BIGCUP** in turn is the function

```
In[14]:=  lambda[x, U[x]]

Out[14]=  BIGCUP
```

The image above has a name:

```
In[22]:= image[BIGCUP, P[x]]

Out[22]= Uclosure[x]
```

The **Uclosure** of a proper class can be approximated by those of of its subsets:

```
In[28]:= U[image[UCLOSURE, P[x]]]

Out[28]= Uclosure[x]
```

We return to the issues raised by Kelley’s comment about the effect of inverse images on a topology. The answer to the question is especially simple for the case of classes that are closed under aribtrary unions. For the class **fix[UCLOSURE]** one can in fact consider general images, not just image inverses of functions:

```
In[15]:= invariant[IMAGE[IMAGE[x]], fix[UCLOSURE]]

Out[15]= True
```

The meaning of **invariant** is:

```
In[16]:= invariant[x, y]

Out[16]= subclass[image[x, y], y]
```

■ derivation of a new rule

Two relations **commute** when their composites are the same in either order:

```
In[18]:= commute[x, y]

Out[18]= equal[composite[x, y], composite[y, x]]
```

The starting point of our derivation is this observation:

```
In[19]:= commute[UCLOSURE, IMAGE[IMAGE[x]]]

Out[19]= True
```

This can be used to prove various new facts; for instance: (this result is not used in the sequel)

```
In[20]:= Assoc[IMAGE[IMAGE[x]], UCLOSURE, POWER] // Reverse

Out[20]= composite[UCLOSURE, IMAGE[IMAGE[x]], POWER] == composite[IMAGE[IMAGE[x]], POWER]

In[21]:= composite[UCLOSURE, IMAGE[IMAGE[x_]], POWER] := composite[IMAGE[IMAGE[x]], POWER]
```

■ A lemma

From the axiom of replacement, it follows:

```
In[23]:= SubstTest[implies, and[FUNCTION[w], member[z, V]], member[image[w, z], V],
             {w -> IMAGE[x], z -> Uclosure[y]}]

Out[23]= or[member[image[IMAGE[x], Uclosure[y]], V], not[member[y, V]]] == True

In[24]:= or[member[image[IMAGE[x_], Uclosure[y_]], V], not[member[y_, V]]] := True
```

■ The open question

```
In[25]:= Map[U, ImageComp[IMAGE[IMAGE[x]], UCLOSURE, singleton[y]]]

Out[25]= intersection[image[V, singleton[y]], Uclosure[image[IMAGE[x], y]]] ==
          intersection[image[V, singleton[y]], image[IMAGE[x], Uclosure[y]]]

In[26]:= Map[implies[member[y, V], equal[Uclosure[image[IMAGE[x], y]], #]] &, %] // Reverse //
          MapNotNot

Out[26]= or[equal[image[IMAGE[x], Uclosure[y]], Uclosure[image[IMAGE[x], y]]],
           not[member[y, V]]] == True

In[27]:= or[equal[image[IMAGE[x_], Uclosure[y_]], Uclosure[image[IMAGE[x_], y_]]],
           not[member[y_, V]]] := True
```

Open Question: Is the same true when **y** is a proper class? Can one leave off the hypothesis **member[y,V]** ?