Uclosure[UNOPS]

Johan G. F. Belinfante
2007 June 24

```
In[1]:= SetDirectory["l:"]; << goedel94.23a; << tools.m

        :Package Title: goedel94.23a      2007 June 23 at 9:55 p.m.

        It is now:  2007 Jun 24 at 8:20

        Loading Simplification Rules

        TOOLS.M                                      Revised 2007 June 23

        weightlimit = 40
```

introduction and summary

The class **UNOPS** is closed under unions of chains.

```
In[2]:= Uchains[UNOPS]

Out[2]= UNOPS
```

The class **UNOPS** is not closed under arbitrary unions. Indeed, the **Uclosure** of **UNOPS** contains every symmetric relation, and is therefore not even a class of functions.

```
In[3]:= SubstTest[implies, subclass[x, y],
          subclass[Uclosure[x], Uclosure[y]], {x → INVOL, y → UNOPS}] // Reverse

Out[3]= subclass[SYM, Uclosure[UNOPS]] == True

In[4]:= subclass[SYM, Uclosure[UNOPS]] := True
```

In this notebook, an explicit formula for the class of all unions of unary operations is derived. If one assumes the axiom of choice, this formula says that **Uclosure[UNOPS]** is the set of all relations whose range is a subset of its domain.

Uclosure[UNOPS]

Lemma. If **t** is a cross-section of a relation **x** whose range is contained in its domain, then **t** is a unary operation.

```
In[5]:= Map[not, SubstTest[and, implies[p1, p3], implies[and[p1, p2], p4],
          implies[and[p1, p3, p4], p5], not[implies[and[p1, p2], p5]],
          {p1 → member[t, X[x]], p2 → subclass[range[x], domain[x]], p3 → FUNCTION[t],
           p4 → subclass[range[t], domain[t]], p5 → member[t, UNOPS]}]] // Reverse

Out[5]= or[member[t, UNOPS], not[equal[domain[t], domain[x]]], not[FUNCTION[t]],
         not[member[t, V]], not[subclass[t, x]], not[subclass[range[x], domain[x]]]] == True

In[6]:= (% /. {t → t_, x → x_}) /. Equal → SetDelayed
```

Theorem. If **range[x]** is contained in **domain[x]**, then the class **X[x]** of cross-sections of **x** is a subclass of the class of unary operations.

```
In[7]:= Map[equal[V, #] &, SubstTest[class, t,
          implies[and[member[t, u], subclass[range[x], domain[x]]], member[t, v]],
          {u → X[x], v → UNOPS}]]

Out[7]= or[not[subclass[range[x], domain[x]]], subclass[X[x], UNOPS]] == True

In[8]:= or[not[subclass[range[x_], domain[x_]]], subclass[X[x_], UNOPS]] := True
```

Eliminating the variable **x** yields a lower bound for **UNOPS**.

```
In[9]:= Map[equal[V, #] &,
         dif[image[inverse[DORA], inverse[S]], image[inverse[XS], P[UNOPS]]] // complement //
          Normality]

Out[9]= subclass[U[image[XS, image[inverse[DORA], inverse[S]]]], UNOPS] == True

In[10]:= % /. Equal → SetDelayed
```

From this lower bound for **UNOPS**, one obtains a lower bound for **Uclosure[UNOPS]**.

```
In[11]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
           {t → BIGCUP, u -> image[XS, image[inverse[DORA], inverse[S]]], v → P[UNOPS]}] // Reverse

Out[11]= subclass[image[BIGCUP, image[XS, image[inverse[DORA], inverse[S]]]], Uclosure[UNOPS]] ==
          True

In[12]:= % /. Equal → SetDelayed
```

Observation: In general, the union of all cross-sections of a class **x** is given by this formula:

```
In[13]:= U[X[x]]

Out[13]= composite[x, id[image[V, X[x]]]]
```

The lower bound for **Uclosure[UNOPS]** derived above can be simplified using this theorem about the sum class of the class of cross-sections. The result involves the class **SELECT** of all sets that admit cross-sections.

```
In[14]:= class[x, not[empty[X[x]]]]

Out[14]= SELECT
```

Lemma. The identity on the class **SELECT** of sets which admit cross-sections commutes with **IMAGE[id[cart[V,V]]]**.

```
In[16]:= SubstTest[empty, symdif[u, v], {u -> composite[id[SELECT], IMAGE[id[cart[V, V]]]],
           v -> composite[IMAGE[id[cart[V, V]]], id[SELECT]]}]

Out[16]= equal[composite[id[SELECT], IMAGE[id[cart[V, V]]]],
           composite[IMAGE[id[cart[V, V]]], id[SELECT]]] == True

In[17]:= composite[IMAGE[id[cart[V, V]]], id[SELECT]] :=
          composite[id[SELECT], IMAGE[id[cart[V, V]]]]
```

Lemma. The class of relations that admit cross-sections and whose ranges are contained in their domains is a lower bound for **Uclosure[UNOPS]**.

```
In[18]:= Map[subclass[image[#, image[inverse[DORA], inverse[S]]], Uclosure[UNOPS]] &,
           composite[BIGCUP, XS] // ReifNormality] // Reverse

Out[18]= subclass[intersection[SELECT, image[inverse[DORA], inverse[S]], P[cart[V, V]]],
           Uclosure[UNOPS]] == True

In[19]:= subclass[intersection[SELECT, image[inverse[DORA], inverse[S]], P[cart[V, V]]],
           Uclosure[UNOPS]] := True
```

In the reverse direction, the class of all relations whose ranges are contained in their domains is an upper bound for **Uclosure[UNOPS]**.

```
In[20]:= SubstTest[implies, subclass[u, v], subclass[Uclosure[u], Uclosure[v]],
           {u → UNOPS, v -> intersection[image[inverse[DORA], inverse[S]], P[cart[V, V]]]}] //
          Reverse

Out[20]= subclass[image[DORA, Uclosure[UNOPS]], inverse[S]] == True

In[21]:= subclass[image[DORA, Uclosure[UNOPS]], inverse[S]] := True
```

Lemma. A general observation.

```
In[22]:= implies[and[subclass[x, z], subclass[intersection[y, z], x]],
           equal[intersection[x, y], intersection[y, z]]] // AssertTest

Out[22]= or[equal[intersection[x, y], intersection[y, z]],
           not[subclass[x, z]], not[subclass[intersection[y, z], x]]] == True

In[23]:= or[equal[intersection[x_, y_], intersection[y_, z_]],
           not[subclass[x_, z_]], not[subclass[intersection[y_, z_], x_]]] := True
```

Theorem.

```
In[24]:= SubstTest[implies, and[subclass[x, z], subclass[intersection[y, z], x]],
          equal[intersection[x, y], intersection[y, z]],
          {x → Uclosure[UNOPS], y → SELECT,
           z → intersection[P[cart[V, V]], image[inverse[DORA], inverse[S]]]}]

Out[24]= True == equal[intersection[SELECT, Uclosure[UNOPS]],
           intersection[SELECT, image[inverse[DORA], inverse[S]], P[cart[V, V]]]]

In[25]:= intersection[SELECT, Uclosure[UNOPS]] :=
          intersection[SELECT, image[inverse[DORA], inverse[S]], P[cart[V, V]]]
```

Corollary. If the axiom of choice holds, then **Uclosure[UNOPS]** is the class of relations whose ranges are contained in their domains.

```
In[27]:= SubstTest[implies, equal[v, SELECT], equal[intersection[v, Uclosure[UNOPS]],
            intersection[v, image[inverse[DORA], inverse[S]], P[cart[V, V]]]], v → V] // Reverse

Out[27]= or[equal[intersection[image[inverse[DORA], inverse[S]], P[cart[V, V]]],
            Uclosure[UNOPS]], not[axch]] == True

In[28]:= or[equal[intersection[image[inverse[DORA], inverse[S]], P[cart[V, V]]],
            Uclosure[UNOPS]], not[axch]] := True
```

This can also be made into a conditional rewrite rule:

```
In[29]:= Uclosure[UNOPS] := intersection[image[inverse[DORA], inverse[S]], P[cart[V, V]]] /; axch
```

The conditional rule could be useful if one simply wants to make the blanket assumption that the axiom of choice holds. If this is the case, one can simply set the flag **axch** equal to **True**.