U[image[IPD,BIJ]] = Q

Johan G. F. Belinfante
2010 February 24

```
In[1]:= SetDirectory["l:"]; << goedel.10feb23a;<< tools.m

        :Package Title: goedel.10feb23a                  2010 February 23 at 11:20 a.m.

        It is now:  2010 Feb 24 at 9:36

        Loading Simplification Rules

        TOOLS.M                                     Revised 2010 January 29

        weightlimit = 40
```

summary

In this notebook a connection between the equipollence relation **Q** and the function **IPD** is derived. Recall that the function **IPD** takes any set **x** to the restriction of **IMAGE[x]** to the class of subsets of **domain[x]**.

```
In[2]:= VERTSECT[reify[x, composite[IMAGE[x], id[P[domain[x]]]]]]

Out[2]= IPD
```

That there should be a connection between **IPD** and **Q** is suggested by the following inclusion which plays a key role in the derivation to be presented below.

```
In[3]:= subclass[composite[IMAGE[oopart[x]], id[P[domain[oopart[x]]]]], Q]

Out[3]= True
```

derivation

The orientation of the following rewrite rule has the advantage that it makes manifest the fact that **U[image[IPD, x]]** is a relation.

Theorem. An equation for **U[image[IPD, x]]**.

```
In[4]:= SubstTest[image, intersection[u, v], x,
          {u -> composite[inverse[FIRST], inverse[S], IMAGE[FIRST]],
           v -> composite[SWAP, inverse[rotate[composite[IMG, SWAP]]]]}] // Reverse

Out[4]= U[image[IPD, x]] ==
         composite[IMG, id[composite[inverse[S], IMAGE[FIRST], id[x]]], inverse[SECOND]]
```

```
In[5]:=  U[image[IPD, x_]] :=
          composite[IMG, id[composite[inverse[S], IMAGE[FIRST], id[x]]], inverse[SECOND]]
```

an inclusion in one direction

Lemma.

```
In[6]:=  dif[DORA, composite[inverse[E], IPD]] // FastReifNormality

Out[6]=  intersection[DORA, composite[complement[inverse[E]], IPD]] == 0

In[7]:=  % /. Equal → SetDelayed
```

Theorem.

```
In[8]:=  SubstTest[empty, dif[u, v], {u -> DORA, v -> composite[inverse[E], IPD]}]

Out[8]=  subclass[DORA, composite[inverse[E], IPD]] == True

In[9]:=  subclass[DORA, composite[inverse[E], IPD]] := True
```

The following inclusion will later be sharpened to an equation.

Corollary. An inclusion in one direction.

```
In[10]:=  SubstTest[implies, subclass[u, v], subclass[image[u, w], image[v, w]],
            {u → DORA, v -> composite[inverse[E], IPD], w → BIJ}] // Reverse

Out[10]=  subclass[Q, composite[IMG,
             id[composite[inverse[S], IMAGE[FIRST], id[BIJ]]], inverse[SECOND]]] == True

In[11]:=  % /. Equal → SetDelayed
```

Serendipity. The following corollary is not used further.

Corollary.

```
In[12]:=  SubstTest[implies, subclass[u, v], subclass[VERTSECT[v], composite[S, VERTSECT[u]]],
            {u → DORA, v -> composite[inverse[E], IPD]}] // Reverse

Out[12]=  subclass[IPD, composite[E, DORA]] == True

In[13]:=  subclass[IPD, composite[E, DORA]] := True
```

reverse inclusion

Lemma.

```
In[14]:= equal[composite[
           intersection[complement[Q], IMAGE[oopart[x]]], id[P[domain[oopart[x]]]]], 0]

Out[14]= True

In[15]:= composite[intersection[complement[Q], IMAGE[oopart[x_]]],
           id[P[domain[oopart[x_]]]]] := 0
```

A variable-free corollary of this will be derived by applying **range[reify[x, -]]** to this result. A series of lemmas is used to simplify the expression obtained when this is done.

Lemma.

```
In[19]:= Map[range, composite[inverse[E], IPD, OOPART]] // FastReifNormality // Reverse

Out[19]= composite[IMAGE[inverse[SINGLETON]], IMAGE[IMG],
           CART, id[composite[inverse[S], IMAGE[SINGLETON], IMAGE[FIRST],
             id[BIJ], inverse[SINGLETON]]], inverse[SECOND], IMAGE[SINGLETON]] ==
          composite[IMG, id[composite[inverse[S], IMAGE[FIRST], id[BIJ]]], inverse[SECOND]]

In[20]:= % /. Equal → SetDelayed
```

Lemma..

```
In[21]:= Map[composite[IMAGE[inverse[SINGLETON]], IMAGE[IMG], CART, #] &,
          composite[id[composite[inverse[S], IMAGE[SINGLETON], IMAGE[FIRST], id[BIJ], inverse[
               SINGLETON]]], inverse[SECOND], IMAGE[SINGLETON]] // FastReifNormality // Reverse]

Out[21]= composite[IMAGE[inverse[SINGLETON]], IMAGE[IMG],
           CART, intersection[composite[inverse[SECOND], IMAGE[SINGLETON]],
            composite[inverse[FIRST], SINGLETON, id[BIJ], inverse[IMAGE[FIRST]], S]]] ==
          composite[IMG, id[composite[inverse[S], IMAGE[FIRST], id[BIJ]]], inverse[SECOND]]

In[22]:= % /. Equal → SetDelayed
```

Lemma.

```
In[23]:= composite[FIRST, intersection[composite[inverse[SECOND], IMAGE[SINGLETON]], composite[
             inverse[rotate[composite[IMAGE[inverse[SINGLETON]], IMAGE[IMG], CART, SWAP]]],
             SINGLETON, id[BIJ], inverse[IMAGE[FIRST]], S]]] // FastReifNormality

Out[23]= composite[FIRST, intersection[composite[inverse[SECOND], IMAGE[SINGLETON]], composite[
             inverse[rotate[composite[IMAGE[inverse[SINGLETON]], IMAGE[IMG], CART, SWAP]]],
             SINGLETON, id[BIJ], inverse[IMAGE[FIRST]], S]]] ==
          composite[IMG, id[composite[inverse[S], IMAGE[FIRST], id[BIJ]]], inverse[SECOND]]

In[24]:= % /. Equal → SetDelayed
```

Lemma. Variable-free inclusion in the opposite direction obtained by applying **range[reify[x, -]]**.

```
In[25]:= Map[empty[range[#]] &, SubstTest[reify, x,
           dif[composite[IMAGE[oopart[x]], id[P[domain[oopart[x]]]]], t], t → Q]]

Out[25]= subclass[composite[IMG,
            id[composite[inverse[S], IMAGE[FIRST], id[BIJ]]], inverse[SECOND]], Q] == True

In[26]:= % /. Equal → SetDelayed
```

The inclusions in both directions can now be combined into an equation.

Main Theorem. A formula for the equipollence relation.

```
In[27]:= SubstTest[and, subclass[u, v], subclass[v, u], {u -> composite[IMG,
             id[composite[inverse[S], IMAGE[FIRST], id[BIJ]]], inverse[SECOND]], v -> Q}]

Out[27]= equal[Q, composite[IMG,
            id[composite[inverse[S], IMAGE[FIRST], id[BIJ]]], inverse[SECOND]]] == True

In[28]:= composite[IMG, id[composite[inverse[S], IMAGE[FIRST], id[BIJ]]], inverse[SECOND]] := Q
```

Restatement.

```
In[29]:= U[image[IPD, BIJ]]

Out[29]= Q
```

---

serendipity

The following related results were also discovered in the course of this work.

Theorem.

```
In[30]:= ImageComp[IPD, IDP, V] // Reverse

Out[30]= image[IPD, P[Id]] == image[IMAGE[DUP], range[POWER]]

In[31]:= image[IPD, P[Id]] := image[IMAGE[DUP], range[POWER]]
```

Theorem.

```
In[32]:= SubstTest[U, image[IPD, x], x → V] // Reverse

Out[32]= U[range[IPD]] ==
          composite[IMG, id[composite[inverse[S], IMAGE[FIRST]]], inverse[SECOND]]

In[33]:= U[range[IPD]] :=
          composite[IMG, id[composite[inverse[S], IMAGE[FIRST]]], inverse[SECOND]]
```

Theorem.

```
In[34]:= Map[range[composite[rotate[flip[#]], inverse[FIRST]]] &,
          intersection[inverse[composite[inverse[SECOND], inverse[S], IMAGE[FIRST]]],
            composite[rotate[composite[IMG, SWAP]]]] // DoubleInverse]

Out[34]= image[IMG, composite[inverse[S], IMAGE[FIRST]]] == V

In[35]:= image[IMG, composite[inverse[S], IMAGE[FIRST]]] := V
```