composites of enumerations

Johan G. F. Belinfante
2010 December 15

```
In[1]:= SetDirectory["l:"]; << goedel.10dec15a

        :Package Title: goedel.10dec15a              2010 December 15 at 8:10 a.m.

        It is now:  2010 Dec 16 at 20:46

        Loading Simplification Rules

        TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

        weightlimit = 40
```

summary

The composite of two enumerations is the enumeration of its range. An application to the enumeration of cardinals is presented.

derivation

Lemma. A temporary simplification rule.

```
In[2]:= SubstTest[implies, and[subcommute[u, w], subcommute[v, w]],
          subcommute[composite[u, v], w], {u → enum[x], v → enum[y], w → E}] // Reverse

Out[2]= subclass[composite[enum[x], enum[y], E], composite[
          inverse[IMAGE[inverse[enum[x]]]], inverse[IMAGE[inverse[enum[y]]]], E]] == True

In[3]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma. Simplification rule.

```
In[4]:= ImageComp[enum[x], id[OMEGA], y] // Reverse

Out[4]= image[enum[x], intersection[OMEGA, y]] == image[enum[x], y]

In[5]:= image[enum[x_], intersection[OMEGA, y_]] := image[enum[x], y]
```

It is not clear how to orient the following equation as a rewrite rule, so it is just kept as a statement of fact.

Main Theorem. The composite of enumerations is the enumeration of its range

```
In[6]:=  SubstTest[equal, t, enum[range[t]], t → composite[enum[x], enum[y]]] // Reverse

Out[6]=  equal[composite[enum[x], enum[y]], enum[image[enum[x], y]]] == True

In[7]:=  equal[composite[enum[x_], enum[y_]], enum[image[enum[x_], y_]]] := True
```

---

an example

In this section a formula for the function ℵ that enumerates the infinite cardinals is derived.

Lemma. The function **ordlist[fix[CARD]]** as the solution of an iteration problem.

```
In[8]:=  SubstTest[iterate, composite[HULL[intersection[x, OMEGA]], SUCC],
           set[A[intersection[OMEGA, x]]], x → fix[CARD]] // Reverse

Out[8]=  iterate[composite[HULL[fix[CARD]], SUCC], set[0]] == ordlist[fix[CARD]]

In[9]:=  iterate[composite[HULL[fix[CARD]], SUCC], set[0]] := ordlist[fix[CARD]]
```

Lemma. Natural numbers are fixed points of **HULL[fix[CARD]]**.

```
In[10]:=  Assoc[HULL[fix[CARD]], id[fix[CARD]], id[omega]]

Out[10]=  composite[HULL[fix[CARD]], id[omega]] == id[omega]

In[11]:=  composite[HULL[fix[CARD]], id[omega]] := id[omega]
```

Theorem. Application of the uniqueness theorem for solutions of iteration problems.

```
In[12]:=  SubstTest[implies, and[equal[composite[w, SUCC], composite[u, w]],
             equal[image[w, set[0]], v]], equal[iterate[u, v], composite[w, id[omega]]]],
            {u → composite[HULL[fix[CARD]], SUCC], v → set[0], w → id[omega]}] // Reverse

Out[12]=  equal[id[omega], ordlist[fix[CARD]]] == True

In[13]:=  ordlist[fix[CARD]] := id[omega]
```

Theorem. Formula for the image occurring in an instance of the formula in the main theorem.

```
In[14]:=  ImageComp[enum[fix[CARD]], id[omega], V] // Reverse

Out[14]=  image[enum[fix[CARD]], omega] == omega

In[15]:=  image[enum[fix[CARD]], omega] := omega
```

Theorem. (Serendipity.)

```
In[16]:=  SubstTest[enum, ord[x], x → omega] // Reverse

Out[16]=  enum[omega] == id[omega]

In[17]:=  enum[omega] := id[omega]
```

Application of the main theorem of the preceding section yields the following result.

Theorem. The function ℵ is the composite of the function enumerating all cardinals and **ordplus[ω]**.

In[18]:=
```
SubstTest[equal, composite[enum[x], enum[y]],
  enum[image[enum[x], y]], {x → fix[CARD], y → complement[omega]}] // Reverse
```

Out[18]=
```
equal[ALEPH, composite[enum[fix[CARD]], ordplus[omega]]] == True
```

In[19]:=
```
composite[enum[fix[CARD]], ordplus[omega]] := ALEPH
```