enumerating all ordinals in a given class: part 9. equipollence

Johan G. F. Belinfante
2010 October 15

```
In[1]:= SetDirectory["l:"]; << goedel.10oct14b

:Package Title: goedel.10oct14b            2010 October 14 at 7:25 p.m.

It is now:  2010 Oct 15 at 3:33

Loading Simplification Rules

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

weightlimit = 40
```

summary

This ninth notebook on enumerating ordinals is concerned with applications to cardinal number theory. The axiom of choice is not assumed here. The cardinality **card[x]** of a class **x** is by definition the intersection of all ordinals that are equipollent to **x**. Thus **card[x] = V** when **x** is a proper class, and also when **x** is a set that is not equipollent to any ordinal. The domain of the cardinality function **CARD = λ[x, card[x]]** is the class **image[Q, Ω]** of all sets that are equipollent to some ordinal. It is shown in this notebook that the every subset of the class **Ω** of ordinals is equipollent to an ordinal. That is, the power class of **Ω** is contained in the domain of the cardinality function **CARD**. From this fact it is deduced that the domain of **CARD** is hereditary in the sense that if a set is equipollent to an ordinal, then so is every subset of that set.

sethood for enum[x]

Most of the results in this notebook are of interest only for sets and not proper classes. In this section some basic sethood results concerning the function **enum[x]** are derived. If **x** is a set, then so is **enum[x]**.

Theorem. The function **enum[x]** is a set if and only if the intersection of **x** with the class **Ω** of ordinals is a set.

```
In[2]:= SubstTest[member, range[oopart[t]], V, t → enum[x]]

Out[2]= member[enum[x], V] == member[intersection[OMEGA, x], V]

In[3]:= member[enum[x_], V] := member[intersection[OMEGA, x], V]
```

Corollary. The same is true for the domain of **enum[x]**.

```
In[7]:= SubstTest[member, domain[funpart[t]], V, t → enum[x]] // Reverse

Out[7]= member[domain[enum[x]], V] == member[intersection[OMEGA, x], V]

In[8]:= member[domain[enum[x_]], V] := member[intersection[OMEGA, x], V]
```

A stronger result holds because **domain[enum[x]]** is a full subclass of **Ω**.

Theorem.

```
In[10]:= SubstTest[and, full[t], member[t, P[OMEGA]], t → domain[enum[x]]]

Out[10]= member[domain[enum[x]], OMEGA] == member[intersection[OMEGA, x], V]

In[11]:= member[domain[enum[x_]], OMEGA] := member[intersection[OMEGA, x], V]
```

Lemma.

```
In[14]:= SubstTest[and, full[t], subclass[t, OMEGA], t → domain[enum[x]]]

Out[14]= or[equal[OMEGA, domain[enum[x]]], member[intersection[OMEGA, x], V]] == True

In[15]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[20]:= SubstTest[implies, and[equal[u, v], member[v, V]],
           member[u, V], {u → OMEGA, v → domain[enum[x]]}] // Reverse

Out[20]= or[not[equal[OMEGA, domain[enum[x]]]], not[member[intersection[OMEGA, x], V]]] == True

In[21]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. The domain of **enum[x]** is **Ω** if and only if the intersection of **x** with the class **Ω** of ordinals is a proper class.

```
In[22]:= equiv[equal[OMEGA, domain[enum[x]]], not[member[intersection[OMEGA, x], V]]]

Out[22]= True

In[24]:= equal[OMEGA, domain[enum[x_]]] := not[member[intersection[OMEGA, x], V]]
```

Theorem. The function **enum[x]** is one of the partial enumerations if and only if $\Omega \cap x$ is a set.

```
In[27]:= SubstTest[and, member[domain[w], OMEGA],
          subclass[w, composite[HULL[intersection[OMEGA, x]], IMAGE[w]]], w → enum[x]]

Out[27]= member[enum[x], partenum[x]] == member[intersection[OMEGA, x], V]

In[28]:= member[enum[x_], partenum[x_]] := member[intersection[OMEGA, x], V]
```

equipollence results

Lemma.

```
In[30]:= SubstTest[implies, member[t, BIJ],
           member[pair[domain[t], range[t]], Q], t → enum[x]] // Reverse

Out[30]= or[member[pair[domain[enum[x]], intersection[OMEGA, x]], Q],
           not[member[intersection[OMEGA, x], V]]] == True

In[31]:= (% /. x → x_) /. Equal → SetDelayed
```

The converse also holds.

Theorem. The domain of **enum[x]** is equipollent to $\Omega \bigcap x$ if and only if $\Omega \bigcap x$ is a set.

```
In[32]:= equiv[member[pair[domain[enum[x]], intersection[OMEGA, x]], Q],
          member[intersection[OMEGA, x], V]]

Out[32]= True

In[34]:= member[pair[domain[enum[x_]], intersection[OMEGA, x_]], Q] :=
          member[intersection[OMEGA, x], V]
```

Lemma.

```
In[37]:= SubstTest[implies,
           and[member[pair[u, v], composite[Id, y]], member[u, z]], member[v, image[y, z]]],
           {u -> domain[enum[x]], v -> range[enum[x]], y → Q, z → OMEGA}] // Reverse

Out[37]= or[member[intersection[OMEGA, x], image[Q, OMEGA]],
           not[member[intersection[OMEGA, x], V]]] == True

In[38]:= (% /. x → x_) /. Equal → SetDelayed
```

Again, the converse is also true.

Theorem.

```
In[39]:= equiv[member[intersection[OMEGA, x], image[Q, OMEGA]],
          member[intersection[OMEGA, x], V]]

Out[39]= True

In[41]:= member[intersection[OMEGA, x_], image[Q, OMEGA]] := member[intersection[OMEGA, x], V]
```

Corollary. Any subset of Ω is equipollent to an ordinal.

```
In[43]:= Map[implies[member[x, y], #] &,
           SubstTest[implies, and[equal[x, intersection[OMEGA, t]], member[x, V]],
            member[x, image[Q, OMEGA]], t → x]] // Reverse

Out[43]= or[member[x, image[Q, OMEGA]], not[member[x, y]], not[subclass[x, OMEGA]]] == True

In[44]:= or[member[x_, image[Q, OMEGA]], not[member[x_, y_]], not[subclass[x_, OMEGA]]] := True
```

The variables can be eliminated as follows.

Theorem. The power class of **Ω** is a subclass of the domain of **CARD**.

```
In[46]:= Map[equal[V, #] &, SubstTest[class, x,
           implies[member[x, u], member[x, v]], {u → P[OMEGA], v → image[Q, OMEGA]}]]

Out[46]= subclass[P[OMEGA], image[Q, OMEGA]] == True

In[47]:= subclass[P[OMEGA], image[Q, OMEGA]] := True
```

Lemma. (This is subsumed by the next theorem.)

```
In[49]:= SubstTest[implies, subclass[u, v],
           subclass[image[t, u], image[t, v]], {t → Q, u → OMEGA, v → P[OMEGA]}] // Reverse

Out[49]= subclass[OMEGA, image[Q, P[OMEGA]]] == True

In[50]:= (% /. Equal → SetDelayed)
```

Theorem. A simplification rule for **image[Q, P[Ω]]**.

```
In[51]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u → image[Q, P[OMEGA]], v → image[Q, OMEGA]}]

Out[51]= equal[image[Q, OMEGA], image[Q, P[OMEGA]]] == True

In[53]:= image[Q, P[OMEGA]] := image[Q, OMEGA]
```

From this and existing rewrite rules it now follows automatically that **image[Q, Ω]** is hereditary.

```
In[54]:= image[inverse[S], image[Q, OMEGA]]

Out[54]= image[Q, OMEGA]
```

The following corollary is a version of this result with variables restored.

Corollary. If $\mathbf{x} \subset \mathbf{y}$ and if **y** is equipollent to an ordinal, then so is **x**.

```
In[56]:= SubstTest[implies, and[subclass[x, y], member[y, t]],
           member[x, image[inverse[S], t]], t → image[Q, OMEGA]] // Reverse

Out[56]= or[member[x, image[Q, OMEGA]],
           not[member[y, image[Q, OMEGA]]], not[subclass[x, y]]] == True
```

```
In[57]:= or[member[x_, image[Q, OMEGA]],
           not[member[y_, image[Q, OMEGA]]], not[subclass[x_, y_]]] := True
```