equipollence for ordinals

Johan G. F. Belinfante
2010 November 6

```
In[1]:= SetDirectory["l:"]; << goedel.10nov05b

        :Package Title: goedel.10nov05b              2010 November 5 at 1:50 p.m.

        It is now:  2010 Nov 6 at 10:8

        Loading Simplification Rules

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

        weightlimit = 40
```

summary

The class of ordinals equipollent to a given ordinal is a set. This fact is derived by using the fact that distinct ordinals can not have similar two-sided restrictions of the subset relation. As a corollary, it follows that the class of cardinals is not a set.

application of rigidity

The part of the derivation that depends on **S** and on **Ω** is a corollary of a rigidity theorem about ordinals. The rest of the derivation can be done without reference to these special classes.

Theorem.

```
In[2]:= SubstTest[subclass, composite[inverse[t], t], Id,
          t → composite[SIMILAR, IMAGE[id[S]], CART, DUP, id[OMEGA]]]

Out[2]= FUNCTION[composite[id[OMEGA], inverse[DUP],
            inverse[CART], inverse[IMAGE[id[S]]], SIMILAR]] == True

In[3]:= FUNCTION[composite[id[OMEGA], inverse[DUP],
            inverse[CART], inverse[IMAGE[id[S]]], SIMILAR]] := True
```

Theorem.

```
In[4]:= SubstTest[implies, and[FUNCTION[t], member[v, V]],
          member[image[t, v], V], {t → composite[id[OMEGA], inverse[DUP], inverse[CART],
             inverse[IMAGE[id[S]]], SIMILAR], v → P[cartsq[ord[x]]]}] // Reverse

Out[4]= member[intersection[OMEGA, fix[image[inverse[CART],
             image[inverse[IMAGE[id[S]]], image[SIMILAR, P[cart[ord[x], ord[x]]]]]]]], V] == True
```

```
In[5]:= member[intersection[OMEGA, fix[image[inverse[CART], image[
            inverse[IMAGE[id[S]]], image[SIMILAR, P[cart[ord[x_], ord[x_]]]]]]]], V] := True
```

This says that the class of ordinals to which the restriction of **S** is similar to a relation on **ord[x]** is a set.

---

the rest of the story

The strategy will be to show that if **v** is equipollent to **x,** then for any relation **s**, there is a relation **r ⊂ x × x** that is similar to the restriction of **s** to **v**. The idea is to take any **t ∈ bij[v, x]** and show that **r = t ∘ s ∘ inverse[t]** does the job.

The following temporary abbreviation saves some writing.

```
In[6]:= rs[s_, x_] := composite[id[x], s, id[x]]
```

Lemma.

```
In[7]:= Map[implies[equal[x, range[t]], #] &,
          Map[implies[equal[v, domain[t]], #] &, SubstTest[implies, and[member[t, BIJ],
               subclass[w, cartsq[domain[t]]]], member[pair[w, composite[t, w, inverse[t]]],
               SIMILAR], w → rs[s, v]]] // Reverse // MapNotNot] // MapNotNot

Out[7]= or[member[pair[composite[id[v], s, id[v]], composite[t, id[v], s, id[v], inverse[t]]],
           SIMILAR], not[member[t, bij[v, x]]],
          not[subclass[intersection[v, image[s, v]], domain[t]]],
          not[subclass[intersection[v, image[inverse[s], v]], domain[t]]]] == True

In[8]:= (% /. {s → s_, t → t_, v → v_}) /. Equal → SetDelayed
```

This lemma can be cleaned up.

Theorem. If **t ∈ bij[v, x]**, then **rs[s,v]** is similar to **t ∘ rs[s,v] ∘ inverse[t]**.

```
In[9]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], implies[p2, p4],
           implies[and[p1, p3, p4], p5], not[implies[p1, p5]], {p1 → member[t, bij[v, x]],
            p2 → equal[v, domain[t]], p3 -> subclass[intersection[v, image[s, v]], domain[t]],
            p4 -> subclass[intersection[v, image[inverse[s], v]], domain[t]],
            p5 -> member[pair[composite[id[v], s, id[v]],
               composite[t, id[v], s, id[v], inverse[t]]], SIMILAR]}]] // Reverse

Out[9]= or[member[pair[composite[id[v], s, id[v]], composite[t, id[v], s, id[v], inverse[t]]],
           SIMILAR], not[member[t, bij[v, x]]]] == True

In[10]:= (% /. {s → s_, t → t_, v → v_, x → x_}) /. Equal → SetDelayed
```

Lemma.

```
In[11]:= Map[implies[member[v, w], #] &, SubstTest[implies,
            and[member[pair[r, t], inverse[w]], member[r, z]], member[t, image[inverse[w], z]],
            {t → rs[s, v], w → SIMILAR, z → P[cartsq[x]]}]] // Reverse

Out[11]= or[member[composite[id[v], s, id[v]], image[SIMILAR, P[cart[x, x]]]],
          not[member[v, w]], not[member[pair[composite[id[v], s, id[v]], r], SIMILAR]],
          not[subclass[r, cart[x, x]]]] == True

In[12]:= (% /. {r → r_, s → s_, v → v_, w → w_, x → x_}) /. Equal → SetDelayed
```

In the following theorem, two proof steps are omitted to speed up execution: **implies[p1, p3]** and **implies[p1, p4]**.

Theorem. If **t ∈ bij[v, x]**, then **rs[s, v]** is similar to a relation contained in the cartesian square of **x**.

```
In[13]:= Map[not, SubstTest[and, implies[and[p0, p1], p2], implies[and[p0, p4], p5],
             implies[and[p2, p3, p5], p6], not[implies[and[p0, p1], p6]],
             {p0 → equal[r, composite[t, id[v], s, id[v], inverse[t]]], p1 → member[t, bij[v, x]],
              p2 -> member[pair[composite[id[v], s, id[v]], r], SIMILAR],
              p3 → member[v, V], p4 → equal[x, range[t]], p5 → subclass[r, cartsq[x]],
              p6 -> member[composite[id[v], s, id[v]], image[SIMILAR, P[cart[x, x]]]]}] /.
            r → composite[t, rs[s, v], inverse[t]]] // Reverse

Out[13]= or[member[composite[id[v], s, id[v]], image[SIMILAR, P[cart[x, x]]]],
          not[member[t, bij[v, x]]]] == True

In[14]:= or[member[composite[id[v_], s_, id[v_]], image[SIMILAR, P[cart[x_, x_]]]],
          not[member[t_, bij[v_, x_]]]] := True
```

Theorem. (Eliminate variables.)

```
In[16]:= Map[empty[range[complement[#]]] &,
          SubstTest[class, pair[t, v], implies[member[t, bij[setpart[v], x]],
            member[setpart[v], w]], w -> fix[image[inverse[CART],
              image[inverse[IMAGE[id[s]]], image[SIMILAR, P[cart[x, x]]]]]]]]

Out[16]= subclass[image[Q, set[x]], fix[image[inverse[CART],
            image[inverse[IMAGE[id[s]]], image[SIMILAR, P[cart[x, x]]]]]]] == True

In[17]:= subclass[image[Q, set[x_]], fix[image[inverse[CART],
             image[inverse[IMAGE[id[s_]]], image[SIMILAR, P[cart[x_, x_]]]]]]] := True
```

Main Theorem. The class of ordinals equipollent to a given ordinal is a set.

```
In[18]:= SubstTest[implies, and[subclass[u, v], member[v, V]],
          member[u, V], {u → intersection[OMEGA, image[Q, set[ord[x]]]],
           v → intersection[OMEGA, fix[image[inverse[CART],
               image[inverse[IMAGE[id[S]]], image[SIMILAR, P[cartsq[ord[x]]]]]]]]}] // Reverse

Out[18]= member[intersection[OMEGA, image[Q, set[ord[x]]]], V] == True

In[19]:= member[intersection[OMEGA, image[Q, set[ord[x_]]]], V] := True
```

Corollary. (Remove **ord** wrapper.)

```
In[20]:= SubstTest[implies, equal[x, ord[t]],
           member[intersection[OMEGA, image[Q, set[x]]], V], t → x] // Reverse

Out[20]= or[member[intersection[OMEGA, image[Q, set[x]]], V], not[member[x, OMEGA]]] == True

In[21]:= or[member[intersection[OMEGA, image[Q, set[x_]]], V], not[member[x_, OMEGA]]] := True
```

Theorem. A variable-free reformulation.

```
In[22]:= Map[equal[V, domain[#]] &, SubstTest[reify, x,
           image[V, set[intersection[w, image[q, set[ord[x]]]]]], {w → OMEGA, q → Q}]]

Out[22]= subclass[OMEGA, domain[VERTSECT[composite[id[OMEGA], Q]]]] == True

In[23]:= subclass[OMEGA, domain[VERTSECT[composite[id[OMEGA], Q]]]] := True
```

applications to cardinal number theory

Some applications to cardinal number theory are derived in this section. These results are independent of both the axiom of regularity and of the axiom of choice.

Lemma.

```
In[24]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u → fix[CARD], v → OMEGA, w -> domain[VERTSECT[composite[id[OMEGA], Q]]]}] // Reverse

Out[24]= subclass[fix[CARD], domain[VERTSECT[composite[id[OMEGA], Q]]]] == True

In[25]:= subclass[fix[CARD], domain[VERTSECT[composite[id[OMEGA], Q]]]] := True
```

Theorem. The class of cardinal numbers (= initial ordinals) is not a set.

```
In[26]:= SubstTest[member, y,
           P[domain[VERTSECT[composite[id[OMEGA], Q]]]], y → fix[CARD]] // Reverse

Out[26]= member[fix[CARD], V] == False

In[27]:= member[fix[CARD], V] := False
```

Corollary. A simplification rule.

```
In[28]:= SubstTest[implies, and[subclass[t, OMEGA], not[member[t, V]]],
           equal[U[t], OMEGA], t → fix[CARD]] // Reverse

Out[28]= equal[OMEGA, U[fix[CARD]]] == True

In[29]:= U[fix[CARD]] := OMEGA
```

Observation.

```
In[30]:= range[ALEPH]

Out[30]= intersection[complement[omega], fix[CARD]]
```

Lemma. The class of alephs is not a set.

```
In[31]:= SubstTest[member, union[u, v], V, {u → omega, v → dif[fix[CARD], omega]}]

Out[31]= member[intersection[complement[omega], fix[CARD]], V] == False

In[32]:= % /. Equal → SetDelayed
```

Theorem. The domain of the ℵ function is Ω.

```
In[33]:= equal[OMEGA, domain[ALEPH]]

Out[33]= True

In[34]:= domain[ALEPH] := OMEGA
```

Corollary. Recursion relation for the ℵ function.

```
In[35]:= SubstTest[composite, HULL[intersection[x, OMEGA]],
           IMAGE[enum[x]], id[domain[enum[x]]], x → dif[fix[CARD], omega]] // Reverse

Out[35]= composite[HULL[intersection[complement[omega], fix[CARD]]],
           IMAGE[ALEPH], id[OMEGA]] == ALEPH

In[36]:= composite[HULL[intersection[complement[omega], fix[CARD]]],
           IMAGE[ALEPH], id[OMEGA]] := ALEPH
```