a counterexample

Johan G. F. Belinfante
2005 December 5


```
In[1]:= SetDirectory["l:"]; << goedel76.03a; << tools.m

:Package Title: goedel76.03a             2005 December 3 at 5:20 p.m.

It is now:  2005 Dec 5 at 14:22

Loading Simplification Rules

TOOLS.M                      Revised 2005 October 25

weightlimit = 40
```

summary

The class **OMEGA** of ordinals can be characterized in various ways, for example as the class of regular sets which are full and whose elements are full.

```
In[2]:= intersection[REGULAR, FULL, P[FULL]]

Out[2]= OMEGA
```

If the axiom of regularity is assumed, then **REGULAR = V** can be omitted. Isbell noted that even without assuming the axiom of regularity, one can characterize ordinals as follows:

```
In[3]:= class[x, equal[intersection[FULL, P[x]], succ[x]]]

Out[3]= OMEGA
```

The following equation for the class of ordinals expresses this:

```
In[4]:= fix[composite[inverse[SUCC], IMAGE[id[FULL]], POWER]]

Out[4]= OMEGA
```

From Isbell's theorem 5, the following corollary was deduced on 1998 April 2. (See Corollary **ISB5-HER**. in the **ON-4** group of theorems proved using **Otter**.)

```
In[5]:= implies[member[x, OMEGA], equal[x, intersection[FULL, image[inverse[S], x]]]]

Out[5]= True
```

A variable-free version of this is:

```
In[6]:= Map[equal[V, #] &, SubstTest[class, x, implies[member[x, u],
            equal[x, intersection[v, image[inverse[S], x]]]], {u → OMEGA, v → FULL}]] // Reverse

Out[6]= subclass[OMEGA, invar[composite[id[FULL], inverse[S]]]] == True

In[7]:= subclass[OMEGA, invar[composite[id[FULL], inverse[S]]]] := True
```

At that time an attempt was made to find a counterexample for the reverse implication, but no example was found at that time, leading to a weak conjecture that one might be able to use this property to define the class of ordinals.

```
In[8]:= class[x, equal[x, intersection[v, image[inverse[S], x]]]] /. {u → OMEGA, v → FULL}

Out[8]= intersection[invar[composite[id[FULL], inverse[S]]], P[FULL]]
```

In this notebook a counterexample is constructed.

lemma

An explicit formula for the power set of a pairset can be derived as follows:

```
In[9]:= SubstTest[image, CUP, cart[P[u], P[v]], {u → set[x], v → set[y]}] // Reverse

Out[9]= P[set[x, y]] == set[0, set[x], set[y], set[x, y]]

In[10]:= P[set[x_, y_]] := set[0, set[x], set[y], set[x, y]]
```

the counterexample

The following set will be shown to satisfy the equation **x = intersection[FULL, image[inverse[S], x]]**, but it is not an ordinal.

```
In[11]:= example = union[set[union[set[0], succ[set[set[0]]]]], succ[succ[set[0]]]]

Out[11]= union[set[union[set[0], succ[set[set[0]]]]], succ[succ[set[0]]]]
```

This set is not an ordinal. In fact it is not even full:

```
In[12]:= full[example] // AssertTest

Out[12]= equal[set[set[0]], union[set[0], succ[set[set[0]]]]] == False

In[13]:= equal[set[set[0]], union[set[0], succ[set[set[0]]]]] := False
```

lemmas

```
In[14]:= subclass[intersection[FULL, P[complement[set[set[0]]]],
            set[set[set[set[0]]], set[0, set[set[0]]]]]], succ[set[0]]] // AssertTest

Out[14]= subclass[intersection[FULL, P[complement[set[set[0]]]],
            set[set[set[set[0]]], set[0, set[set[0]]]]]], succ[set[0]]] == True

In[15]:= subclass[intersection[FULL, P[complement[set[set[0]]]],
            set[set[set[set[0]]], set[0, set[set[0]]]]]], succ[set[0]]] := True
```

Corollary.

```
In[16]:= subclass[intersection[FULL,
            P[union[intersection[complement[set[set[0]]], succ[set[set[0]]]], set[0]]]],
          succ[set[0]]] // AssertTest

Out[16]= subclass[intersection[FULL,
            P[union[intersection[complement[set[set[0]]], succ[set[set[0]]]], set[0]]]],
          succ[set[0]]] == True

In[17]:= subclass[intersection[FULL,
            P[union[intersection[complement[set[set[0]]], succ[set[set[0]]]], set[0]]]], succ[
            set[0]]] := True

In[18]:= intersection[complement[set[0]], succ[set[set[0]]]] // Normality

Out[18]= intersection[complement[set[0]], succ[set[set[0]]]] == succ[set[set[0]]]

In[19]:= intersection[complement[set[0]], succ[set[set[0]]]] := succ[set[set[0]]]

In[25]:= equal[0, intersection[FULL, succ[set[set[set[0]]]]]] // AssertTest

Out[25]= equal[0, intersection[FULL, succ[set[set[set[0]]]]]] == True

In[26]:= intersection[FULL, succ[set[set[set[0]]]]] := 0

In[20]:= Map[equal[0, #] &, dif[P[succ[set[set[0]]]],
            union[set[0, succ[set[set[0]]]], succ[set[set[set[0]]]]]] // Renormality]

Out[20]= subclass[P[succ[set[set[0]]]],
          union[set[0, succ[set[set[0]]]], succ[set[set[set[0]]]]]] == True

In[21]:= subclass[P[succ[set[set[0]]]],
          union[set[0, succ[set[set[0]]]], succ[set[set[set[0]]]]]] := True

In[22]:= equal[intersection[complement[set[0]], P[succ[set[set[0]]]]],
          union[set[succ[set[set[0]]]], succ[set[set[set[0]]]]]] // AssertTest

Out[22]= equal[intersection[complement[set[0]], P[succ[set[set[0]]]]],
          union[set[succ[set[set[0]]]], succ[set[set[set[0]]]]]] == True
```

```
In[23]:= intersection[complement[set[0]], P[succ[set[set[0]]]]] :=
          union[set[succ[set[set[0]]]], succ[set[set[set[0]]]]]

In[27]:= subclass[intersection[FULL, P[succ[set[set[0]]]]], set[0]] // AssertTest

Out[27]= subclass[intersection[FULL, P[succ[set[set[0]]]]], set[0]] == True

In[28]:= subclass[intersection[FULL, P[succ[set[set[0]]]]], set[0]] := True
```

The following step is time consuming:

```
In[29]:= Map[equal[0, #] &, dif[intersection[FULL, P[union[set[0], succ[set[set[0]]]]]],
            union[set[union[set[0], succ[set[set[0]]]]], succ[succ[set[0]]]]] // Renormality]

Out[29]= subclass[intersection[FULL, P[union[set[0], succ[set[set[0]]]]]],
           union[set[union[set[0], succ[set[set[0]]]]], succ[succ[set[0]]]]] == True

In[30]:= subclass[intersection[FULL, P[union[set[0], succ[set[set[0]]]]]],
           union[set[union[set[0], succ[set[set[0]]]]], succ[succ[set[0]]]]] := True

In[31]:= equal[intersection[FULL, P[union[set[0], succ[set[set[0]]]]]],
           union[set[union[set[0], succ[set[set[0]]]]], succ[succ[set[0]]]]] // AssertTest

Out[31]= equal[intersection[FULL, P[union[set[0], succ[set[set[0]]]]]],
           union[set[union[set[0], succ[set[set[0]]]]], succ[succ[set[0]]]]] == True

In[32]:= equal[intersection[FULL, P[union[set[0], succ[set[set[0]]]]]],
           union[set[union[set[0], succ[set[set[0]]]]], succ[succ[set[0]]]]] := True
```

The counterexample:

```
In[33]:= equal[example, intersection[FULL, image[inverse[S], example]]]

Out[33]= True
```

---

variable-free

The following inclusion holds:

```
In[34]:= subclass[OMEGA, intersection[invar[composite[id[FULL], inverse[S]]], P[FULL]]]

Out[34]= True
```

The reverse inclusion does not hold. In fact, the counterexample constructed in this notebook implies the following stronger result:

```
In[35]:= Map[not, SubstTest[implies, and[member[x, y], subclass[y, z]], member[x, z],
           {x → example,
            y → intersection[P[FULL], invar[composite[id[FULL], inverse[S]]]], z → FULL}]]

Out[35]= subclass[intersection[invar[composite[id[FULL], inverse[S]]], P[FULL]], FULL] == False
```

```
In[36]:= subclass[intersection[invar[composite[id[FULL], inverse[S]]], P[FULL]], FULL] := False
```