Zermelo's ordinals are full.

Johan G. F. Belinfante
2010 June 9

```
In[1]:= SetDirectory["l:"]; << goedel.10jun09a;<< tools.m

:Package Title: goedel.10jun09a              2010 June 9 at 1:45 p.m.

It is now:  2010 Jun 9 at 14:26

Loading Simplification Rules

TOOLS.M                                      Revised 2010 February 26

weightlimit = 40
```

summary

According to Paul Bernays (see pages 6 and 10), Ernst Zermelo had attempted in 1915 (in an unpublished notebook) to define ordinals as sets satisfying **image[SUCC, x] ⊂ succ[x]** and **Uclosure[x] ⊂ succ[x]**. For short, such sets are here called Zermelo numbers.

```
In[2]:= "Paul Bernays, A system of axiomatic set theory,
          Part 2, Journal of Symbolic Logic, vol. 6(1941), pp. 1–17.";
```

It was shown in the notebook **ZER-ON-1.NB** that Zermelo numbers belong to the class **REGULAR**. In this second notebook on Zermelo's numbers, various other properties of such sets are derived. It is shown that any Zermelo number **x** is full; that is, every member of **x** is a subset of **x**. Moreover, either **U[x] = x,** or else **succ[U[x]] = x**. It is also shown that Zermelo numbers are invariant under the function **BIGCUP**; that is, if **y ∈ x**, then also **U[y] ∈ x**. A counterexample is presented to show that in general a full set need not be invariant under **BIGCUP**.

fullness

Lemma. If **x** is an almost Uclosed set, then either **U[x] = x** or **U[x] ∈ x**.

```
In[3]:= Map[implies[member[x, y], #] &, SubstTest[implies, and[member[t, u], subclass[u, v]],
           member[t, v], {t → U[x], u → Uclosure[x], v → succ[x]}]] // Reverse

Out[3]= or[equal[x, U[x]], member[U[x], x],
          not[member[x, y]], not[subclass[Uclosure[x], succ[x]]]] == True

In[4]:= or[equal[x_, U[x_]], member[U[x_], x_],
          not[member[x_, y_]], not[subclass[Uclosure[x_], succ[x_]]]] := True
```

Lemma. If **y** belongs to an almost-successor invariant class **x**, then **succ[y] ∈ succ[x]**.

```
In[5]:= SubstTest[implies, and[member[y, x], subclass[x, z]],
          member[y, z], z → image[inverse[SUCC], succ[x]]] // Reverse

Out[5]= or[equal[x, succ[y]], member[succ[y], x],
          not[member[y, x]], not[subclass[image[SUCC, x], succ[x]]]] == True

In[6]:= or[equal[x_, succ[y_]], member[succ[y_], x_],
          not[member[y_, x_]], not[subclass[image[SUCC, x_], succ[x_]]]] := True
```

Restatement.

```
In[7]:= implies[and[subclass[image[SUCC, x], succ[x]], member[y, x]], member[succ[y], succ[x]]]

Out[7]= True
```

Theorem. If **x** is a regular almost-successor invariant set, and **U[x] ∈ x**, then **x = succ[U[x]]**.

```
In[8]:= Map[not,
          SubstTest[and, implies[and[p1, p3], or[p4, p5]], implies[p5, p6], implies[p2, not[p6]],
           not[implies[and[p1, p2, p3], p4]], {p1 -> subclass[image[SUCC, x], succ[x]],
            p2 → member[x, REGULAR], p3 → member[U[x], x], p4 → equal[x, succ[U[x]]],
            p5 → member[succ[U[x]], x], p6 → member[U[x], U[x]]}]] // Reverse

Out[8]= or[equal[x, succ[U[x]]], not[member[x, REGULAR]],
          not[member[U[x], x]], not[subclass[image[SUCC, x], succ[x]]]] == True

In[9]:= or[equal[x_, succ[U[x_]]], not[member[x_, REGULAR]],
          not[member[U[x_], x_]], not[subclass[image[SUCC, x_], succ[x_]]]] := True
```

Theorem. If **x = succ[U[x]]**, then **x** is full.

```
In[10]:= SubstTest[implies, and[subclass[u, v], equal[v, w]],
           subclass[u, w], {u → U[x], v → succ[U[x]], w → x}] // Reverse

Out[10]= or[not[equal[x, succ[U[x]]]], subclass[U[x], x]] == True

In[11]:= or[not[equal[x_, succ[U[x_]]]], subclass[U[x_], x_]] := True
```

Corollary. Variable-free restatement of the preceding theorem.

```
In[14]:= Map[empty, dif[fix[composite[SUCC, BIGCUP]], FULL] // Renormality]

Out[14]= subclass[fix[composite[SUCC, BIGCUP]], FULL] == True

In[15]:= subclass[fix[composite[SUCC, BIGCUP]], FULL] := True
```

Lemma.

```
In[21]:= and[member[x, V], equal[x, succ[U[x]]]] // AssertTest // Reverse

Out[21]= and[member[U[x], x], subclass[x, succ[U[x]]], subclass[U[x], x]] ==
           and[equal[x, succ[U[x]]], member[x, V]]
```

```
In[22]:= and[member[U[x_], x_], subclass[x_, succ[U[x_]]], subclass[U[x_], x_]] :=
          and[equal[x, succ[U[x]]], member[x, V]]
```

Theorem. Membership rule for the class of sets satisfying **x = succ[U[x]]**.

```
In[23]:= member[x, fix[composite[SUCC, BIGCUP]]] // AssertTest

Out[23]= member[x, fix[composite[SUCC, BIGCUP]]] == and[equal[x, succ[U[x]]], member[x, V]]

In[24]:= member[x_, fix[composite[SUCC, BIGCUP]]] := and[equal[x, succ[U[x]]], member[x, V]]
```

Counterexample. In general, a full set **x** need not satisfy either **U[x] = x** nor **x = succ[U[x]]**.

```
In[27]:= Map[not, SubstTest[implies, and[member[u, v], subclass[v, w]],
            member[u, w], {u → set[0, set[0], set[set[0]]], v → FULL,
             w → union[fix[BIGCUP], fix[composite[SUCC, BIGCUP]]]}]] // Reverse

Out[27]= subclass[FULL, union[fix[BIGCUP], fix[composite[SUCC, BIGCUP]]]] == False

In[28]:= subclass[FULL, union[fix[BIGCUP], fix[composite[SUCC, BIGCUP]]]] := False
```

To reduce execution time to about 14 seconds, the following two proof steps are omitted from the following derivation, relying on rewrite rules instead: **implies[and[p0, p2, p3], p1]** and **implies[and[p1, p2, p5], p6]**.

Theorem. Zermelo numbers are full.

```
In[29]:= Map[not, SubstTest[and, implies[and[p1, p3], or[p4, p5]],
            implies[p6, p7], not[implies[and[p0, p2, p3], p7]],
            {p0 → member[x, y], p1 → member[x, REGULAR], p2 -> subclass[image[SUCC, x], succ[x]],
             p3 → subclass[Uclosure[x], succ[x]], p4 → equal[U[x], x], p5 → member[U[x], x],
             p6 → equal[x, succ[U[x]]], p7 → subclass[U[x], x]}]] // Reverse

Out[29]= or[not[member[x, y]], not[subclass[image[SUCC, x], succ[x]]],
           not[subclass[Uclosure[x], succ[x]]], subclass[U[x], x]] == True

In[30]:= or[not[member[x_, y_]], not[subclass[image[SUCC, x_], succ[x_]]],
           not[subclass[Uclosure[x_], succ[x_]]], subclass[U[x_], x_]] := True
```

Corollary. Variable-free restatement.

```
In[31]:= Map[equal[V, #] &, SubstTest[class, x, implies[member[x, u], member[x, v]],
           {u -> intersection[fix[composite[inverse[SUCC], S, UCLOSURE]],
              fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]], v → FULL}]]

Out[31]= subclass[intersection[fix[composite[inverse[SUCC], S, UCLOSURE]],
            fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]], FULL] == True

In[32]:= subclass[intersection[fix[composite[inverse[SUCC], S, UCLOSURE]],
            fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]], FULL] := True
```

Theorem. If **x** is a Zermelo number, then either **x = U[x]** or **x = succ[U[x]]**.

```
In[36]:= Map[not, SubstTest[and, implies[and[p0, p1, p2], p3], implies[and[p0, p1], or[p4, p5]],
            implies[and[p2, p3, p5], p6], not[implies[and[p0, p1, p2], or[p4, p6]]],
            {p0 → member[x, y], p1 → subclass[Uclosure[x], succ[x]],
             p2 → subclass[image[SUCC, x], succ[x]], p3 → member[x, REGULAR],
             p4 → equal[x, U[x]], p5 → member[U[x], x], p6 → equal[x, succ[U[x]]]}]] // Reverse

Out[36]= or[equal[x, succ[U[x]]], equal[x, U[x]], not[member[x, y]],
          not[subclass[image[SUCC, x], succ[x]]], not[subclass[Uclosure[x], succ[x]]]] == True

In[38]:= or[equal[x_, succ[U[x_]]], equal[x_, U[x_]],
          not[member[x_, y_]], not[subclass[image[SUCC, x_], succ[x_]]],
          not[subclass[Uclosure[x_], succ[x_]]]] := True
```

Theorem. (Variable-free restatement.)

```
In[40]:= Map[equal[V, #] &, SubstTest[class, x, implies[member[x, u], member[x, v]],
          {u -> intersection[fix[composite[inverse[SUCC], S, UCLOSURE]],
             fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]],
           v → union[fix[BIGCUP], fix[composite[SUCC, BIGCUP]]]}]]

Out[40]= subclass[intersection[fix[composite[inverse[SUCC], S, UCLOSURE]],
            fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]],
          union[fix[BIGCUP], fix[composite[SUCC, BIGCUP]]]] == True

In[41]:= subclass[intersection[fix[composite[inverse[SUCC], S, UCLOSURE]],
            fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]],
          union[fix[BIGCUP], fix[composite[SUCC, BIGCUP]]]] := True
```

BIGCUP invariance

In this section it is shown that Zermelo numbers are **BIGCUP**-invariant.

Lemma.

```
In[47]:= Map[implies[member[t, y], #] &, SubstTest[implies, and[member[u, v], subclass[v, w]],
            member[u, w], {u → U[t], v → Uclosure[x], w → succ[x]}]] // Reverse

Out[47]= or[equal[x, U[t]], member[U[t], x], not[equal[core[x, U[t]], U[t]]],
          not[member[t, y]], not[subclass[Uclosure[x], succ[x]]]] == True

In[48]:= (% /. {t → t_, x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem. If **x** is a full, regular, almost Uclosed set, and if **y ∈ x**, then **U[y] ∈ x**.

```
In[50]:= Map[not, SubstTest[and, implies[and[p1, p3], p4],
            implies[and[p3, p4], p5], implies[and[p2, p5], or[p6, p7]],
            implies[and[p0, p3], not[p7]], not[implies[and[p0, p1, p2, p3], p6]],
            {p0 → member[x, REGULAR], p1 → full[x], p2 → subclass[Uclosure[x], succ[x]],
             p3 → member[y, x], p4 → subclass[y, x], p5 → member[U[y], Uclosure[x]],
             p6 → member[U[y], x], p7 → equal[U[y], x]}]] // Reverse

Out[50]= or[member[U[y], x], not[member[x, REGULAR]], not[member[y, x]],
           not[subclass[U[x], x]], not[subclass[Uclosure[x], succ[x]]]] == True

In[52]:= or[member[U[y_], x_], not[member[x_, REGULAR]], not[member[y_, x_]],
           not[subclass[U[x_], x_]], not[subclass[Uclosure[x_], succ[x_]]]] := True
```

Eliminating the variable y yields the following.

Theorem. Any full, regular, almost Uclosed set is invariant under **BIGCUP**.

```
In[53]:= Map[equal[V, #] &, SubstTest[class, t,
           or[member[U[t], x], not[member[t, x]], not[member[x, w]], not[subclass[u, v]]],
           {u → Uclosure[x], v → succ[x], w → intersection[FULL, REGULAR]}]]

Out[53]= or[not[member[x, REGULAR]], not[subclass[U[x], x]],
           not[subclass[Uclosure[x], succ[x]]], subclass[image[BIGCUP, x], x]] == True

In[54]:= or[not[member[x_, REGULAR]], not[subclass[U[x_], x_]],
           not[subclass[Uclosure[x_], succ[x_]]], subclass[image[BIGCUP, x_], x_]] := True
```

Theorem. Variable-free restatement.

```
In[55]:= Map[equal[V, #] &, SubstTest[class, x,
           implies[and[member[x, u], subclass[Uclosure[x], succ[x]]], member[x, v]],
           {u -> intersection[FULL, REGULAR], v → invar[BIGCUP]}]]

Out[55]= subclass[intersection[FULL, REGULAR, fix[composite[inverse[SUCC], S, UCLOSURE]]],
           invar[BIGCUP]] == True

In[56]:= subclass[intersection[FULL, REGULAR, fix[composite[inverse[SUCC], S, UCLOSURE]]],
           invar[BIGCUP]] := True
```

Corollary. Zermelo numbers are invariant under **BIGCUP**.

```
In[57]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u → intersection[fix[composite[inverse[SUCC], S, UCLOSURE]],
              fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]],
            v -> intersection[FULL, REGULAR, fix[composite[inverse[SUCC], S, UCLOSURE]]],
            w → invar[BIGCUP]}] // Reverse

Out[57]= subclass[intersection[fix[composite[inverse[SUCC], S, UCLOSURE]],
            fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]], invar[BIGCUP]] == True

In[58]:= subclass[intersection[fix[composite[inverse[SUCC], S, UCLOSURE]],
            fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]], invar[BIGCUP]] := True
```

a counterexample

Counterexample. A full set need not be invariant under **BIGCUP**.

```
In[82]:= Map[not, SubstTest[implies, and[member[u, v], subclass[v, w]],
           member[u, w], {u → union[set[set[set[0]], succ[set[set[0]]]], succ[set[0]]],
            v → FULL, w → invar[BIGCUP]}]] // Reverse

Out[82]= subclass[FULL, invar[BIGCUP]] == False

In[83]:= subclass[FULL, invar[BIGCUP]] := False
```

This counterexample was discovered as follows.

```
In[81]:= Select[Select[Map[ens, Range[100]], full], not[invariant[BIGCUP, #]] &]

Out[81]= {union[set[set[set[0]], succ[set[set[0]]]], succ[set[0]]],
          union[set[succ[set[set[0]]]], succ[set[0]], succ[set[set[set[0]]]]]}
```