subclasses of OMEGA

Johan G. F. Belinfante
2006 March 11

```
In[1]:= SetDirectory["l:"]; << goedel79.07a; << tools.m

:Package Title: goedel79.07a           2006 March 7 at 9:20 p.m.

It is now:  2006 Mar 11 at 17:53

Loading Simplification Rules

TOOLS.M                      Revised 2006 March 7

weightlimit = 40
```

summary

A theorem about classes of ordinals is derived in this notebook, along with two corollaries. The theorem says that if **y** is a member of a class **x** of ordinals, then either **y** is the largest member of **x**, or else **y** is less than some other member of **x**. In the former case, one has **equal[y, U[x]]**, and in the latter case, **member[y, U[x]]** holds. Eliminating the set variable **y** yields the statement that any subclass **x** of **OMEGA** satisfies **subclass[x, succ[U[x]]]**. An **Otter** proof of this corollary was obtained on 1998 March 5 for the special case that **x** is a set. (See theorem **ON-HER-2** in the **ON-6** group.) A variable-free corollary is derived by eliminating the variable **x** as well. These results resemble rewrite rules already available in the **GOEDEL** program, but with the hypothesis **member[x, OMEGA]** in place of the hypothesis **subclass[x, OMEGA]**. In general, any theorem about classes of ordinals must imply a corresponding one about ordinals themselves because any ordinal number is the set of all lesser ordinals.

derivation

Theorem.

```
In[2]:= Map[not, SubstTest[and, implies[p1, or[p2, p3]],
          implies[and[p1, p3], p4], implies[and[p1, q1], q2], implies[p4, q3],
          implies[and[q1, q3], q4], implies[and[p4, q2, q4], q5], implies[and[p2, q1], q5],
          not[implies[and[p1, q1], q5]], {p1 → subclass[x, OMEGA],
           p2 → subclass[x, U[x]], p3 → member[U[x], x], p4 → member[U[x], OMEGA],
           q1 → member[y, x], q2 → member[y, OMEGA], q3 → not[member[U[x], U[x]]],
           q4 → not[member[U[x], y]], q5 → or[member[y, U[x]], equal[y, U[x]]]}]]

Out[2]= or[equal[y, U[x]], member[y, U[x]], not[member[y, x]], not[subclass[x, OMEGA]]] == True

In[3]:= or[equal[y_, U[x_]], member[y_, U[x_]],
          not[member[y_, x_]], not[subclass[x_, OMEGA]]] := True
```

Eliminating the variable **y** yields:

```
In[4]:= Map[equal[V, #] &, SubstTest[class, y, or[equal[y, u], member[y, u],
            not[member[y, x]], not[subclass[x, w]]], {u → U[x], w → OMEGA}]] // Reverse

Out[4]= or[not[subclass[x, OMEGA]], subclass[x, succ[U[x]]]] == True

In[5]:= or[not[subclass[x_, OMEGA]], subclass[x_, succ[U[x_]]]] := True
```

Eliminating the variable **x** yields this corollary.

```
In[6]:= Map[equal[V, #] &, SubstTest[class, x,
            implies[subclass[x, w], subclass[x, succ[U[x]]]], w → OMEGA]] // InvertFix // Reverse

Out[6]= subclass[P[OMEGA], fix[composite[inverse[S], SUCC, BIGCUP]]] == True

In[7]:= subclass[P[OMEGA], fix[composite[inverse[S], SUCC, BIGCUP]]] := True
```