P[OMEGA] is contained in fix[ACLOSURE]

Johan G. F. Belinfante
2004 October 16

```
In[1]:=  SetDirectory["i:"]; << goedel62.14a; << tools.m

        :Package Title: goedel62.14a          2004 October 14 at  12:15 noon

        It is now:  2004 Oct 16 at 22:14

        Loading Simplification Rules

        TOOLS.M                      Revised 2004 September 25

        weightlimit = 40
```

summary

Every nonempty set of ordinals holds a least member. It follows from this that every subclass of **OMEGA** is closed under arbitrary intersections. An involved argument leading up to this fact was discovered yesterday as an accidental byproduct of an attempt to use ordinal numbers to obtain a counterexample to a theorem about coverings in topology, but this involved argument only holds for the special case of sets. After several unsuccessful attempts to eliminate the sethood hypothesis, the problem was presented today to McCune's automated reasoning program **Otter**, using the standard input file with very little effort being expended to set flags or to prepare a suitable weight list. After a few seconds, a proof for the general case without the unwanted sethood hypothesis was found by **Otter**. Examination of the output file produced by **Otter** revealed that the key step in the short argument discovered by **Otter** uses Corollary **ON-BA1-A** in the **ON-5** group of theorems. This corollary is one of several that had been proved using **Otter** way back in 2000 June 6. This corollary had not yet found its way into the **GOEDEL** program. An independent derivation of this prerequisite fact is presented first, and then this result will be used to prove the main theorem, following the argument that **Otter** found today.

Theorem ON-BA1-A used in Otter's proof

Lemma.

```
In[2]:= Map[not, SubstTest[and, implies[p2, p3],
          implies[and[p1, p3], p4], implies[p4, or[p5, p6]],
          implies[and[p2, p5], p7], not[implies[and[p1, p2], or[p6, p7]]],
          {p1 → subclass[U[x], OMEGA], p2 → member[y, x],
           p3 → subclass[y, U[x]], p4 → subclass[y, OMEGA],
           p5 → member[A[y], y], p6 → equal[0, y], p7 → member[A[y], U[x]]}]]

Out[2]= or[equal[0, y], member[A[y], U[x]],
          not[member[y, x]], not[subclass[U[x], OMEGA]]] == True

In[3]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[4]:= Map[equal[V, #] &, SubstTest[class, y, or[equal[0, y], member[A[y], U[x]],
            not[member[y, x]], not[subclass[U[x], w]]], w → OMEGA]] // Reverse

Out[4]= or[not[subclass[U[x], OMEGA]],
          subclass[x, union[image[inverse[BIGCAP], U[x]], singleton[0]]]] == True

In[5]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[6]:= SubstTest[implies, subclass[x, y], subclass[image[w, x], image[w, y]],
         {y -> union[image[inverse[BIGCAP], U[x]], singleton[0]], w → BIGCAP}]

Out[6]= or[not[subclass[x, union[image[inverse[BIGCAP], U[x]], singleton[0]]]],
          subclass[image[BIGCAP, x], U[x]]] == True

In[7]:= (% /. x → x_) /. Equal → SetDelayed
```

Otter's Theorem **ON-BA1-A** proved 2000 June 6.

```
In[8]:= Map[not,
         SubstTest[and, implies[p1, p2], implies[p2, p3], not[implies[p1, p3]],
          {p1 -> subclass[U[x], OMEGA],
           p2 -> subclass[x, union[image[inverse[BIGCAP], U[x]], singleton[0]]],
           p3 -> subclass[image[BIGCAP, x], U[x]]}]]

Out[8]= or[not[subclass[U[x], OMEGA]], subclass[image[BIGCAP, x], U[x]]] == True

In[9]:= or[not[subclass[U[x_], OMEGA]], subclass[image[BIGCAP, x_], U[x_]]] := True
```

corollary

This section contains **Otter**'s short proof of the main theorem, based on the theorem presented in the preceding section.

```
In[10]:= SubstTest[implies, subclass[U[y], OMEGA],
          subclass[image[BIGCAP, y], U[y]], y → P[x]]

Out[10]= or[not[subclass[x, OMEGA]], subclass[Aclosure[x], x]] == True

In[11]:= (% /. x → x_) /. Equal → SetDelayed
```

This result can be sharpened up by replacing the inclusion in the conclusion by an equation. This is the main theorem.

```
In[12]:= or[not[subclass[x, OMEGA]], equal[Aclosure[x], x]] // AssertTest

Out[12]= or[equal[x, Aclosure[x]], not[subclass[x, OMEGA]]] == True

In[13]:= or[equal[x_, Aclosure[x_]], not[subclass[x_, OMEGA]]] := True
```

A variable-free formulation is obtained by eliminating the variable **x**. This is slightly weaker in that **x** need not be a set. The variable-free version only implies the main theorem for the special case that **x** is a set.

```
In[14]:= Map[equal[V, #] &, SubstTest[class, x,
            implies[subclass[x, w], equal[x, Aclosure[x]]], w → OMEGA]] // Reverse

Out[14]= subclass[P[OMEGA], fix[ACLOSURE]] == True

In[15]:= subclass[P[OMEGA], fix[ACLOSURE]] := True
```