Theorem ON−SUC−1revisited.

Johan G. F. Belinfante
2003 June 3

```
In[1]:= << goedel52.r93; << tools.m

        :Package Title: goedel52.r93       2003 June 1 at 11:11 a.m.

        It is now:  2003 Jun 3 at 10:16

        Loading Simplification Rules

        TOOLS.M                       Revised 2003 May 31

        weightlimit = 40
```

■ summary

One of the many theorems proved using **Otter** in my 1999 paper on computer proofs in ordinal number theory just says that if **x** and **y** are ordinals satisfying **x < y**, then **x + 1 < y + 1**.

**Johan G. F. Belinfante, "On Computer−Assisted Proofs in Ordinal Number Theory,"
Journal of Automated Reasoning,
volume 22, pages 341 − 378 (1999). (Theorem ON − SUC − 1 in the middle of page 372)**

In this notebook an elegant reformulation of this theorem is obtained by eliminating the variable **x**.

■ the statement of Theorem ON−SUC−1

This theorem **ON−SUC−1** is not immediately recognized by the **GOEDEL** program because the membership rule for **succ** causes the statement of the theorem to be expanded as follows:

```
In[2]:= implies[and[member[x, y], member[y, OMEGA]], member[succ[x], succ[y]]]

Out[2]= or[and[equal[y, succ[x]], member[x, V]],
          member[succ[x], y], not[member[x, y]], not[member[y, OMEGA]]]
```

The truth of the following simpler statements are however immediately recognized by the **GOEDEL** program:

```
In[5]:= or[member[x, V], not[member[x, y]]]

Out[5]= True

In[6]:= or[equal[y, succ[x]], member[succ[x], y], not[member[x, y]], not[member[y, OMEGA]]]

Out[6]= True
```

It only takes an application of double−negation for the **GOEDEL** program to deduce the truth of Theorem **ON−SUC−1** from these simpler facts.

```
In[7]:= implies[and[member[x, y], member[y, OMEGA]], member[succ[x], succ[y]]] // NotNotTest

Out[7]= or[and[equal[y, succ[x]], member[x, V]],
          member[succ[x], y], not[member[x, y]], not[member[y, OMEGA]]] == True
```

For convenience, this will be added as a temporary rewrite rule.

```
In[8]:= or[and[equal[y_, succ[x_]], member[x_, V]],
          member[succ[x_], y_], not[member[x_, y_]], not[member[y_, OMEGA]]] := True
```

■ removing the variables from Theorem ON−SUC−1

The process of eliminating the variable **x** from theorem **ON−SUC−1** is simplified by removing a rewrite rule for images of successors:

```
In[9]:= image[x_, succ[y_]] =.
```

A reformulation involving only one variable can now be obtained as follows:

```
In[10]:= Map[equal[V, #] &,
           SubstTest[class, x, implies[and[member[x, y], member[y, z]], member[succ[x], w]],
            {z -> OMEGA, w -> succ[y]}]] // Reverse

Out[10]= or[not[member[y, OMEGA]], subclass[image[SUCC, y], succ[y]]] == True
```

This rule can be added to the **GOEDEL** program:

```
In[11]:= or[not[member[y_, OMEGA]], subclass[image[SUCC, y_], succ[y_]]] := True
```

One can also remove the other variable, but the result is admittedly a bit cryptic:

```
In[12]:= Map[equal[V, #] &, SubstTest[class, y,
            implies[member[y, w], subclass[image[SUCC, y], succ[y]]], w -> OMEGA]] // Reverse

Out[12]= subclass[OMEGA, fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]] == True

In[13]:= subclass[OMEGA, fix[composite[inverse[SUCC], S, IMAGE[SUCC]]]] := True
```