ordlist[x] ⊂ S

Johan G. F. Belinfante
2007 May 12

```
In[1]:= SetDirectory["l:"]; << goedel93.11a; << tools.m

        :Package Title: goedel93.11a      2007 May 11 at 5:30 p.m.

        It is now:  2007 May 12 at 22:26

        Loading Simplification Rules

        TOOLS.M                                      Revised 2007 May 6

        weightlimit = 40
```

summary

In this notebook it is shown that the function **ordlist[x]** is a subclass of the subset relation **S**. What this inclusion means is intuitively obvious: if one begins listing the ordinals in a class **x** in strictly increasing order, **o[0] < o[1] < o[2] < ⋯** , then **n ≤ o[n]** for every natural number **n ∈ ω**. The main tools of the derivation are the monotonicity and uniqueness theorems for **iterate**.

derivation

Lemma.

```
In[2]:= Assoc[composite[id[OMEGA], S], SUCC, id[omega]]

Out[2]= composite[id[OMEGA], S, id[omega], SUCC] ==
         composite[id[OMEGA], inverse[IMAGE[id[omega]]], E]

In[3]:= % /. Equal → SetDelayed
```

Lemma.

```
In[4]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
          {t → id[image[inverse[S], ord[x]]], u → omega, v → OMEGA}] // Reverse

Out[4]= subclass[intersection[omega, image[inverse[S], ord[x]]], ord[x]] == True

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

Removing the **ord** wrapper yields:

```
In[6]:= SubstTest[implies, equal[x, ord[t]],
          member[x, invar[composite[id[omega], inverse[S]]]], t → x] // Reverse

Out[6]= or[not[member[x, OMEGA]], subclass[intersection[omega, image[inverse[S], x]], x]] == True

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

The variable **x** can be eliminated:

```
In[8]:= Map[equal[V, #] &, SubstTest[class, x, implies[member[x, u], member[x, v]],
          {u → OMEGA, v → invar[composite[id[omega], inverse[S]]]}]]

Out[8]= subclass[OMEGA, invar[composite[id[omega], inverse[S]]]] == True

In[9]:= % /. Equal → SetDelayed
```

Corollary.

```
In[10]:= equal[composite[id[OMEGA], inverse[IMAGE[id[omega]]], E], composite[id[OMEGA],
           inverse[IMAGE[inverse[S]]], inverse[IMAGE[id[omega]]], E]] // AssertTest

Out[10]= equal[composite[id[OMEGA], inverse[IMAGE[id[omega]]], E], composite[
           id[OMEGA], inverse[IMAGE[inverse[S]]], inverse[IMAGE[id[omega]]], E]] == True

In[11]:= % /. Equal → SetDelayed
```

Theorem.

```
In[12]:= SubstTest[implies,
          and[equal[composite[w, SUCC], composite[u, w]], equal[image[w, set[0]], v]],
          equal[iterate[u, v], composite[w, id[omega]]],
          {u → composite[id[OMEGA], E], v → OMEGA, w → composite[id[OMEGA], S, id[omega]]}]

Out[12]= True ==
          equal[composite[id[OMEGA], S, id[omega]], iterate[composite[id[OMEGA], E], OMEGA]]

In[13]:= iterate[composite[id[OMEGA], E], OMEGA] := composite[id[OMEGA], S, id[omega]]
```

Main Theorem.

```
In[14]:= SubstTest[implies, and[subclass[t, u], subclass[v, w]],
           subclass[iterate[t, v], iterate[u, w]],
           {t -> composite[HULL[intersection[OMEGA, x]], SUCC], u -> composite[id[OMEGA], E],
            v -> set[A[intersection[OMEGA, x]]], w → OMEGA}] // Reverse

Out[14]= subclass[ordlist[x], S] == True

In[15]:= subclass[ordlist[x_], S] := True
```