ordlist[complement[ord[x]]]

Johan G. F. Belinfante
2007 May 23

```
In[1]:= SetDirectory["l:"]; << goedel93.22a; << tools.m

        :Package Title: goedel93.22a      2007 May 22 at 8:35 p.m.

        It is now:  2007 May 23 at 6:52

        Loading Simplification Rules

        TOOLS.M                                      Revised 2007 May 6

        weightlimit = 40
```

summary

The function **ordlist[complement[ord[x]]]** lists the ordinals that are not less than a given ordinal **ord[x]** in increasing order, starting with the least one. The first ordinal to be listed clearly is **ord[x]** itself, and then its successor, and so on, ad infinitum. In this notebook it is shown that, as one would suspect, this function is identical to the function **iterate[SUCC, set[ord[x]]]**. (Comment: In 2003 June 6 this function was used to show that the class of limit ordinals is a proper class. For details, see the notebook **limitord.nb**.)

derivation

Lemma.

```
In[2]:= SubstTest[subclass, range[ordlist[t]], t, t -> complement[ord[x]]] // Reverse

Out[2]= equal[0, intersection[ord[x], range[ordlist[complement[ord[x]]]]]] == True

In[3]:= intersection[ord[x_], range[ordlist[complement[ord[x_]]]]] := 0
```

Lemma.

```
In[4]:= equal[composite[id[intersection[OMEGA, complement[ord[x]]]],
          ordlist[complement[ord[x]]]], ordlist[complement[ord[x]]]]

Out[4]= True

In[5]:= composite[id[intersection[OMEGA, complement[ord[x_]]]],
          ordlist[complement[ord[x_]]]] := ordlist[complement[ord[x]]]
```

Corollary.

```
In[6]:= Assoc[HULL[intersection[OMEGA, complement[ord[x]]]],
          id[intersection[OMEGA, complement[ord[x]]]], ordlist[complement[ord[x]]]]

Out[6]= composite[HULL[intersection[OMEGA, complement[ord[x]]]], ordlist[complement[ord[x]]]] ==
          ordlist[complement[ord[x]]]

In[7]:= composite[HULL[intersection[OMEGA, complement[ord[x_]]]],
           ordlist[complement[ord[x_]]]] := ordlist[complement[ord[x]]]
```

Lemma.

```
In[8]:= SubstTest[implies, and[invariant[t, u], invariant[t, v]],
           invariant[t, intersection[u, v]],
           {t → SUCC, u → OMEGA, v → complement[ord[x]]}] // Reverse

Out[8]= equal[0, intersection[image[SUCC, intersection[OMEGA, complement[ord[x]]]], ord[x]]] ==
          True

In[9]:= intersection[image[SUCC, intersection[OMEGA, complement[ord[x_]]]], ord[x_]] := 0
```

Lemma.

```
In[10]:= equal[intersection[OMEGA, image[SUCC, ord[x]]], image[SUCC, ord[x]]]

Out[10]= True

In[11]:= intersection[OMEGA, image[SUCC, ord[x_]]] := image[SUCC, ord[x]]
```

Lemma.

```
In[12]:= equal[composite[id[intersection[OMEGA, complement[ord[x]]]],
            SUCC, id[intersection[OMEGA, complement[ord[x]]]]],
           composite[SUCC, id[intersection[OMEGA, complement[ord[x]]]]]]

Out[12]= True

In[13]:= composite[id[intersection[OMEGA, complement[ord[x_]]]],
            SUCC, id[intersection[OMEGA, complement[ord[x_]]]]] :=
           composite[SUCC, id[intersection[OMEGA, complement[ord[x]]]]]
```

Lemma.

```
In[14]:= Map[composite[#, ordlist[complement[ord[x]]]] &,
           Assoc[HULL[intersection[OMEGA, complement[ord[x]]]],
            id[intersection[OMEGA, complement[ord[x]]]],
            composite[SUCC, id[intersection[OMEGA, complement[ord[x]]]]]]]

Out[14]= composite[HULL[intersection[OMEGA, complement[ord[x]]]], SUCC,
            ordlist[complement[ord[x]]]] == composite[SUCC, ordlist[complement[ord[x]]]]

In[15]:= composite[HULL[intersection[OMEGA, complement[ord[x_]]]], SUCC,
            ordlist[complement[ord[x_]]]] := composite[SUCC, ordlist[complement[ord[x]]]]
```

Main theorem. (Uniqueness of **iterate**.)

```
In[16]:= SubstTest[implies, and[equal[composite[w, SUCC], composite[u, w]],
           equal[image[w, set[0]], v]],
          equal[composite[w, id[omega]], iterate[u, v]],
          {u → SUCC, v → set[ord[x]], w → ordlist[complement[ord[x]]]}]

Out[16]= True == equal[iterate[SUCC, set[ord[x]]], ordlist[complement[ord[x]]]]

In[17]:= ordlist[complement[ord[x_]]] := iterate[SUCC, set[ord[x]]]
```