ordlist[x] is one-to-one

Johan G. F. Belinfante
2006 March 12

```
In[1]:= SetDirectory["l:"]; << goedel79.11a; << tools.m

        :Package Title: goedel79.11a           2006 March 11 at 6:10 p.m.

        It is now:  2006 Mar 12 at 9:24

        Loading Simplification Rules

        TOOLS.M                      Revised 2006 March 7

        weightlimit = 40
```

summary

The function **ordlist[x]** lists in order the first countably many ordinals in a given class **x**. It is defined as follows:

```
In[2]:= Begin["Goedel`Private`"];

In[3]:= InfoMatch[class[w_, HoldPattern[member[x_, ordlist[y_]]]]] // First // First

Out[3]= class[w_, member[x_, ordlist[y_]]] := Module[{p = Unique[]}, class[w, exists[p, and[
          member[x, p], subclass[p, union[cart[set[0], set[A[intersection[OMEGA, y]]]],
          composite[HULL[intersection[OMEGA, y]], SUCC, p, inverse[SUCC], id[omega]]]]]]]]
```

An expicit formula can be given in terms of **iterate**. Many properties of **ordlist[x]** have already been derived from the general rewrite rules for **iterate**.

```
In[4]:= iterate[composite[HULL[intersection[OMEGA, x]], SUCC], set[A[intersection[OMEGA, x]]]]

Out[4]= ordlist[x]
```

For example, it follows from the general theory of **iterate** that the domain of **ordlist[x]** is either a natural number or the set of all natural numbers. In this notebook some propertues of the range are derived, and it is shown that the function **ordlist[x]** is one-to-one.

domain and range

```
In[5]:= SubstTest[subclass, domain[iterate[u, v]], omega,
          {u → composite[HULL[intersection[OMEGA, x]], id[OMEGA], SUCC],
           v → set[A[intersection[OMEGA, x]]], w → ordlist[x]}]

Out[5]= True
```

```
In[6]:= subclass[domain[ordlist[x_]], omega] := True
```

Since the domain of **ordlist[x]** is contained in **omega**, one has:

```
In[7]:= ImageComp[ordlist[x], id[omega], V] // Reverse

        Reverse::normal : Nonatomic expression expected at position 1 in Reverse[True]. More…

Out[7]= Reverse[True]
```

```
In[8]:= image[ordlist[x_], omega] := range[ordlist[x]]
```

Lemma.

```
In[9]:= SubstTest[subclass, range[iterate[u, v]], union[range[u], v],
         {u -> composite[HULL[intersection[OMEGA, x]], SUCC],
          v -> set[A[intersection[OMEGA, x]]]}]

Out[9]= subclass[range[ordlist[x]], union[image[HULL[intersection[OMEGA, x]], range[SUCC]],
          set[A[intersection[OMEGA, x]]]]] == True
```

```
In[10]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem 1.

```
In[11]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
          {u -> range[ordlist[x]], v -> union[image[HULL[intersection[OMEGA, x]], range[SUCC]],
             set[A[intersection[OMEGA, x]]]], w → OMEGA}]

Out[11]= subclass[range[ordlist[x]], OMEGA] == True
```

```
In[12]:= subclass[range[ordlist[x_]], OMEGA] := True
```

Theorem 2.

```
In[13]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
          {u -> range[ordlist[x]], v -> union[image[HULL[intersection[OMEGA, x]], range[SUCC]],
             set[A[intersection[OMEGA, x]]]], w → x}]

Out[13]= subclass[range[ordlist[x]], x] == True
```

```
In[14]:= subclass[range[ordlist[x_]], x_] := True
```

Corollary 1.

```
In[15]:= equal[composite[id[OMEGA], ordlist[x]], ordlist[x]]

Out[15]= True
```

```
In[16]:= composite[id[OMEGA], ordlist[x_]] := ordlist[x]
```

Corollary 2.

```
In[17]:= equal[composite[id[x], ordlist[x]], ordlist[x]]

Out[17]= True

In[18]:= composite[id[x_], ordlist[x_]] := ordlist[x]
```

an alternate formula for ordlist[x]

From the uniqueness theorem for **iterate** one deduces the following alternate formula for **ordlist[x]**:

```
In[19]:= 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 → composite[HULL[intersection[OMEGA, x]], id[OMEGA], SUCC],
            v → set[A[intersection[OMEGA, x]]], w → ordlist[x]}]

Out[19]= equal[iterate[composite[HULL[intersection[OMEGA, x]], id[OMEGA], SUCC],
           set[A[intersection[OMEGA, x]]]], ordlist[x]] == True

In[20]:= iterate[composite[HULL[intersection[OMEGA, x_]], id[OMEGA], SUCC],
           set[A[intersection[OMEGA, x_]]]] := ordlist[x]
```

one-to-one property

Lemma.

```
In[21]:= TRANSITIVE[composite[id[OMEGA], E]] // AssertTest

Out[21]= TRANSITIVE[composite[id[OMEGA], E]] == True

In[22]:= TRANSITIVE[composite[id[OMEGA], E]] := True

In[23]:= SubstTest[implies, and[subclass[u, v], equal[0, fix[trv[v]]], equal[0, fix[trv[u]]],
           {u -> composite[HULL[intersection[OMEGA, x]], id[OMEGA], SUCC],
            v → composite[id[OMEGA], E]}]

Out[23]= equal[0, fix[trv[composite[HULL[intersection[OMEGA, x]], id[OMEGA], SUCC]]]] == True

In[24]:= fix[trv[composite[HULL[intersection[OMEGA, x_]], id[OMEGA], SUCC]]] := 0

In[25]:= SubstTest[implies, and[FUNCTION[u], equal[0, fix[trv[u]]]],
           FUNCTION[inverse[iterate[u, set[v]]]],
           {u → composite[HULL[intersection[OMEGA, x]], id[OMEGA], SUCC],
            v → A[intersection[OMEGA, x]]}]

Out[25]= FUNCTION[inverse[ordlist[x]]] == True

In[26]:= FUNCTION[inverse[ordlist[x_]]] := True
```

repercussions

Since the range of **ordlist[x]** is a subclass of **x**, it follows that this range is finite when **x** is finite.

```
In[27]:= SubstTest[implies, and[subclass[y, x], member[x, FINITE]],
          member[y, FINITE], y → range[ordlist[x]]]

Out[27]= or[member[range[ordlist[x]], FINITE], not[member[x, FINITE]]] == True

In[28]:= or[member[range[ordlist[x_]], FINITE], not[member[x_, FINITE]]] := True
```

Since **ordlist[x]** is one-to-one, finiteness of its range implies finiteness of its domain.

```
In[29]:= SubstTest[implies, and[ONEONE[y], member[range[y], FINITE]],
          member[domain[y], FINITE], y → ordlist[x]]

Out[29]= or[member[domain[ordlist[x]], FINITE], not[member[range[ordlist[x]], FINITE]]] == True

In[30]:= (% /. x → x_) /. Equal → SetDelayed
```

It follows that the domain of **ordlist[x]** is a natural number when **x** is finite.

```
In[31]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], implies[p3, p4],
            implies[p4, p5], not[implies[p1, p5]], {p1 → member[x, FINITE],
             p2 → member[range[ordlist[x]], FINITE], p3 → member[domain[ordlist[x]], FINITE],
             p4 → not[equal[omega, domain[ordlist[x]]]], p5 → member[domain[ordlist[x]], omega]}]]

Out[31]= or[member[domain[ordlist[x]], omega], not[member[x, FINITE]]] == True

In[32]:= or[member[domain[ordlist[x_]], omega], not[member[x_, FINITE]]] := True
```

empty ordlist[x]

The function **ordlist[x]** is empty if **x** has no ordinal elements.

```
In[33]:= SubstTest[equal, 0, iterate[u, v], {u -> composite[HULL[intersection[OMEGA, x]], SUCC],
            v -> set[A[intersection[OMEGA, x]]]}]

Out[33]= equal[0, ordlist[x]] == equal[0, intersection[OMEGA, x]]

In[34]:= equal[0, ordlist[x_]] := equal[0, intersection[OMEGA, x]]
```

Lemma.

```
In[35]:= equiv[equal[0, domain[funpart[x]]], equal[0, funpart[x]]]

Out[35]= True

In[36]:= equal[0, domain[funpart[x_]]] := equal[0, funpart[x]]
```

Corollary.

```
In[37]:= SubstTest[equal, 0, domain[funpart[y]], y → ordlist[x]]

Out[37]= equal[0, domain[ordlist[x]]] == equal[0, intersection[OMEGA, x]]

In[38]:= equal[0, domain[ordlist[x_]]] := equal[0, intersection[OMEGA, x]]
```

Corollary.

```
In[39]:= SubstTest[member, 0, domain[iterate[u, v]],
           {u -> composite[HULL[intersection[OMEGA, x]], SUCC],
            v -> set[A[intersection[OMEGA, x]]]}]

Out[39]= member[0, domain[ordlist[x]]] == not[equal[0, intersection[OMEGA, x]]]

In[40]:= member[0, domain[ordlist[x_]]] := not[equal[0, intersection[OMEGA, x]]]
```