the double wrapper card[ord[x]]

Johan G. F. Belinfante
2008 June 3

```
In[1]:= SetDirectory["l:"]; << goedel.08may29a; << tools.m

:Package Title: goedel.08may29a                    2008 May 29 at  6:20 a.m.

It is now:  2008 Jun 3 at 15:22

Loading Simplification Rules

TOOLS.M                                         Revised 2008 May 17

weightlimit = 40
```

summary

The double-wrapper **card[ord[x]]** is used to derive formulas for the class **fix[CARD]** of cardinal numbers (=initial ordinals).

a double wrapper

The set **card[ord[x]]** is always a cardinal number:

```
In[2]:= member[card[ord[x]], fix[CARD]]

Out[2]= True
```

Every cardinal can be expressed in this fashion. In the rest of this section a wrapper-removal rewrite rule is derived. Some lemmas are needed for this.

Lemma.

```
In[3]:= SubstTest[implies, equal[x, ord[t]],
          or[equal[x, card[ord[x]]], not[equal[x, card[x]]]], t → x] // Reverse

Out[3]= or[equal[x, card[ord[x]]], not[equal[x, card[x]]], not[member[x, OMEGA]]] == True

In[4]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[5]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p2, p3], p4],
          not[implies[and[p1, p3], p4]], {p1 → member[x, fix[CARD]], p2 → member[x, OMEGA],
           p3 → equal[x, card[x]], p4 -> equal[x, card[ord[x]]]}]] // Reverse

Out[5]= or[equal[x, card[ord[x]]], not[equal[x, card[x]]], not[member[x, V]]] == True

In[6]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. (Double-wrapper removal rule.)

```
In[7]:= equiv[equal[x, card[ord[x]]], and[equal[x, card[x]], member[x, V]]]

Out[7]= True

In[8]:= equal[x_, card[ord[x_]]] := and[equal[x, card[x]], member[x, V]]
```

OMEGA ⊂ invar[CARD]

Lemma.

```
In[9]:= Map[composite[Id, complement[#]] &,
         dif[composite[CARD, id[OMEGA]], inverse[S]] // complement // RelnNormality]

Out[9]= composite[intersection[CARD, complement[inverse[S]]], id[OMEGA]] == 0

In[10]:= % /. Equal → SetDelayed
```

Theorem. (Inclusion: **CARD** ∘ **id[Ω]** ⊂ **inverse[S]**).

```
In[11]:= SubstTest[empty, dif[u, v], {u -> composite[CARD, id[OMEGA]], v -> inverse[S]}]

Out[11]= subclass[composite[id[OMEGA], inverse[CARD]], S] == True

In[12]:= subclass[composite[id[OMEGA], inverse[CARD]], S] := True
```

Restatement of the theorem.

```
In[13]:= subclass[composite[CARD, id[OMEGA]], inverse[S]]

Out[13]= True
```

Corollary.

```
In[14]:= SubstTest[implies, subclass[u, v], subclass[image[u, w], image[v, w]],
           {u -> composite[CARD, id[OMEGA]], v -> inverse[S], w → ord[x]}] // Reverse

Out[14]= subclass[image[CARD, ord[x]], image[inverse[S], ord[x]]] == True

In[15]:= (% /. x → x_) /. Equal → SetDelayed
```

A better result is possible:

```
In[16]:= SubstTest[subclass, t, intersection[u, v],
           {t -> image[CARD, ord[x]], u → OMEGA, v -> image[inverse[S], ord[x]]}] // Reverse

Out[16]= subclass[image[CARD, ord[x]], ord[x]] == True

In[17]:= subclass[image[CARD, ord[x_]], ord[x_]] := True
```

The **ord** wrappper can be removed.

```
In[18]:= SubstTest[implies, equal[x, ord[t]], invariant[CARD, x], t → x] // Reverse

Out[18]= or[not[member[x, OMEGA]], subclass[image[CARD, x], x]] == True

In[19]:= or[not[member[x_, OMEGA]], subclass[image[CARD, x_], x_]] := True
```

Corollary. (A variable-free restatement.)

```
In[20]:= Map[equal[V, #] &, complement[dif[OMEGA, invar[CARD]]] // Normality]

Out[20]= subclass[OMEGA, invar[CARD]] == True

In[21]:= subclass[OMEGA, invar[CARD]] := True
```

---

a general result

The following generalizes a result derived in the next section, where a much stronger result is obtained for the special case where **x** is **CARD**.

```
In[22]:= Map[equal[V, #] &,
          dif[fix[composite[inverse[E], IMAGE[x]]], range[x]] // complement // Normality]

Out[22]= subclass[fix[composite[inverse[E], IMAGE[x]]], range[x]] == True

In[23]:= subclass[fix[composite[inverse[E], IMAGE[x_]]], range[x_]] := True
```

---

fix[inverse[E] ∘ IMAGE[CARD]]] is empty

Lemma. (A corollary of **Ω ⊂ invar[CARD]**.)

```
In[24]:= Map[not, SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
            {u → ord[x], v -> image[CARD, ord[x]], w -> image[inverse[S], ord[x]]}]] // Reverse

Out[24]= member[ord[x], image[CARD, ord[x]]] == False

In[25]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. (An improved result obtained by removing the **ord** wrapper.)

```
In[26]:= Map[not, SubstTest[implies, equal[x, ord[t]], not[member[x, image[CARD, x]]], t → x]] //
          Reverse

Out[26]= member[x, image[CARD, x]] == False

In[27]:= member[x_, image[CARD, x_]] := False
```

Theorem. (Variable-free restatement.)

```
In[28]:= SubstTest[class, x, member[x, image[t, x]], t → CARD] // InvertFix

Out[28]= fix[composite[inverse[E], IMAGE[CARD]]] == 0

In[29]:= fix[composite[inverse[E], IMAGE[CARD]]] := 0
```

a formula for fix[CARD]

Lemma. (A corollary of the fact that, for ordinals, containment is reversed nonmembership.)

```
In[30]:= Assoc[composite[id[OMEGA], complement[E]], id[OMEGA], CARD]

Out[30]= composite[id[OMEGA], complement[E], CARD] == composite[id[OMEGA], inverse[S], CARD]

In[31]:= composite[id[OMEGA], complement[E], CARD] := composite[id[OMEGA], inverse[S], CARD]
```

Theorem. (A formula for the class of cardinals = initial ordinals.)

```
In[32]:= SubstTest[fix, composite[id[w], complement[E], CARD], w → OMEGA]

Out[32]= intersection[OMEGA, complement[fix[composite[E, CARD]]]] == fix[CARD]

In[33]:= intersection[OMEGA, complement[fix[composite[E, CARD]]]] := fix[CARD]
```

Corollary.

```
In[34]:= SubstTest[dif, OMEGA, dif[OMEGA, t], t → fix[composite[E, CARD]]]

Out[34]= intersection[OMEGA, fix[composite[E, CARD]]] ==
          intersection[OMEGA, complement[fix[CARD]]]

In[35]:= intersection[OMEGA, fix[composite[E, CARD]]] :=
          intersection[OMEGA, complement[fix[CARD]]]
```

another characterization of fix[CARD]

In this section, another characterization is derived for the class **fix[CARD]** of cardinal numbers. The result resembles the formula in the preceding section, but with **E ∘ CARD** replaced by **E ∘ Q**.

Lemma. (A membership rule.)

```
In[36]:= member[x, fix[composite[E, y]]] // AssertTest

Out[36]= member[x, fix[composite[E, y]]] == member[x, image[inverse[y], x]]

In[37]:= member[x_, fix[composite[E, y_]]] := member[x, image[inverse[y], x]]
```

Lemma. (An inclusion derived from the result of the preceding section.)

```
In[39]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
           {t → id[OMEGA], u -> fix[composite[E, CARD]], v -> fix[composite[E, Q]]}] // Reverse

Out[39]= subclass[OMEGA, union[fix[CARD], fix[composite[E, Q]]]] == True

In[40]:= subclass[OMEGA, union[fix[CARD], fix[composite[E, Q]]]] := True

In[42]:= SubstTest[implies, and[member[x, y], subclass[y, z]], member[x, z],
           {y → OMEGA, z -> union[fix[CARD], fix[composite[E, Q]]]}] // Reverse

Out[42]= or[equal[x, card[x]], member[x, image[Q, x]], not[member[x, OMEGA]]] == True

In[43]:= or[equal[x_, card[x_]], member[x_, image[Q, x_]], not[member[x_, OMEGA]]] := True
```

Corollary.

```
In[44]:= SubstTest[implies, member[t, OMEGA],
           or[equal[t, card[t]], member[t, image[Q, t]]], t → ord[x]] // Reverse

Out[44]= or[equal[card[ord[x]], ord[x]], member[ord[x], image[Q, ord[x]]]] == True

In[45]:= or[equal[card[ord[x_]], ord[x_]], member[ord[x_], image[Q, ord[x_]]]] := True
```

Lemma. (A simplification rule.)

```
In[46]:= ImageComp[id[image[Q, OMEGA]], id[OMEGA], ord[x]] // Reverse

Out[46]= intersection[image[Q, OMEGA], ord[x]] == ord[x]

In[47]:= intersection[image[Q, OMEGA], ord[x_]] := ord[x]
```

Theorem. (A result similar to the lemma, but with **CARD** replaced by **Q**.)

```
In[49]:= Map[member[ord[t], #] &, ImageComp[Q, id[image[Q, OMEGA]], ord[t]] // Reverse] /.
          t → card[ord[x]]

Out[49]= member[card[ord[x]], image[Q, card[ord[x]]]] == False

In[50]:= member[card[ord[x_]], image[Q, card[ord[x_]]]] := False
```

Corollary. (Removing the double wrapper.)

```
In[51]:= SubstTest[implies, equal[x, card[ord[t]]], not[member[x, image[Q, x]]], t → x] // Reverse

Out[51]= or[not[equal[x, card[x]]], not[member[x, image[Q, x]]]] == True

In[52]:= or[not[equal[x_, card[x_]]], not[member[x_, image[Q, x_]]]] := True
```

Theorem. (Eliminating the variable **x**.)

```
In[53]:= Map[equal[V, #] &, SubstTest[class, x,
           implies[member[x, u], not[member[x, v]]], {u → fix[CARD], v → fix[composite[E, Q]]}]]

Out[53]= equal[0, intersection[fix[CARD], fix[composite[E, Q]]]] == True

In[54]:= intersection[fix[CARD], fix[composite[E, Q]]] := 0
```

Corollary. (A formula for the class of cardinals.)

```
In[55]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u → fix[CARD], v → dif[OMEGA, fix[composite[E, Q]]]}]

Out[55]= equal[fix[CARD], intersection[OMEGA, complement[fix[composite[E, Q]]]]] == True

In[56]:= intersection[OMEGA, complement[fix[composite[E, Q]]]] := fix[CARD]
```

Corollary. (A similar result.)

```
In[57]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u → dif[OMEGA, fix[CARD]], v → intersection[OMEGA, fix[composite[E, Q]]]}]

Out[57]= equal[intersection[OMEGA, complement[fix[CARD]]],
           intersection[OMEGA, fix[composite[E, Q]]]] == True

In[58]:= intersection[OMEGA, fix[composite[E, Q]]] := intersection[OMEGA, complement[fix[CARD]]]
```