rewrite rules for the ordinal wrapper ord[x]

Johan G. F. Belinfante and Ming Li
2006 March 14

In[1]:= SetDirectory["l:"]; << goedel79.12b; << tools.m

:Package Title: goedel79.12b           2006 March 12 at 7:25 p.m.
It is now:     2006 Mar 14 at 14:2

Loading Simplification Rules

TOOLS.M                 Revised 2006 March 7
weightlimit = 40

summary

Theorems about ordinals can often be formulated compactly by using the \texttt{ord} wrapper. In this notebook some facts about ordinals are reformulated as rewrite rules involving this wrapper.

A rule

Lemma.

\begin{verbatim}
In[2]:= SubstTest[implies, member[y, OMEGA], or[equal[0, A[y]], equal[0, y]], y -> ord[x]]
Out[2]= or[equal[0, A[ord[x]]], equal[0, ord[x]]] = True

In[3]:= (% /. x -> x_) /. Equal -> SetDelayed
\end{verbatim}

The least member of any nonzero ordinal is 0.

\begin{verbatim}
In[4]:= equal[A[ord[x]], complement[image[V, ord[x]]]]
In[5]:= A[ord[x_]] := complement[image[V, ord[x]]]
\end{verbatim}

fix-HULL

Any ordinal is closed under arbitrary intersections.
image under inverse[SUCC]

Theorem.

As a corollary, one obtains the following formula for the restriction of the successor function to an ordinal, generalizing an existing rule for the special case of \( \omega \).

ordinals less than or equal to a given ordinal

The set of ordinals less than or equal to a given ordinal is its successor.

other intersection rules

Lemma.
In[16]:= (member[y, fix[composite[inverse[E], IMAGE[inverse[S]]]]] // AssertTest // Reverse) /. y -> ord[x]
Out[16]= member[ord[x], image[inverse[S], ord[x]]] = False

In[17]:= member[ord[x_], image[inverse[S], ord[x_]]] := False

Theorem.

In[18]:= Map[intersection[FULL, #] &, ImageComp[id[P[ord[x]]], composite[inverse[S], id[P[ord[x]]]], ord[x]]]
Out[18]= intersection[FULL, image[inverse[S], ord[x]]] = ord[x]

In[19]:= intersection[FULL, image[inverse[S], ord[x_]]] := ord[x]

Corollary.

In[20]:= AssInt[image[inverse[S], ord[x]], FULL, OMEGA]
Out[20]= intersection[OMEGA, image[inverse[S], ord[x]]] = ord[x]

In[21]:= intersection[OMEGA, image[inverse[S], ord[x_]]] := ord[x]

fullness

Every ordinal is full:

In[22]:= SubstTest[implies, and[member[x, z], member[z, OMEGA]], subclass[x, z], z -> ord[y]]
Out[22]= or[not[member[x, ord[y]]], subclass[x, ord[y]]] = True

In[23]:= or[not[member[x_, ord[y_]]], subclass[x_, ord[y_]]] := True

successors

If x is less than an ordinal y, then the successor of x is less than or equal to y. Comment: The set x is also an ordinal, but one need not say so explicitly.

In[24]:= or[not[member[y, OMEGA]], not[member[x, y]], subclass[succ[x], y]] // AssertTest
Out[24]= or[not[member[x, y]], not[member[y, OMEGA]], subclass[succ[x], y]] = True

In[25]:= or[not[member[x_, y_]], not[member[y_, OMEGA]], subclass[succ[x_], y_]] := True

Reformulation using the ord wrapper.
an ordinal with no largest member is a limit ordinal

The direction of this rule is chosen to encourage the use of equality substitution rules, which helps to reduce the total number of rules needed for reasoning about ordinals.

members of predecessors

The converse also holds.

The direction of this rule is chosen to agree with a similar rule for the nat wrapper. Comment. The set $x$ is also an ordinal, but no explicit ord wrapper is needed here.

The dichotomy rule involves predecessors:
subclass rules involving omega

Because of dichotomy, any statement that one ordinal is contained in another can be rewritten in terms of membership.

\[
\text{In[36]} := \text{SubstTest}[[\text{subclass, ord}[x], \text{ord}[y], y \rightarrow \text{omega}]\\
\text{Out[36]} = \text{subclass}[\text{ord}[x], \text{omega}] = \text{not}[\text{member}[\text{omega}, \text{ord}[x]]]\\
\text{In[37]} := \text{subclass}[\text{ord}[x_\_], \text{omega}] = \text{not}[\text{member}[\text{omega}, \text{ord}[x]]]
\]

A similar result.

\[
\text{In[38]} := \text{SubstTest}[[\text{or, equal}[\text{ord}[y], \text{ord}[x]], \text{member}[\text{ord}[x], \text{ord}[y]], y \rightarrow \text{omega}]\\
\text{Out[38]} = \text{or}[[\text{equal}[\text{omega}, \text{ord}[x]], \text{member}[\text{ord}[x], \text{omega}]]] = \text{not}[\text{member}[\text{omega}, \text{ord}[x]]]\\
\text{In[39]} := \text{or}[[\text{equal}[\text{omega}, \text{ord}[x_\_]], \text{member}[\text{ord}[x_\_], \text{omega}]]] = \text{not}[\text{member}[\text{omega}, \text{ord}[x]]]
\]

**RUSSELL**

The Russell class frequently enters into reasoning about ordinals.

\[
\text{In[40]} := \text{SubstTest}[[\text{implies, and}[[\text{subclass}[u, v], \text{subclass}[v, w]], \\
\text{subclass}[u, w], [u \rightarrow \text{ord}[x], v \rightarrow \text{OMEGA}, w \rightarrow \text{RUSSELL}]]]\\
\text{Out[40]} = \text{subclass}[\text{ord}[x], \text{RUSSELL}] = \text{True}\\
\text{In[41]} := \text{subclass}[\text{ord}[x_\_], \text{RUSSELL}] = \text{True}
\]

Corollary.

\[
\text{In[42]} := \text{equal}[\text{intersection}[\text{RUSSELL}, \text{ord}[x]], \text{ord}[x]]\\
\text{Out[42]} = \text{True}\\
\text{In[43]} := \text{intersection}[\text{RUSSELL}, \text{ord}[x_\_]] = \text{ord}[x]
\]

**Uclosure rule**

Successor ordinals are closed under arbitrary unions.

\[
\text{In[44]} := \text{SubstTest}[[\text{Uclosure, ord}[y], y \rightarrow \text{succ}[\text{ord}[x]]]]\\
\text{Out[44]} = \text{Uclosure}[[\text{succ}[\text{ord}[x]]]] = \text{succ}[\text{ord}[x]]\\
\text{In[45]} := \text{Uclosure}[[\text{succ}[\text{ord}[x_\_]]]] = \text{succ}[\text{ord}[x]]
\]