the limit ordinals form a proper class

Johan G. F. Belinfante
2003 June 6

In[1]:= << goedel52.r97; << tools.m

:Package Title: goedel52.r97 2003 June 4 at 12:05 noon

It is now: 2003 Jun 6 at 11:15

Loading Simplification Rules

TOOLS.M Revised 2003 May 31

weightlimit = 40

summary

In Gödel’s class theory, there are two kinds of classes, sets and proper classes. A class is a set if it is a member of some class. Proper classes are not members of any class. The class of all ordinals OMEGA is an example of a proper class; that is, this class does not belong to any class. Every ordinal \( x \) is full, that is, its sum class \( U[x] \) is contained in \( x \). An ordinal \( x \) is called a limit ordinal if equality holds: \( U[x] = x \). The class of limit ordinals is

\[
\text{In[2]}:= \text{class}\{x, \text{and}\{\text{member}\{x, \text{OMEGA}\}, \text{equal}\{U[x], x\}\}\}\]

\[
\text{Out[2]}= \text{intersection}\{\text{OMEGA}, \text{fix}\{\text{BIGCUP}\}\}\]

A successor ordinal is an ordinal that is the successor of some other ordinal. An ordinal is a successor ordinal if and only if it is not a limit ordinal:

\[
\text{In[3]}:= \text{class}\{x, \text{exists}\{y, \text{and}\{\text{equal}\{x, \text{succ}\{y\}\}, \text{member}\{y, \text{OMEGA}\}\}\}\}\]

\[
\text{Out[3]}= \text{intersection}\{\text{OMEGA}, \text{complement}\{\text{fix}\{\text{BIGCUP}\}\}\}\]

It was shown in the notebook ON-SC-4B.NB that the successor ordinals form a proper class.

\[
\text{In[4]}:= \text{member}\{\text{intersection}\{\text{OMEGA}, \text{complement}\{\text{fix}\{\text{BIGCUP}\}\}\}, V\}\]

\[
\text{Out[4]}= \text{False}
\]

In this notebook the uniqueness theorem for iterate is used to derive the fact that the class of limit ordinals is also a proper class. The basic strategy is to show that any ordinal belongs to a limit ordinal. If \( x \) is an ordinal number, then the set \( \text{range}[\text{iterate}[\text{suc}\{x\}]] \) is the successor–invariant set of ordinals \( \{x, x+1, x+2, \ldots\} \). It will be shown in this notebook that the sum class of this set is a limit ordinal to which \( x \) belongs. Knowing that any ordinal belongs to a limit ordinal allows one to deduce that the sum class of the class of limit ordinals is the class of all ordinals. Since the latter is a proper class, it follows immediately that the class of limit ordinals is also a proper class. This is because a class is a proper class if and only if its sum class is a proper class.
The first task is to show that the class of limit ordinals is identical with the class of unions of successor–invariant sets of ordinals. We begin with a lemma based on Theorem ON–SUC–1 which says that for ordinals $x < y$ implies $x + 1 < y + 1$. The implication \texttt{implies[p4,p5]} in the following reasoning can be obtained from Theorem ON–SUC–1 by eliminating the variable $x$.

\begin{verbatim}
In[5]:= Map[not, SubstTest[and, implies[and[p1, p2], p4], implies[p4, p5],
        implies[and[p1, p3], p6], implies[p6, p7], implies[and[p5, p7], p8],
        not[implies[and[p1, p2, p3], p8]],
        {p1 \rightarrow member[z, x], p2 \rightarrow subclass[x, OMEGA],
        p3 \rightarrow invariant[SUCC, x], p4 \rightarrow member[z, OMEGA],
        p5 \rightarrow subclass[image[SUCC, z], succ[z]],
        p6 \rightarrow member[succ[z], x], p7 \rightarrow subclass[succ[z], U[x]],
        p8 \rightarrow subclass[image[SUCC, z], U[x]]}]]

Out[5]= or[not[member[z, x]], not[subclass[x, OMEGA]],
        not[subclass[image[SUCC, x], x]], subclass[image[SUCC, z], U[x]]] == True
\end{verbatim}

The variable $z$ can be eliminated:

\begin{verbatim}
In[6]:= Map[assert[forall[z, #]] \&, %]

Out[6]= or[not[subclass[x, OMEGA]],
        not[subclass[image[SUCC, x], x]], subclass[image[SUCC, U[x]], U[x]]] == True
\end{verbatim}

\begin{verbatim}
In[7]:= or[not[subclass[x_, OMEGA]],
        not[subclass[image[SUCC, x_], x_]],
        subclass[image[SUCC, U[x_]], U[x_]]] := True
\end{verbatim}

Restatement: if $x$ is a successor–invariant set of ordinals, the $U[x]$ is also successor–invariant. Eliminating the variable $x$ yields:

\begin{verbatim}
In[8]:= Map[equal[0, #] \&, dif[intersection[P[OMEGA], invar[SUCC]],
        image[inverse[BIGCUP], invar[SUCC]]] // Renormality]

Out[8]= subclass[image[BIGCUP, intersection[invar[SUCC], P[OMEGA]]], invar[SUCC]] == True
\end{verbatim}

\begin{verbatim}
In[9]:= subclass[image[BIGCUP, intersection[invar[SUCC], P[OMEGA]]], invar[SUCC]] := True
\end{verbatim}

The sum class of any subset of the class of ordinal numbers is an ordinal number: The following corollary of this will be needed:

\begin{verbatim}
In[10]:= SubstTest[equal, 0, dif[u, v],
        {u \rightarrow intersection[x, P[OMEGA]], v \rightarrow image[inverse[BIGCUP], OMEGA]}] // Reverse

\end{verbatim}

\begin{verbatim}
In[11]:= subclass[image[BIGCUP, intersection[x_, P[OMEGA]]], OMEGA] := True
\end{verbatim}

From the above facts one deduces:

\begin{verbatim}
In[12]:= SubstTest[subclass, u, intersection[v, w],
        {u \rightarrow image[BIGCUP, intersection[invar[SUCC], P[OMEGA]]],
        v \rightarrow OMEGA, w \rightarrow invar[SUCC]]]

Out[12]= subclass[image[BIGCUP, intersection[invar[SUCC], P[OMEGA]]], fix[BIGCUP]] == True
\end{verbatim}
\[\text{In[13]}:= \text{subclass}\[\text{image[BIGCUP, intersection[invar[SUCC, P[OMEGA]]]], fix[BIGCUP]} := \text{True}\]

In the next section, the reverse inclusion is derived.

### inclusion in the other direction

We begin with a lemma:

\[\text{In[14]}:= \text{SubstTest[implies, and[member[y, x], subclass[x, z]], member[y, z], z -> fix[BIGCUP]]} \// \text{MapNotNot}\]

\[\text{Out[14]}= \text{or[equal[y, U[y]], not[member[y, x]], not[subclass[x, fix[BIGCUP]]]]} \text{=} \text{True}\]

\[\text{In[15]}:= \text{or[equal[y_, U[y_]], not[member[y_, x_]], not[subclass[x_, fix[BIGCUP]]]]} := \text{True}\]

From this one obtains:

\[\text{In[16]}:= \text{Map[not, SubstTest[and, implies[and[p1, p2], p3], implies[p1, p4], implies[and[p3, p4], p5], not[implies[and[p1, p2], p5]], }\]
\[\{p1 -> \text{member[y, x], p2 -> subclass[x, fix[BIGCUP]], p3 -> equal[U[y], y], p4 -> member[U[y], image[BIGCUP, x]], p5 -> member[y, image[BIGCUP, x]]}\}]]\]

\[\text{Out[16]}= \text{or[member[y, image[BIGCUP, x]], not[member[y, x]], not[subclass[x, fix[BIGCUP]]]]} \text{=} \text{True}\]

The variable \( y \) can be eliminated:

\[\text{In[17]}:= \text{Map[assert[forall[y, #]] &, %]}
\]

\[\text{Out[17]}= \text{or[not[subclass[x, fix[BIGCUP]]], subclass[x, image[BIGCUP, x]]]} \text{=} \text{True}\]

This says that any class of fixed points of \( \text{BIGCUP} \) is subvariant under \( \text{BIGCUP} \).  Remark: This is by no means limited to \( \text{BIGCUP} \).

\[\text{In[18]}:= \text{or[not[subclass[x_, fix[BIGCUP]]], subclass[x_, image[BIGCUP, x_]]]} := \text{True}\]

To apply this to the case at hand requires two inclusions.  This is the first:

\[\text{In[19]}:= \text{SubstTest[implies, subclass[x, fix[BIGCUP]], subclass[x, image[BIGCUP, x]], x -> intersection[OMEGA, fix[BIGCUP]]]}\]

\[\text{Out[19]}= \text{subclass[intersection[OMEGA, fix[BIGCUP]], image[BIGCUP, intersection[OMEGA, fix[BIGCUP]]]]} \text{=} \text{True}\]

\[\text{In[20]}= \text{subclass[intersection[OMEGA, fix[BIGCUP]], image[BIGCUP, intersection[OMEGA, fix[BIGCUP]]]]} := \text{True}\]

We need a lemma to derive the second inclusion.

\[\text{In[21]}:= \text{SubstTest[subclass, intersection[x, y], intersection[x, P[U[y]]], y -> OMEGA]}\]

\[\text{Out[21]}= \text{subclass[U[intersection[OMEGA, x]], OMEGA]} \text{=} \text{True}\]

\[\text{In[22]}= \text{subclass[U[intersection[OMEGA, x_]], OMEGA]} := \text{True}\]

This is the second inclusion:
Applying transitivity of inclusion, one obtains:

\[
\text{In}[25]:= \text{SubstTest}[\text{implies, and}[\text{subclass}[u, v], \text{subclass}[v, w]], \text{subclass}[u, w], \\
\quad \{u \rightarrow \text{intersection}[\text{OMEGA}, \text{fix}[\text{BIGCUP}]], \\
\quad v \rightarrow \text{image}[\text{BIGCUP}, \text{intersection}[\text{OMEGA}, \text{fix}[\text{BIGCUP}]]], \\
\quad w \rightarrow \text{image}[\text{BIGCUP}, \text{intersection}[\text{invar}[\text{SUCC}], \text{P}[\text{OMEGA}]]])
\]
\]
\[
\text{Out}[25]= \text{subclass}[\text{intersection}[\text{OMEGA}, \text{fix}[\text{BIGCUP}]], \\
\quad \text{image}[\text{BIGCUP}, \text{intersection}[\text{invar}[\text{SUCC}], \text{P}[\text{OMEGA}]]]) := \text{True}
\]

It only remains to combine the two inclusions into an equation:

\[
\text{In}[26]:= \text{SubstTest}[\text{implies, and}[\text{subclass}[u, v], \text{subclass}[v, w]], \text{subclass}[u, w], \\
\quad \{u \rightarrow \text{intersection}[\text{OMEGA}, \text{fix}[\text{BIGCUP}]], \\
\quad v \rightarrow \text{image}[\text{BIGCUP}, \text{intersection}[\text{OMEGA}, \text{fix}[\text{BIGCUP}]]], \\
\quad w \rightarrow \text{image}[\text{BIGCUP}, \text{intersection}[\text{invar}[\text{SUCC}], \text{P}[\text{OMEGA}]]])
\]
\]
\[
\text{Out}[26]= \text{subclass}[\text{intersection}[\text{OMEGA}, \text{fix}[\text{BIGCUP}]], \\
\quad \text{image}[\text{BIGCUP}, \text{intersection}[\text{invar}[\text{SUCC}], \text{P}[\text{OMEGA}]]]) := \text{True}
\]

\textbf{iterate uniqueness}

Apply the uniqueness theorem for \texttt{iterate}:

\[
\text{In}[29]:= \text{SubstTest}[\text{implies, and}[\text{equal}[\text{image}[w, \text{singleton}[0]], v], \\
\quad \text{equal}[\text{composite}[u, w], \text{composite}[w, \text{SUCC}]], \\
\quad \text{equal}[\text{iterate}[u, v], \text{composite}[w, \text{id}[\text{OMEGA}]]], \\
\quad \{u \rightarrow \text{SUCC}, v \rightarrow \text{singleton}[x], \\
\quad w \rightarrow \text{composite}[\text{id}[\text{OMEGA}], \text{iterate}[\text{SUCC}, \text{singleton}[x]]])
\]
\]
\[
\text{Out}[29]= \text{or}[\text{and}[\text{member}[x, V], \text{not}[\text{member}[x, \text{OMEGA}]]), \\
\quad \text{subclass}[\text{range}[\text{iterate}[\text{SUCC}, \text{singleton}[x]]], \text{OMEGA}]] := \text{True}
\]

This can be cleaned up:

\[
\text{In}[30]:= \text{Map}[\text{implies}[\text{member}[x, \text{OMEGA}], \#, \%]
\]
\]
\[
\text{Out}[30]= \text{or}[\text{not}[\text{member}[x, \text{OMEGA}]], \text{subclass}[\text{range}[\text{iterate}[\text{SUCC}, \text{singleton}[x]]], \text{OMEGA}]] := \text{True}
\]

\[
\text{In}[31]:= \text{or}[\text{not}[\text{member}[x, \text{OMEGA}]], \\
\quad \text{subclass}[\text{range}[\text{iterate}[\text{SUCC}, \text{singleton}[x]]], \text{OMEGA}]] := \text{True}
\]

This just says that if \( x \) is an ordinal, then \( \{x, x+1, x+2, \ldots \} \) is a set of ordinals.
This section is concerned with a general lemma, not specifically about ordinals. This lemma is needed to show that \( x \) belongs to the sum class of the set of ordinals \( \{x, x+1, \ldots \} \).

```math
In[32]:= \text{SubstTest[subclass, image[u, v], range[u],}
\{u \mapsto \text{iterate}[x, y], v \mapsto \text{singleton}[\text{singleton}[0]]\}]\nOut[32]= \text{subclass[image[x, y], range[iterate[x, y]]] == True}
```

```math
In[33]:= \text{subclass[image[x_, y_], range[iterate[x_, y_]]] := True}
```

```math
In[34]:= \text{SubstTest[subclass, image[u, v], range[iterate[u, v]],}
\{u \mapsto \text{SUCC}, v \mapsto \text{singleton}[x]\}]\nOut[34]= \text{or[member[\text{succ}[x], range[iterate[\text{SUCC}, \text{singleton}[x]]]],}
\not[member[x, V]]\] == True
```

```math
In[35]:= \text{or[member[\text{succ}[x_], range[\text{iterate}[\text{SUCC}, \text{singleton}[x_]]]],}
\not[member[x_, V]]\] := True
```

```math
In[36]:= \text{Map[not, SubstTest[\text{and}, \text{implies}[p1, p2],}
\text{implies}[p2, p3], \text{implies}[p3, p4],}
\not[\text{implies}[p1, p4]],}
\{p1 \mapsto \text{member}[x, y], p2 \mapsto \text{member}[x, V],}
\text{p3 \mapsto member[\text{succ}[x], range[iterate[\text{SUCC}, \text{singleton}[x]]]],}
\text{p4 \mapsto member[x, U[range[iterate[\text{SUCC}, \text{singleton}[x]]]]]]])}
Out[36]= \text{or[member[x, U[range[iterate[\text{SUCC}, \text{singleton}[x]]]]]],}
\not[member[x, y]]\] == True
```

```math
In[37]:= \text{or[member[x_, U[range[iterate[\text{SUCC}, \text{singleton}[x_]]]]],}
\not[member[x_, y_]]\] := True
```

For the next section we rewrite this a bit. Note that

```math
In[38]:= \text{class[y, member[x, U[y]]]}
Out[38]= \text{complement[P[P[complement[\text{singleton}[x]]]]]}
```

The result of this section can therefore be rewritten as follows:

```math
In[39]:= \text{implies[member[x, y], member[range[iterate[\text{SUCC}, \text{singleton}[x]]],}
\complement[P[P[complement[\text{singleton}[x]]]]]]}]
Out[39]= \text{True}
```

**eliminating the details of the construction**

The next step is to hide the details of the \text{iterate} construction. All one really needs to know is that there exists a limit ordinal to which any given ordinal belongs; how it is constructed is not important. We will shortly need this:

```math
In[40]:= \text{SubstTest[U, image[\text{BIGCUP}, x], x \mapsto intersection[invar[\text{SUCC}], P[\text{OMEGA}]]] // Reverse}
Out[40]= U[U[intersection[invar[\text{SUCC}], P[\text{OMEGA}]]]] == U[\text{intersection[\text{OMEGA}, fix[\text{BIGCUP}]]]}
```

```math
In[41]:= U[U[intersection[invar[\text{SUCC}], P[\text{OMEGA}]]]] := U[\text{intersection[\text{OMEGA}, fix[\text{BIGCUP}]]]}
```

and this..
In[42]:= \[\text{\texttt{SubstTest[implies, implies[p, member[x, OMEGA]], implies[p, not[equal[0, s]]],}
}
\text{\texttt{\{p \rightarrow member[x, OMEGA], x \rightarrow range[iterate[SUCC, singleton[x]]], s \rightarrow}}
\text{\texttt{intersection[P[OMEGA], invar[SUCC], complement[P[complement[singleton[x]]]]]]}]]
\text{\texttt{]}}
\text{\texttt{// NotNotTest}}

Out[42]= \text{\texttt{or[and[member[x, \text{\texttt{range[iterate[SUCC, singleton[x]]]}],}
\text{\texttt{subclass[range[iterate[SUCC, singleton[x]]], OMEGA]},}}
\text{\texttt{not[member[x, OMEGA]]] == True}}}

In[43]:= \text{\texttt{or[and[member[x, OMEGA, \text{\texttt{range[iterate[SUCC, singleton[x]]]}],}
\text{\texttt{subclass[range[iterate[SUCC, singleton[x]]], OMEGA]},}}
\text{\texttt{not[member[x, OMEGA]]] == True}}}

These facts are combined as follows:

In[44]:= \text{\texttt{SubstTest[implies, implies[p, member[x, s]],}
\text{\texttt{implies[p, not[equal[0, s]]],}
\text{\texttt{\{p \rightarrow member[x, OMEGA], x \rightarrow range[iterate[SUCC, singleton[x]]],}}
\text{\texttt{s \rightarrow}}
\text{\texttt{intersection[P[OMEGA], invar[SUCC], complement[P[complement[singleton[x]]]]]]}]]
\text{\texttt{]}}
\text{\texttt{// NotNotTest}}

Out[44]= \text{\texttt{or[member[x, \text{\texttt{intersection[OMEGA, fix[BIGCUP]]]}],}}
\text{\texttt{not[member[x, OMEGA]]] == True}}

The variable \(x\) is eliminated:

In[45]:= \text{\texttt{Map[assert[forall[x, \#]] &, \%]}}

Out[45]= \text{\texttt{subclass[OMEGA, \text{\texttt{intersection[OMEGA, fix[BIGCUP]]]] == True}}}

In[46]:= \text{\texttt{subclass[OMEGA, \text{\texttt{intersection[OMEGA, fix[BIGCUP]]]]] := True}}}

This can be strengthened to an equation:

In[47]:= \text{\texttt{SubstTest[and, subclass[u, v], subclass[v, u],}}
\text{\texttt{\{u \rightarrow OMEGA, v \rightarrow \text{\texttt{intersection[OMEGA, fix[BIGCUP]]]}\]]}]
\text{\texttt{]}}

Out[47]= \text{\texttt{True == equal[OMEGA, \text{\texttt{intersection[OMEGA, fix[BIGCUP]]]]}}}

The sum class of the class of limit ordinals is the class of all ordinals:

In[48]:= \text{\texttt{U[intersection[OMEGA, fix[BIGCUP]]] := OMEGA}}

It now follows immediately that the class of limit ordinals is a proper class:

In[49]:= \text{\texttt{SubstTest[member, U[x], V, x \rightarrow \text{\texttt{intersection[OMEGA, fix[BIGCUP]]]}] // Reverse}}
\text{\texttt{]}}

Out[49]= \text{\texttt{member[intersection[OMEGA, fix[BIGCUP]], V] == False}}

In[50]:= \text{\texttt{member[intersection[OMEGA, fix[BIGCUP]], V] := False}}

A more general rewrite rule can be derived:

In[51]:= \text{\texttt{member[intersection[OMEGA, fix[BIGCUP]], x] // AssertTest}}
\text{\texttt{]}}

Out[51]= \text{\texttt{member[intersection[OMEGA, fix[BIGCUP]], x] == False}}

In[52]:= \text{\texttt{member[intersection[OMEGA, fix[BIGCUP]], x_] := False}}

To summarize: the class of all ordinals is the union of two disjoint subclasses, the class of successor ordinals and the class of limit ordinals, both of which are proper classes.