A. P. Morse’s definition of ordinals

Johan G. F. Belinfante
2010 June 25

In[1]:=
```
SetDirectory["l:"]; << goedel.10jun24a; << tools.m
```

```
:Package Title: goedel.10jun24a                2010 June 24 at 8:30 p.m.

It is now:  2010 Jun 25 at 10:51

Loading Simplification Rules

TOOLS.M                                        Revised 2010 February 26

weightlimit = 40
```

summary

This notebook was inspired by A. P. Morse’s (somewhat complicated) definition of ordinals. However, instead of considering two-sided restrictions of the membership relation **E**, here only one-sided restrictions are considered, which yields simpler results.

In[2]:=
```
"Anthony Perry Morse, A Theory of Sets, Academic
  Press, Inc., Orlando, Florida, second ed., 1986.  See page 100";
```

derivation

Theorem.

In[3]:=
```
equal[intersection[OMEGA,
  cliques[union[complement[S], composite[inverse[SUCC], E]]]], OMEGA]
```

Out[3]=
```
True
```

In[4]:=
```
intersection[OMEGA, cliques[union[complement[S], composite[inverse[SUCC], E]]]] := OMEGA
```

Lemma.

In[5]:=
```
Map[equal[V, #] &, intersection[complement[FUND],
    complement[image[S, intersection[complement[set[0]], subvar[PS]]]],
    H[FULL], P[RUSSELL]] // complement // Normality]
```

Out[5]=
```
subclass[intersection[H[FULL], P[RUSSELL]],
  union[FUND, image[S, intersection[complement[set[0]], subvar[PS]]]]] == True
```

In[6]:=
```
% /. Equal → SetDelayed
```

Lemma.

```
In[7]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
          {t → id[H[FULL]], u -> intersection[H[FULL], P[RUSSELL]],
           v -> union[FUND, image[S, intersection[complement[set[0]], subvar[PS]]]]}] // Reverse

Out[7]= subclass[intersection[H[FULL], P[RUSSELL]],
          union[OMEGA, image[S, intersection[complement[set[0]], subvar[PS]]]]] == True

In[8]:= % /. Equal → SetDelayed
```

Main Theorem. A characterization of ordinals.

```
In[9]:= SubstTest[and, subclass[u, v], subclass[v, u],
         {u -> intersection[H[FULL], P[RUSSELL], FUND], v -> intersection[H[FULL], P[RUSSELL],
            complement[image[S, intersection[complement[set[0]], subvar[PS]]]]]}]

Out[9]= equal[OMEGA,
          intersection[complement[image[S, intersection[complement[set[0]], subvar[PS]]]],
           H[FULL], P[RUSSELL]]] == True

In[10]:= intersection[complement[image[S, intersection[complement[set[0]], subvar[PS]]]],
           H[FULL], P[RUSSELL]] := OMEGA
```

other intersections

Theorem.

```
In[22]:= equal[intersection[chains[S],
            fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]]],
          fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]]]

Out[22]= True

In[23]:= intersection[chains[S], fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]]] :=
          fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]]
```

Theorem.

```
In[24]:= AssInt[complement[image[S, intersection[complement[set[0]], subvar[PS]]]],
          complement[image[S, intersection[complement[set[0]], subvar[PS]]]], chains[S]]

Out[24]= intersection[complement[image[S, intersection[complement[set[0]], subvar[PS]]]],
            fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]]] ==
          fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]]

In[25]:= intersection[complement[image[S, intersection[complement[set[0]], subvar[PS]]]],
            fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]]] :=
          fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]]
```

Lemma.

```
In[26]:= SubstTest[equal, intersection[u, v], v,
          {u -> complement[image[S, intersection[complement[set[0]], subvar[PS]]]],
           v -> fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]]}]

Out[26]= subclass[intersection[fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]],
            subvar[PS]], set[0]] == True

In[27]:= % /. Equal → SetDelayed
```

Theorem.

```
In[28]:= equal[intersection[
           fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]], subvar[PS]], set[0]]

Out[28]= True

In[29]:= intersection[
           fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]], subvar[PS]] := set[0]
```

introducing variables

Theorem.

```
In[30]:= SubstTest[implies, equal[x, ord[t]],
           subclass[P[x], union[fix[composite[E, BIGCAP]], set[0]]], t → x] // Reverse

Out[30]= or[not[member[x, OMEGA]],
           subclass[P[x], union[fix[composite[E, BIGCAP]], set[0]]]] == True

In[31]:= or[not[member[x_, OMEGA]],
           subclass[P[x_], union[fix[composite[E, BIGCAP]], set[0]]]] := True
```

Theorem.

```
In[32]:= Map[implies[and[member[x, y], #], member[x, OMEGA]] &,
          SubstTest[member, x, intersection[t, u, v, w], {t → P[FULL], u → P[RUSSELL],
            v -> fix[image[inverse[CART], image[inverse[IMAGE[id[S]]], WO]]], w -> FULL}]]

Out[32]= or[member[x, OMEGA], not[member[x, y]],
           not[subclass[x, FULL]], not[subclass[x, RUSSELL]],
           not[subclass[P[x], union[fix[composite[E, BIGCAP]], set[0]]]],
           not[subclass[U[x], x]]] == True

In[34]:= or[member[x_, OMEGA], not[member[x_, y_]],
           not[subclass[P[x_], union[fix[composite[E, BIGCAP]], set[0]]]],
           not[subclass[x_, FULL]], not[subclass[x_, RUSSELL]], not[subclass[U[x_], x_]]] := True
```

Theorem.

```
In[67]:= Map[implies[member[x, y], #] &, SubstTest[or, not[and[member[x, y], subclass[y, z]]],
            member[x, z], {y -> intersection[H[FULL], P[RUSSELL]], z ->
              union[OMEGA, image[S, intersection[complement[set[0]], subvar[PS]]]]}]] // Reverse

Out[67]= or[member[x, OMEGA], not[member[x, y]],
           not[subclass[x, FULL]], not[subclass[x, RUSSELL]],
           not[subclass[U[x], x]], not[WELLFOUNDED[composite[id[x], PS]]]] == True

In[68]:= or[member[x_, OMEGA], not[member[x_, y_]],
           not[subclass[x_, FULL]], not[subclass[x_, RUSSELL]],
           not[subclass[U[x_], x_]], not[WELLFOUNDED[composite[id[x_], PS]]]] := True
```

It is not immediately clear whether any of these conditions can be omitted. When the axiom of regularity is assumed, one only needs two:

Theorem.

```
In[63]:= Map[implies[member[x, y], #] &, SubstTest[member, x,
           complement[dif[intersection[H[FULL], complement[image[V, complement[t]]]], OMEGA]],
           t → REGULAR]]

Out[63]= or[member[x, OMEGA], not[AxReg], not[member[x, y]],
           not[subclass[x, FULL]], not[subclass[U[x], x]]] == True

In[70]:= or[member[x_, OMEGA], not[AxReg], not[member[x_, y_]],
           not[subclass[x_, FULL]], not[subclass[U[x_], x_]]] := True
```

serendipity

Theorem.

```
In[38]:= AssInt[H[FULL], P[FULL], P[x]]

Out[38]= intersection[H[FULL], P[intersection[FULL, x]]] == intersection[H[FULL], P[x]]

In[39]:= intersection[H[FULL], P[intersection[FULL, x_]]] := intersection[H[FULL], P[x]]
```

Theorem.

```
In[40]:= AssInt[RUSSELL, REGULAR, P[x]]

Out[40]= intersection[RUSSELL, P[intersection[REGULAR, x]]] == P[intersection[REGULAR, x]]

In[41]:= intersection[RUSSELL, P[intersection[REGULAR, x_]]] := P[intersection[REGULAR, x]]
```

Theorem.

```
In[42]:= AssInt[RUSSELL, P[RUSSELL], P[x]]

Out[42]= intersection[RUSSELL, P[intersection[RUSSELL, x]]] == P[intersection[RUSSELL, x]]
```

```
In[43]:= intersection[RUSSELL, P[intersection[RUSSELL, x_]]] := P[intersection[RUSSELL, x]]
```