# Hartogs numbers are cardinals

Johan G. F. Belinfante
2008 June 5

*In[1]:=* **SetDirectory["l:"]; << goedel.08jun03a; << tools.m**

```
:Package Title: goedel.08jun03a                  2008 June 3 at 4:25 p.m.

It is now:  2008 Jun 5 at 13:4

Loading Simplification Rules

TOOLS.M                                      Revised 2008 May 17

weightlimit = 40
```

## summary

It is shown in this notebook that Hartogs numbers are cardinals. More precisely, the range of the function **HARTOGS** is contained in the class **fix[CARD]** of inital ordinals. The class **hartogs[x]** is defined as follows: (here **x** can be any class, not necessarily a set)

*In[2]:=* **hartogs[x]**

*Out[2]=* intersection[OMEGA, image[Q, P[x]]]

In general, **hartogs[x]** can be either an ordinal or the class **OMEGA** of all ordinals.

*In[3]:=* **or[member[hartogs[x], OMEGA], equal[hartogs[x], OMEGA]]**

*Out[3]=* True

The function **HARTOGS** is the function that takes a set **x** to **hartogs[x]**, provided that the latter is an ordinal. At the present stage of development of the theory of Hartogs numbers within the **GOEDEL** program, Hartogs' theorem has not yet been derived, and not much is available about the domain or range of the function **HARTOGS**.

*In[4]:=* **lambda[x, hartogs[x]]**

*Out[4]=* HARTOGS

## hartogs[x] does not belong to itself

The result derived in this section was adapted from the following reference:

*In[5]:=* **"Patrick Suppes, *Axiomatic Set Theory*, Dover Publications, Inc., New York, 1972.";**

The following variable-free statement is already available in the **GOEDEL** program. In this section, a version of this result is derived for the class **hartogs[x]**.

```
In[6]:= intersection[HARTOGS, composite[Q, inverse[S]]]

Out[6]= 0
```

A cloely related variable-free statement that looks rather different is easily derived:

```
In[7]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]],
          subclass[u, w], {u -> range[HARTOGS], v → OMEGA, w -> RUSSELL}] // Reverse

Out[7]= subclass[range[HARTOGS], RUSSELL] == True

In[8]:= subclass[range[HARTOGS], RUSSELL] := True
```

Lemma.

```
In[9]:= Map[not, SubstTest[and, member[t, t], member[t, OMEGA], t → hartogs[x]]] // Reverse

Out[9]= or[not[member[intersection[OMEGA, image[Q, P[x]]], OMEGA]],
          not[member[intersection[OMEGA, image[Q, P[x]]], image[Q, P[x]]]]] == True

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

Lemma. (Corollary of the fact that **OMEGA** is a proper class.)

```
In[11]:= SubstTest[implies, equal[OMEGA, t],
           not[member[t, y]], t → intersection[OMEGA, x]] // Reverse

Out[11]= or[not[member[intersection[OMEGA, x], y]], not[subclass[OMEGA, x]]] == True

In[12]:= or[not[member[intersection[OMEGA, x_], y_]], not[subclass[OMEGA, x_]]] := True
```

The following theorem improves upon the rewrite rule in a lemma above. (Cf. Suppes, page 227, theorem 57.)

```
In[13]:= SubstTest[and, or[p1, p2], implies[p1, p3], implies[p2, p3],
           not[p3], {p1 → member[hartogs[x], OMEGA], p2 → equal[hartogs[x], OMEGA],
            p3 -> not[member[intersection[OMEGA, image[Q, P[x]]], image[Q, P[x]]]]}] // Reverse

Out[13]= member[intersection[OMEGA, image[Q, P[x]]], image[Q, P[x]]] == False

In[14]:= member[intersection[OMEGA, image[Q, P[x_]]], image[Q, P[x_]]] := False
```

Restatement:

```
In[15]:= member[pair[hartogs[x], x], composite[Q, S]]

Out[15]= False
```

Corollary.

```
In[16]:= member[hartogs[x], hartogs[x]]

Out[16]= False
```

## main theorem

Lemma.

```
In[17]:= SubstTest[implies, and[member[x, y], subclass[y, z]], member[x, z],
           {y → P[x], z → image[Q, P[x]]}] // Reverse

Out[17]= or[member[x, image[Q, P[x]]], not[member[x, V]]] == True

In[18]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. (A better rewrite rule.)

```
In[19]:= equiv[member[x, image[Q, P[x]]], member[x, V]]

Out[19]= True

In[20]:= member[x_, image[Q, P[x_]]] := member[x, V]
```

Lemma.

```
In[21]:= SubstTest[subclass, composite[Id, x], cart[V, y], {x → HARTOGS, y → OMEGA}] // Reverse

Out[21]= subclass[HARTOGS, cart[V, OMEGA]] == True

In[22]:= subclass[HARTOGS, cart[V, OMEGA]] := True
```

Theorem.

```
In[23]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u → pair[x, hartogs[x]], v → HARTOGS, w → cart[V, OMEGA]}] // Reverse

Out[23]= or[member[intersection[OMEGA, image[Q, P[x]]], OMEGA],
           not[member[x, V]], not[member[intersection[OMEGA, image[Q, P[x]]], V]]] == True

In[24]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[25]:= idempotent[composite[id[OMEGA], Q]] // AssertTest

Out[25]= equal[composite[id[OMEGA], Q], composite[id[OMEGA], Q, id[OMEGA], Q]] == True

In[26]:= composite[id[OMEGA], Q, id[OMEGA], Q] := composite[id[OMEGA], Q]
```

Temporary lemma that involves the unknown domain of **HARTOGS**.

```
In[27]:= composite[inverse[E], HARTOGS] // ReifNormality // Reverse

Out[27]= composite[id[OMEGA], Q, inverse[S], id[domain[HARTOGS]]] ==
          composite[inverse[E], HARTOGS]

In[28]:= % /. Equal → SetDelayed
```

Theorem. (A key step.)

```
In[29]:= Assoc[composite[id[OMEGA], Q], composite[id[OMEGA], Q],
          composite[inverse[S], id[domain[HARTOGS]]]]

Out[29]= composite[id[OMEGA], Q, inverse[E], HARTOGS] == composite[inverse[E], HARTOGS]

In[30]:= composite[id[OMEGA], Q, inverse[E], HARTOGS] := composite[inverse[E], HARTOGS]
```

Lemma.

```
In[31]:= member[x, setpart[hartogs[y]]] // AssertTest

Out[31]= member[x, setpart[intersection[OMEGA, image[Q, P[y]]]]] == and[member[x, OMEGA],
           member[x, image[Q, P[y]]], member[intersection[OMEGA, image[Q, P[y]]], V]]

In[32]:= member[x_, setpart[intersection[OMEGA, image[Q, P[y_]]]]] := and[member[x, OMEGA],
           member[x, image[Q, P[y]]], member[intersection[OMEGA, image[Q, P[y]]], V]]
```

Lemma.

```
In[33]:= SubstTest[member, pair[card[x], x], inverse[t], t → Q] // Reverse

Out[33]= member[pair[card[x], x], Q] == member[x, image[Q, OMEGA]]

In[34]:= member[pair[card[x_], x_], Q] := member[x, image[Q, OMEGA]]
```

Theorem. (The cardinality of **hartogs[x]** is not less than **hartogs[x]**.)

```
In[35]:= SubstTest[implies, and[member[pair[u, v], composite[Id, t]],
            member[pair[v, w], composite[Id, s]]], member[pair[u, w], composite[s, t]],
           {s → composite[id[OMEGA], Q], t → composite[inverse[E], HARTOGS],
            u → x, v → card[hartogs[x]], w → hartogs[x]}] // Reverse

Out[35]= or[not[member[x, V]],
           not[member[card[intersection[OMEGA, image[Q, P[x]]]], image[Q, P[x]]]],
           not[member[intersection[OMEGA, image[Q, P[x]]], OMEGA]]] == True

In[36]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[37]:= Map[not, SubstTest[and, implies[p1, p0], implies[p1, p2], implies[p2, or[p3, p4]],
            implies[p1, p6], implies[and[p0, p2, p6], not[p4]], not[implies[p1, p3]],
            {p0 → member[x, V], p1 → member[pair[x, y], HARTOGS], p2 → member[y, OMEGA], p3 →
              equal[y, card[y]], p4 → member[card[y], y], p6 → equal[y, hartogs[x]]}]] // Reverse

Out[37]= or[equal[y, card[y]], not[equal[y, intersection[OMEGA, image[Q, P[x]]]]],
          not[member[x, V]], not[member[y, V]]] == True

In[38]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Main Theorem.

```
In[39]:= Map[empty[composite[Id, complement[#]]] &, SubstTest[class, pair[x, y],
          or[member[y, t], not[equal[y, intersection[OMEGA, image[Q, P[x]]]]],
            not[member[x, V]], not[member[y, V]]], {t → fix[CARD]}]]

Out[39]= subclass[range[HARTOGS], fix[CARD]] == True

In[40]:= subclass[range[HARTOGS], fix[CARD]] := True
```