HARTOGS is a total function

Johan G. F. Belinfante
2010 November 8

```
In[1]:= SetDirectory["l:"]; << goedel.10nov06a

        :Package Title: goedel.10nov06a              2010 November 6 at 10:35 a.m.

        It is now:  2010 Nov 8 at 15:46

        Loading Simplification Rules

        TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

        weightlimit = 40
```

summary

It is shown in this notebook that the Hartogs function is total: **domain[HARTOGS] = V**. As a corollary, it follows that the relation **id[Ω] ∘ Q** is thin. In general, the Hartogs function satisfies **HARTOGS ⋂ (Q ∘ inverse[S]) = 0**. The stronger condition **HARTOGS ⊂ SMALLER** implies the axiom of choice.

derivation

The Hartogs operation is defined for any class:

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

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

Reference:

```
In[3]:=   "F. Hartogs, On the Problem of Well
          -Ordering, Math. Annalen, vol. 76 (1915), pp. 438-443.";
```

Since **CARD** is a function, and the power class of a set is a set, it follows that the class **image[CARD, P[x]]** of cardinalities of subsets of a set **x** is a set. Since the class **fix[CARD]** of all cardinals is a proper class, it cannot be a subclass of this set.

Lemma.

```
In[4]:= Map[not, SubstTest[implies, and[subclass[u, v], member[v, V]],
          member[u, V], {u → fix[CARD], v -> image[CARD, P[setpart[x]]]}]] // Reverse

Out[4]= subclass[fix[CARD], image[CARD, P[setpart[x]]]] == False
```

```
In[5]:= subclass[fix[CARD], image[CARD, P[setpart[x_]]]] := False
```

For any class **x** either **hartogs[x]** is an ordinal or is equal to the class of all ordinals.

Theorem. If **x** is a set, then **hartogs[x]** is not the class **Ω** of all ordinals.

```
In[6]:= Map[equal[OMEGA, image[#, P[setpart[x]]]] &,
          Assoc[id[OMEGA], id[domain[CARD]], Q]] // Reverse

Out[6]= subclass[OMEGA, image[Q, P[setpart[x]]]] == False

In[7]:= subclass[OMEGA, image[Q, P[setpart[x_]]]] := False
```

It follows that if **x** is a set, then **hartogs[x]** is an ordinal.

Corollary. If **x** is a set, then **hartogs[x] ∈ Ω**.

```
In[8]:= SubstTest[or, member[hartogs[t], OMEGA],
          equal[hartogs[t], OMEGA], t → setpart[x]] // Reverse

Out[8]= member[intersection[OMEGA, image[Q, P[setpart[x]]]], OMEGA] == True

In[9]:= member[intersection[OMEGA, image[Q, P[setpart[x_]]]], OMEGA] := True
```

Corollary. (Removing the **setpart** wrapper.)

```
In[10]:= Map[implies[member[x, y], #] &, SubstTest[implies, equal[x, setpart[t]],
           member[intersection[OMEGA, image[Q, P[x]]], OMEGA], t → x]] // Reverse

Out[10]= or[member[intersection[OMEGA, image[Q, P[x]]], OMEGA], not[member[x, y]]] == True

In[11]:= or[member[intersection[OMEGA, image[Q, P[x_]]], OMEGA], not[member[x_, y_]]] := True
```

the function HARTOGS is total

In this section it is shown that the function **HARTOGS = λx . hartogs[x]** is total.

Theorem.

```
In[12]:= SubstTest[implies, member[u, v],
           member[u, V], {u → hartogs[setpart[x]], v → OMEGA}] // Reverse

Out[12]= member[intersection[OMEGA, image[Q, P[setpart[x]]]], V] == True

In[13]:= member[intersection[OMEGA, image[Q, P[setpart[x_]]]], V] := True
```

Corollary. (Removing the **setpart** wrapper.)

```
In[14]:= Map[implies[member[x, y], #] &, SubstTest[implies, equal[x, setpart[t]],
            member[intersection[OMEGA, image[Q, P[x]]], V], t → x]] // Reverse

Out[14]= or[member[intersection[OMEGA, image[Q, P[x]]], V], not[member[x, y]]] == True

In[15]:= or[member[intersection[OMEGA, image[Q, P[x_]]], V], not[member[x_, y_]]] := True
```

Theorem. The function **HARTOGS** is total.

```
In[16]:= SubstTest[class, x, member[setpart[x], t], t → domain[HARTOGS]]

Out[16]= domain[HARTOGS] == V

In[17]:= domain[HARTOGS] := V
```

a fixed point class

In this section two variable-free formulations of the following fact are derived.

```
In[18]:= member[x, hartogs[x]]

Out[18]= member[x, OMEGA]
```

Lemma.

```
In[19]:= member[x, fix[composite[inverse[E], HARTOGS]]] // AssertTest

Out[19]= member[x, fix[composite[inverse[E], HARTOGS]]] == member[x, OMEGA]

In[20]:= member[x_, fix[composite[inverse[E], HARTOGS]]] := member[x, OMEGA]
```

Theorem.

```
In[21]:= fix[composite[inverse[E], HARTOGS]] // Normality

Out[21]= fix[composite[inverse[E], HARTOGS]] == OMEGA

In[22]:= fix[composite[inverse[E], HARTOGS]] := OMEGA
```

Corollary.

```
In[23]:= fix[composite[inverse[HARTOGS], E]] // InvertFixTest

Out[23]= fix[composite[inverse[HARTOGS], E]] == OMEGA

In[24]:= % /. Equal → SetDelayed
```

Lemma.

```
In[25]:= SubstTest[composite, t, id[domain[t]], t -> dif[composite[HARTOGS, id[OMEGA]], E]]

Out[25]= composite[intersection[HARTOGS, complement[E]], id[OMEGA]] == 0
```

```
In[26]:= % /. Equal → SetDelayed
```

Theorem. Another variable-free formulation.

```
In[27]:= SubstTest[empty, dif[u, v], {u → composite[HARTOGS, id[OMEGA]], v → E}]

Out[27]= subclass[composite[HARTOGS, id[OMEGA]], E] == True

In[28]:= subclass[composite[HARTOGS, id[OMEGA]], E] := True
```

the relation id[Ω] ∘ Q is thin

The observation **set[x] ⊂ P[setpart[x]]** allows one to eliminate a **setpart** wrapper in the theorem below.

Lemma.

```
In[29]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
           {t → composite[id[OMEGA], Q], u → set[x], v → P[setpart[x]]}] // Reverse

Out[29]= subclass[intersection[OMEGA, image[Q, set[x]]], image[Q, P[setpart[x]]]] == True

In[30]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem.

```
In[31]:= SubstTest[implies, and[subclass[u, v], member[v, V]], member[u, V],
           {u -> intersection[OMEGA, image[Q, set[x]]], v → hartogs[setpart[x]]}] // Reverse

Out[31]= member[intersection[OMEGA, image[Q, set[x]]], V] == True

In[32]:= member[intersection[OMEGA, image[Q, set[x_]]], V] := True
```

Theorem. The relation **id[Ω] ∘ Q** is thin.

```
In[33]:= SubstTest[class, x, member[setpart[x], t],
          t → domain[VERTSECT[composite[id[OMEGA], Q]]]]

Out[33]= domain[VERTSECT[composite[id[OMEGA], Q]]] == V

In[34]:= domain[VERTSECT[composite[id[OMEGA], Q]]] := V
```

Theorem. Every ordinal is less than some Hartogs number.

```
In[35]:= Map[U, ImageComp[IMAGE[composite[id[OMEGA], Q]], POWER, V]]

Out[35]= U[range[HARTOGS]] == OMEGA

In[36]:= U[range[HARTOGS]] := OMEGA
```

replacement for an APPLY rule

The old **APPLY** rule for **HARTOGS** will be replaced with a simpler one. First the old rule is removed.

```
In[37]:= APPLY[HARTOGS, x_] =.
```

Temporary Lemma. An **APPLY** rule for the special case of a set.

```
In[38]:= ApComp[IMAGE[composite[id[OMEGA], Q]], POWER, setpart[x]] // Reverse

Out[38]= APPLY[HARTOGS, setpart[x]] == intersection[OMEGA, image[Q, P[setpart[x]]]]

In[39]:= APPLY[HARTOGS, setpart[x_]] := intersection[OMEGA, image[Q, P[setpart[x]]]]
```

Lemma. (Removing the **setpart** wrapper.)

```
In[40]:= SubstTest[implies, equal[x, setpart[t]],
           equal[APPLY[HARTOGS, x], hartogs[x]], t → x] // Reverse

Out[40]= or[equal[APPLY[HARTOGS, x], intersection[OMEGA, image[Q, P[x]]]], not[member[x, V]]] ==
          True

In[41]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. Replacement rule.

```
In[42]:= equal[APPLY[HARTOGS, x],
          union[complement[image[V, set[x]]], intersection[OMEGA, image[Q, P[x]]]]]

Out[42]= True

In[43]:= APPLY[HARTOGS, x_] :=
          union[complement[image[V, set[x]]], intersection[OMEGA, image[Q, P[x]]]]
```

replacement for membership rule

The old membership rule for **HARTOGS** can be replaced with a simpler one. The old rule is first removed:

```
In[44]:= member[pair[x_, y_], HARTOGS] =.
```

Lemma.

```
In[45]:= SubstTest[implies, and[member[x, domain[funpart[t]]], equal[y, APPLY[funpart[t], x]]],
           member[pair[x, y], funpart[t]], t → HARTOGS] // Reverse

Out[45]= or[member[pair[x, y], HARTOGS],
           not[equal[y, intersection[OMEGA, image[Q, P[x]]]]], not[member[x, V]]] == True

In[46]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[47]:= SubstTest[implies, member[pair[x, y], funpart[t]],
           equal[y, APPLY[funpart[t], x]], t → HARTOGS] // Reverse // MapNotNot

Out[47]= or[equal[y, intersection[OMEGA, image[Q, P[x]]]],
           not[member[pair[x, y], HARTOGS]]] == True

In[48]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem. Replacement membership rule.

```
In[49]:= equiv[member[pair[x, y], HARTOGS],
           and[equal[y, intersection[OMEGA, image[Q, P[x]]]], member[x, V]]] // not // not

Out[49]= True

In[50]:= member[pair[x_, y_], HARTOGS] :=
          and[equal[y, intersection[OMEGA, image[Q, P[x]]]], member[x, V]]
```

Corollary. Membership rule for composites involving **HARTOGS**.

```
In[51]:= (member[pair[x, y], composite[z, funpart[t]]] // AssertTest) /. t → HARTOGS

Out[51]= member[pair[x, y], composite[z, HARTOGS]] == and[member[x, V],
           member[y, V], member[pair[intersection[OMEGA, image[Q, P[x]]], y], z]]

In[52]:= member[pair[x_, y_], composite[z_, HARTOGS]] := and[member[x, V],
           member[y, V], member[pair[intersection[OMEGA, image[Q, P[x]]], y], z]]
```

Lemma.

```
In[53]:= SubstTest[member, APPLY[funpart[t], setpart[x]],
           range[funpart[t]], t → HARTOGS] // Reverse

Out[53]= member[intersection[OMEGA, image[Q, P[setpart[x]]]], range[HARTOGS]] == True

In[54]:= member[intersection[OMEGA, image[Q, P[setpart[x_]]]], range[HARTOGS]] := True
```

The **setpart** wrapper can be replaced with a sethood literal.

Theorem. If **x** is a set, then **hartogs[x] ∈ range[HARTOGS]**.

```
In[55]:= Map[implies[member[x, y], #] &, SubstTest[implies,
           equal[x, setpart[t]], member[hartogs[x], range[HARTOGS]], t → x]] // Reverse

Out[55]= or[member[intersection[OMEGA, image[Q, P[x]]], range[HARTOGS]], not[member[x, y]]] ==
          True

In[56]:= or[member[intersection[OMEGA, image[Q, P[x_]]], range[HARTOGS]],
           not[member[x_, y_]]] := True
```

Theorem. If **x** is a set, then **hartogs[x]** is a cardinal.

```
In[57]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u → hartogs[setpart[x]], v → range[HARTOGS], w → fix[CARD]}] // Reverse

Out[57]= equal[card[intersection[OMEGA, image[Q, P[setpart[x]]]]],
           intersection[OMEGA, image[Q, P[setpart[x]]]]] == True

In[58]:= card[intersection[OMEGA, image[Q, P[setpart[x_]]]]] :=
          intersection[OMEGA, image[Q, P[setpart[x]]]]
```

Corollary. (Removing the **setpart** wrapper.)

```
In[59]:= Map[implies[member[x, y], #] &, SubstTest[implies,
            equal[x, setpart[t]], equal[card[hartogs[x]], hartogs[x]], t → x]] // Reverse

Out[59]= or[equal[card[intersection[OMEGA, image[Q, P[x]]]],
            intersection[OMEGA, image[Q, P[x]]]], not[member[x, y]]] == True

In[60]:= or[equal[card[intersection[OMEGA, image[Q, P[x_]]]],
            intersection[OMEGA, image[Q, P[x_]]]], not[member[x_, y_]]] := True
```

HARTOGS ⊂ SMALLER ⟹ axch

In general, **hartogs[x]** is neither equipollent to **x**, nor smaller than **x**. However, without the axiom of choice, one cannot conclude that **x** is smaller than **hartogs[x]**.

Theorem.

```
In[61]:= Map[not, SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
            {u → pair[x, hartogs[x]], v → HARTOGS, w → complement[Q]}]] // Reverse

Out[61]= member[pair[x, intersection[OMEGA, image[Q, P[x]]]], Q] == False

In[62]:= member[pair[x_, intersection[OMEGA, image[Q, P[x_]]]], Q] := False
```

It will now be shown that the inclusion **HARTOGS ⊂ SMALLER** implies the axiom of choice.

Lemma.

```
In[63]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u → P[intersection[OMEGA, x]], v → P[OMEGA], w → domain[CARD]}] // Reverse

Out[63]= subclass[P[intersection[OMEGA, x]], image[Q, OMEGA]] == True

In[64]:= subclass[P[intersection[OMEGA, x_]], image[Q, OMEGA]] := True
```

Lemma.

```
In[65]:= Map[implies[#, member[x, domain[CARD]]] &,
          (member[x, fix[y]] // AssertTest) /. y -> composite[inverse[SMALLER], HARTOGS]]

Out[65]= or[member[x, image[Q, OMEGA]],
           not[member[x, fix[composite[inverse[SMALLER], HARTOGS]]]]] == True

In[66]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem.

```
In[67]:= Map[equal[V, #] &,
          complement[dif[fix[composite[inverse[SMALLER], HARTOGS]], domain[CARD]]] // Normality]

Out[67]= subclass[fix[composite[inverse[SMALLER], HARTOGS]], image[Q, OMEGA]] == True

In[68]:= subclass[fix[composite[inverse[SMALLER], HARTOGS]], image[Q, OMEGA]] := True
```

Lemma.

```
In[69]:= SubstTest[implies, subclass[u, v],
           subclass[fix[composite[u, w]], fix[composite[v, w]]],
           {u → inverse[HARTOGS], v → inverse[SMALLER], w → HARTOGS}] // Reverse

Out[69]= or[not[subclass[HARTOGS, SMALLER]],
           subclass[V, fix[composite[inverse[SMALLER], HARTOGS]]]] == True

In[70]:= % /. Equal → SetDelayed
```

Lemma.

```
In[71]:= Map[implies[#, equal[V, image[Q, OMEGA]]] &, SubstTest[and, equal[V, x], subclass[x, y],
            {x -> fix[composite[inverse[SMALLER], HARTOGS]], y -> image[Q, OMEGA]}]] // Reverse

Out[71]= or[equal[V, image[Q, OMEGA]],
           not[equal[V, fix[composite[inverse[SMALLER], HARTOGS]]]]] == True

In[72]:= % /. Equal → SetDelayed
```

Theorem. If **HARTOGS ⊂ SMALLER**, then the axiom of choice is true.

```
In[73]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], implies[p3, p4],
            implies[p4, p5], not[implies[p1, p5]], {p1 → subclass[HARTOGS, SMALLER],
             p2 → subclass[V, fix[composite[inverse[SMALLER], HARTOGS]]],
             p3 → equal[V, fix[composite[inverse[SMALLER], HARTOGS]]],
             p4 -> equal[V, domain[CARD]], p5 → axch}]] // Reverse

Out[73]= or[axch, not[subclass[HARTOGS, SMALLER]]] == True

In[74]:= or[axch, not[subclass[HARTOGS, SMALLER]]] := True
```

These results show that without the axiom of choice, one cannot conclude that **HARTOGS ⊂ SMALLER**, although the following weaker result does hold in general.

```
In[75]:= disjoint[HARTOGS, composite[Q, inverse[S]]]

Out[75]= True
```