

the set of iterated singletons: $0, \{0\}, \{\{0\}\}, \dots$

Johan G. F. Belinfante
2007 July 3

```
In[1]:= SetDirectory["1:"]; << goedel95.02a; << tools.m

:Package Title: goedel95.02a      2007 July 2 at 11:45 p.m.

It is now: 2007 Jul 3 at 18:15

Loading Simplification Rules

TOOLS.M                          Revised 2007 June 25

weightlimit = 40
```

introduction

In this notebook it is shown that the class of iterated singletons: $0, \{0\}, \{\{0\}\}, \dots$ of the empty set is infinite. A historical comment: the existence of the set of iterated singletons of 0 was one of the earliest formulations for the axiom of infinity.

```
In[2]:= "E. Zermelo, Untersuchungen über die Grundlagen der
        Mengenlehre: I., Math. Annalen, vol. 65, (1908), pp. 261-281";
```

In the **GOEDEL** program, a different formulation of the axiom of infinity has been adopted, which uses successors instead of singletons. Zermelo's formulation of the axiom of infinity is:

```
In[3]:= assert[exists[x, and[member[0, x], forall[y, implies[member[y, x], member[set[y], x]]]]]]
Out[3]= True
```

See also:

```
In[4]:= "Patrick Suppes, Axiomatic Set Theory,
        1972, Dover Publications, New York. See p. 138.";
```

subvariance and invariance

To avoid circular reasoning, one needs a definition of finiteness that does not depend on the axiom of infinity. Dedekind's definition is of this nature, but one needs to assume the axiom of choice to show that it is equivalent to the usual concept.

```
In[5]:= complement[fix[composite[PS, Q]]]
Out[5]= DEDEKIND
```

In the **GOEDEL** program, the class **FINITE** of finite sets is defined without reference to natural numbers as the class of all sets that are not members of a set that is subvariant under the proper subset relation **PS**.

```
In[6]:= complement[U[subvar[PS]]]
```

```
Out[6]= FINITE
```

The definition of subvariance resembles that of invariance, but with the inclusion turned around:

```
In[7]:= invariant[x, y]
```

```
Out[7]= subclass[image[x, y], y]
```

```
In[8]:= subvariant[x, y]
```

```
Out[8]= subclass[y, image[x, y]]
```

One can think of subvariance as a recycling condition: if **y** is subvariant under **x**, then any member of **y** is **x**-related to some other member of **y**. For functions, the two concepts are related as follows:

```
In[9]:= implies[and[FUNCTION[x], subvariant[inverse[x], y]], invariant[x, y]]
```

```
Out[9]= True
```

The classes **invar[x]** and **subvar[x]** hold all the invariant and subvariant sets for **x**, respectively.

```
In[10]:= class[y, invariant[x, y]]
```

```
Out[10]= invar[x]
```

```
In[11]:= class[y, subvariant[x, y]]
```

```
Out[11]= subvar[x]
```

The class **invar[x]** is closed under arbitrary intersections and unions. The class **subvar[x]** is closed under arbitrary unions, but not under arbitrary intersections.

```
In[12]:= Uclosure[invar[x]]
```

```
Out[12]= invar[x]
```

```
In[13]:= Aclosure[invar[x]]
```

```
Out[13]= invar[x]
```

```
In[14]:= Uclosure[subvar[x]]
```

```
Out[14]= subvar[x]
```

classes closed under iterated singletons

The class of all sets that satisfy Zermelo's axiom is

```
In[15]:= class[x, and[member[0, x], forall[y, implies[member[y, x], member[set[y], x]]]]]
Out[15]= intersection[complement[P[complement[set[0]]]], invar[SINGLETON]]
```

Zermelo's axiom implies that this class is not empty. The intersection of all such classes is the smallest one:

```
In[16]:= A[intersection[complement[P[complement[set[0]]]], invar[SINGLETON]]]
Out[16]= hull[invar[SINGLETON], set[0]]
```

Since the intersection of any nonempty class of sets is a set, this class is a set:

```
In[17]:= member[hull[invar[SINGLETON], set[0]], V]
Out[17]= True
```

The form of the axiom of infinity adopted in the **GOEDEL** program is similar, but with successors replacing singletons. The corresponding set is the set **omega** of von Neumann natural numbers:

```
In[18]:= hull[invar[SUCC], set[0]]
Out[18]= omega
```

restatement of the basic properties

Observation: The set `hull[invar[SINGLETON], set[0]]` is a set that holds **0** and is invariant under the function **SINGLETON**.

```
In[19]:= member[0, hull[invar[SINGLETON], set[0]]]
Out[19]= True

In[20]:= subclass[image[SINGLETON, hull[invar[SINGLETON], set[0]]],
             hull[invar[SINGLETON], set[0]]]
Out[20]= True
```

Theorem. This minimal successor-invariant set is contained in every other one.

```
In[21]:= SubstTest[implies,
                  and[member[u, x], subclass[x, domain[funpart[t]]], invariant[funpart[t], x]],
                  subclass[hull[invar[funpart[t]], set[u]], x], {u -> 0, t -> SINGLETON}] // Reverse
Out[21]= or[not[member[0, x]], not[subclass[image[SINGLETON, x], x]],
           subclass[hull[invar[SINGLETON], set[0]], x]] == True
```

```
In[22]:= or[not[member[0, x_]], not[subclass[image[SINGLETON, x_], x_]],
      subclass[hull[invar[SINGLETON], set[0]], x_] := True
```

In this notebook further properties of this set are derived. In particular it is shown that this set is infinite. It is easy to show directly that the first few members 0 , $\text{set}[0]$, $\text{set}[\text{set}[0]]$, ... are all distinct. What one needs to be show is that no iterated singleton of 0 ever coincides with any later one. Comment: In the **GOEDEL** program, neither the axiom of regularity nor the axiom of choice is assumed to hold, so it is even conceivable for a set to be equal to its own singleton. It will be shown below that this sort of bizarre behavior does not occur for this particular collection of singletons.

relation to subvar[inverse[E]]

An upper bound for $\text{invar}[\text{SINGLETON}]$ can be derived:

```
In[23]:= SubstTest[implies, subclass[u, v], subclass[subvar[u], subvar[v]],
      {u -> inverse[SINGLETON], v -> inverse[E]}] // Reverse
```

```
Out[23]= subclass[invar[SINGLETON], subvar[inverse[E]]] == True
```

```
In[24]:= subclass[invar[SINGLETON], subvar[inverse[E]]] := True
```

In the terminology of the **GOEDEL** program, the class $\text{subvar}[\text{inverse}[E]]$ holds all sets that are subvariant under $\text{inverse}[E]$, the inverse of the membership relation E . In more familiar terms, the class $\text{subvar}[\text{inverse}[E]]$ holds all sets x that are contained in their own sum class $U[x]$. Each member of such a set is a member of another member.

```
In[25]:= member[x, subvar[inverse[E]]]
```

```
Out[25]= and[member[x, V], subclass[x, U[x]]]
```

This upper bound for $\text{invar}[\text{SINGLETON}]$ yields a lower bound for the set $\text{hull}[\text{invar}[\text{SINGLETON}], \text{set}[0]]$. But since this lower bound is a finite set, this is not particularly useful. (See the appendix.)

```
In[26]:= SubstTest[implies, subclass[u, v], subclass[hull[v, w], hull[u, w]],
      {u -> invar[SINGLETON], v -> subvar[inverse[E]], w -> set[0]}] // Reverse
```

```
Out[26]= subclass[hull[subvar[inverse[E]], set[0]], hull[invar[SINGLETON], set[0]]] == True
```

What the **GOEDEL** program does know is that $\text{subvar}[\text{inverse}[E]]$ is a class whose only finite regular member is 0 .

```
In[27]:= intersection[FINITE, REGULAR, subvar[inverse[E]]]
```

```
Out[27]= set[0]
```

Theorem. The set of iterated singletons of 0 is subvariant under $\text{inverse}[E]$. That is, it is contained in its own sum class.

```
In[28]:= SubstTest[implies, and[member[x, y], subclass[y, z]], member[x, z],
      {x -> hull[invar[SINGLETON], set[0]],
       y -> invar[SINGLETON], z -> subvar[inverse[E]]}] // Reverse
```

```
Out[28]= subclass[hull[invar[SINGLETON], set[0]], U[hull[invar[SINGLETON], set[0]]]] == True
```

```
In[29]:= subclass[hull[invar[SINGLETON], set[0]], U[hull[invar[SINGLETON], set[0]]] := True
```

the set of iterated singletons of 0 is infinite

Theorem. The members of `hull[invar[SINGLETON],set[0]]` are all regular. This rules out in particular that one of its members is its own singleton, and more generally, that any member of the sequence of iterated singletons of **0** is equal to a later one.

```
In[30]:= SubstTest[implies, and[member[u, x], subclass[x, domain[funpart[t]]],
    invariant[funpart[t], x]], subclass[hull[invar[funpart[t]], set[u]], x],
    {u -> 0, t -> SINGLETON, x -> REGULAR}] // Reverse
```

```
Out[30]= subclass[hull[invar[SINGLETON], set[0]], REGULAR] == True
```

```
In[31]:= subclass[hull[invar[SINGLETON], set[0]], REGULAR] := True
```

The class **REGULAR** is defined of sets that do not belong to any set which is subvariant under **E**. This class is its own sum class and its own power class. (By Cantor's theorem it therefore follows that this class is not a set. No set can be its own power set.)

```
In[32]:= complement[U[subvar[E]]]
```

```
Out[32]= REGULAR
```

Corollary. The set `hull[invar[SINGLETON],set[0]]` is itself regular.

```
In[33]:= SubstTest[member, x, P[y], {x -> hull[invar[SINGLETON], set[0]], y -> REGULAR}] // Reverse
```

```
Out[33]= member[hull[invar[SINGLETON], set[0]], REGULAR] == True
```

```
In[34]:= member[hull[invar[SINGLETON], set[0]], REGULAR] := True
```

Theorem. The set of iterated singletons of **0** holds the empty set and otherwise only singletons.

```
In[35]:= SubstTest[implies, and[member[u, x], subclass[x, domain[funpart[t]]],
    invariant[funpart[t], x]], subclass[hull[invar[funpart[t]], set[u]], x],
    {u -> 0, t -> SINGLETON, x -> union[set[0], range[SINGLETON]]} // Reverse
```

```
Out[35]= subclass[hull[invar[SINGLETON], set[0]], union[range[SINGLETON], set[0]] == True
```

```
In[36]:= subclass[hull[invar[SINGLETON], set[0]], union[range[SINGLETON], set[0]] := True
```

Theorem. The set `hull[invar[SINGLETON], set[0]]` is infinite.

```
In[37]:= SubstTest[member, t, intersection[u, v, w],
    {t -> hull[invar[SINGLETON], set[0]], u -> FINITE, v -> REGULAR, w -> subvar[inverse[E]]}]
```

```
Out[37]= member[hull[invar[SINGLETON], set[0]], FINITE] == False
```

```
In[38]:= member[hull[invar[SINGLETON], set[0]], FINITE] := False
```

hereditarily finite property

Lemma. The singleton of a hereditarily finite set is hereditarily finite.

```
In[39]:= Map[equal[V, #] &,
           SubstTest[class, x, implies[member[x, y], member[set[x], y]], y → H[FINITE]]]
```

```
Out[39]= subclass[image[SINGLETON, H[FINITE]], H[FINITE]] == True
```

```
In[40]:= subclass[image[SINGLETON, H[FINITE]], H[FINITE]] := True
```

Theorem. Every iterated singleton of the empty set is hereditarily finite.

```
In[41]:= SubstTest[implies, and[member[u, x], subclass[x, domain[funpart[t]]],
                       invariant[funpart[t], x]], subclass[hull[invar[funpart[t]], set[u]], x],
                  {u → 0, t → SINGLETON, x → H[FINITE]}] // Reverse
```

```
Out[41]= subclass[hull[invar[SINGLETON], set[0]], H[FINITE]] == True
```

```
In[42]:= subclass[hull[invar[SINGLETON], set[0]], H[FINITE]] := True
```

appendix: the set hull[subvar[inverse[E]],set[0]]

In this appendix it is shown that `hull[subvar[inverse[E]], set[0]]` is a finite set, namely `set[0]`.

Lemma.

```
In[43]:= Map[not, SubstTest[and, implies[and[p1, p2], p3], implies[and[p1, p3], p4],
                not[implies[and[p1, p2], p4]], {p1 → subclass[x, omega], p2 → member[U[x], x],
                p3 → member[U[x], omega], p4 → member[x, FINITE]}]] // Reverse
```

```
Out[43]= or[member[x, FINITE], not[member[U[x], x]], not[subclass[x, omega]]] == True
```

```
In[44]:= or[member[x_, FINITE], not[member[U[x_], x_]], not[subclass[x_, omega]]] := True
```

Observation.

```
In[45]:= class[x, member[U[x], x]]
```

```
Out[45]= fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]]
```

Theorem.

```
In[46]:= equal[intersection[complement[set[0]], image[inverse[S], omega]], intersection[
                fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]], P[omega]]] // AssertTest
```

```
Out[46]= equal[intersection[complement[set[0]], image[inverse[S], omega]],
                intersection[fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]]], P[omega]]] == True
```

```
In[47]:= intersection[fix[composite[inverse[E], IMAGE[inverse[BIGCUP]]], P[omega]] :=
intersection[complement[set[0]], image[inverse[S], omega]]
```

Theorem. Every infinite subset of **omega** is subvariant under **inverse[E]**.

```
In[48]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, or[p3, p5]],
implics[and[p1, p3], p4], not[implies[p1, or[p4, p5]]],
{p1 → subclass[x, omega], p2 → subclass[x, OMEGA], p3 → member[U[x], x],
p4 → member[x, FINITE], p5 → subclass[x, U[x]]}] // Reverse
```

```
Out[48]= or[member[x, FINITE], not[subclass[x, omega]], subclass[x, U[x]]] = True
```

```
In[49]:= or[member[x_, FINITE], not[subclass[x_, omega]], subclass[x_, U[x_]]] := True
```

Corollary. (Variable-free reformulation of the above theorem.)

```
In[50]:= subclass[P[omega], union[FINITE, subvar[inverse[E]]] // AssertTest
```

```
Out[50]= subclass[P[omega], union[FINITE, subvar[inverse[E]]] = True
```

```
In[51]:= subclass[P[omega], union[FINITE, subvar[inverse[E]]] := True
```

Lemma. The class of even numbers holds no singletons.

```
In[52]:= AssInt[even, omega, range[SINGLETON]] // Reverse
```

```
Out[52]= intersection[even, range[SINGLETON]] = 0
```

```
In[53]:= intersection[even, range[SINGLETON]] := 0
```

Lemma. The only iterated singleton of **0** that is an even number is **0**.

```
In[54]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
{t → id[even], u → hull[invar[SINGLETON], set[0]],
v → union[set[0], range[SINGLETON]]} // Reverse
```

```
Out[54]= subclass[intersection[even, hull[invar[SINGLETON], set[0]]], set[0]] = True
```

```
In[55]:= % /. Equal → SetDelayed
```

Theorem. (A rewrite rule for this fact.)

```
In[56]:= equal[intersection[even, hull[invar[SINGLETON], set[0]]], set[0]]
```

```
Out[56]= True
```

```
In[57]:= intersection[even, hull[invar[SINGLETON], set[0]]] := set[0]
```

Lemma.

```
In[58]:= SubstTest[implies, subclass[u, v], subclass[A[v], A[u]],
{u → x, v → intersection[y, image[S, set[z]]}] // Reverse
```

```
Out[58]= or[not[subclass[x, y]], not[subclass[z, A[x]]], subclass[hull[y, z], A[x]]] = True
```

```
In[59]:= or[not[subclass[x_, y_]],  
          not[subclass[z_, A[x_]]], subclass[hull[y_, z_], A[x_]]] := True
```

Theorem. A formula for **hull[subvar[inverse[E]], set[0]]**.

```
In[60]:= SubstTest[implies, and[subclass[x, y], subclass[z, A[x]]], subclass[hull[y, z], A[x]],  
                {x → set[even, hull[invar[SINGLETON], set[0]]], y → subvar[inverse[E]], z → set[0]}] //  
                Reverse
```

```
Out[60]= subclass[hull[subvar[inverse[E]], set[0]], set[0]] == True
```

```
In[61]:= % /. Equal → SetDelayed
```

Theorem.

```
In[62]:= equal[hull[subvar[inverse[E]], set[0]], set[0]]
```

```
Out[62]= True
```

```
In[63]:= hull[subvar[inverse[E]], set[0]] := set[0]
```