iterate[K, singleton[0]]

Johan G. F. Belinfante
2003 December 9

```
In[1]:=  << goedel52.t18; << tools.m;

    :Package Title: goedel52.t18      2003 November 28 at 11:15 p.m.

    It is now:  2003 Dec 9 at 10:29

    Loading Simplification Rules

    TOOLS.M                      Revised 2003 November 15

    weightlimit = 40
```

summary

It is shown in this notebook that adding a new element to a finite set increases the number of elements by one. The relation **iterate[K, singleton[0]]** is shown to have as its **n**−th vertical section the class of all sets with cardinality **n** for every natural number **n**. Some additional formulas relating the cardinality function **CARD**, the equipollence relation **Q** and the cover relation **K = dif[PS, composite[PS,PS]]** for finite sets are derived. Here **PS = dif[S, Id]** is the proper subset relation.

a function lemma

The lemma in this section is used in the next section to derive a formula for the restriction of **K** to **omega**.

```
In[2]:=  SubstTest[implies, subclass[u, v], subclass[composite[x, u, y], composite[x, v, y]],
             {u -> composite[K, inverse[K]], v -> Q, x -> id[omega], y -> id[omega]}]

Out[2]=  subclass[composite[id[omega], K, inverse[K], id[omega]], id[omega]] == True

In[3]:=  subclass[composite[id[omega], K, inverse[K], id[omega]], id[omega]] := True

In[4]:=  SubstTest[subclass, composite[x, inverse[x]],
           id[omega], x -> composite[id[omega], K]] // Reverse

Out[4]=  FUNCTION[composite[id[omega], K]] == True

In[5]:=  FUNCTION[composite[id[omega], K]] := True
```

connecting K and SUCC

The first lemma follows from the fact that natural numbers do not belong to themselves.

```
In[6]:= subclass[composite[SINGLETON, id[omega]], DISJOINT] // AssertTest

Out[6]= subclass[composite[SINGLETON, id[omega]], DISJOINT] == True

In[7]:= subclass[composite[SINGLETON, id[omega]], DISJOINT] := True
```

A connection between **K** and **CUP** has been explored previously, but adding an extra factor of **id[DISJOINT]** yields a somewhat cleaner result:

```
In[8]:= composite[CUP, id[DISJOINT],
          id[cart[V, range[SINGLETON]]], inverse[FIRST]] // RelnRenormality

Out[8]= composite[CUP, id[composite[id[range[SINGLETON]], DISJOINT]], inverse[FIRST]] == K

In[9]:= composite[CUP, id[composite[id[range[SINGLETON]], DISJOINT]], inverse[FIRST]] := K
```

From this one derives an inclusion:

```
In[10]:= SubstTest[implies, subclass[u, v], subclass[composite[x, u, y], composite[x, v, y]],
             {u -> id[composite[SINGLETON, id[omega]]],
           v -> id[composite[id[range[SINGLETON]], DISJOINT]], x -> CUP, y -> inverse[FIRST]}]

Out[10]= subclass[composite[id[omega], SUCC], K] == True

In[11]:= subclass[composite[id[omega], SUCC], K] := True
```

As a corollary, one obtains a formula for the restriction of the cover relation **K** to the set **omega** of natural numbers.

```
In[12]:= SubstTest[implies, and[subclass[x, y], FUNCTION[y]],
          equal[x, composite[y, id[domain[x]]]],
             {x -> composite[id[omega], SUCC], y -> composite[id[omega], K]}]

Out[12]= equal[composite[id[omega], SUCC], composite[id[omega], K, id[omega]]] == True

In[13]:= composite[id[omega], K, id[omega]] := composite[id[omega], SUCC]
```

The inverse formula will also be used below.

```
In[14]:= composite[id[omega], inverse[K], id[omega]] // DoubleInverse

Out[14]= composite[id[omega], inverse[K], id[omega]] == composite[inverse[SUCC], id[omega]]

In[15]:= composite[id[omega], inverse[K], id[omega]] := composite[inverse[SUCC], id[omega]]
```

another function lemma

To derive the main theorem in this notebook, another function lemma is used:

```
In[16]:= SubstTest[implies, subclass[x, y], subclass[composite[u, x, v], composite[u, y, v]],
             {u -> composite[id[omega], CARD], v -> composite[inverse[CARD], id[omega]],
                 x -> composite[K, inverse[K]], y -> Q}]

Out[16]= subclass[
           composite[id[omega], CARD, K, inverse[K], inverse[CARD], id[omega]], id[omega]] == True

In[17]:= subclass[composite[id[omega], CARD, K,
            inverse[K], inverse[CARD], id[omega]], id[omega]] := True
```

```
In[18]:= SubstTest[subclass, composite[x, inverse[x]], id[omega],
             x -> composite[id[omega], CARD, inverse[K]]] // Reverse

Out[18]= FUNCTION[composite[id[omega], CARD, inverse[K]]] == True

In[19]:= FUNCTION[composite[id[omega], CARD, inverse[K]]] := True

In[20]:= SubstTest[implies, subclass[u, v], subclass[composite[u, w], composite[v, w]],
             {u -> restrict[inverse[K], omega, omega], v -> inverse[K], w -> CARD}]

Out[20]= subclass[composite[inverse[SUCC], id[omega], CARD], composite[inverse[K], CARD]] == True

In[21]:= subclass[composite[inverse[SUCC], id[omega], CARD],
           composite[inverse[K], CARD]] := True

In[22]:= SubstTest[implies, subclass[u, v], subclass[composite[w, u], composite[w, v]],
             {u -> CARD, v -> Q, w -> composite[id[omega], inverse[K]]}]

Out[22]= subclass[composite[id[omega], inverse[K], CARD], composite[CARD, inverse[K]]] == True

In[23]:= subclass[composite[id[omega], inverse[K], CARD], composite[CARD, inverse[K]]] := True

In[24]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
             {u -> composite[inverse[SUCC], id[omega], CARD],
           v -> composite[id[omega], inverse[K], CARD],
                 w -> composite[id[omega], CARD, inverse[K]]}]

Out[24]= subclass[composite[inverse[SUCC], id[omega], CARD], composite[CARD, inverse[K]]] == True

In[25]:= subclass[composite[inverse[SUCC], id[omega], CARD],
           composite[CARD, inverse[K]]] := True
```

CARD, Q and FINITE

In this section further simplification rules involving the cardinality function **CARD**, the equipollence relation **Q** and the class **FINITE** of finite sets are derived.

```
In[26]:= equal[composite[CARD, id[FINITE]], composite[id[omega], CARD]] // AssertTest

Out[26]= equal[composite[CARD, id[FINITE]], composite[id[omega], CARD]] == True

In[27]:= composite[CARD, id[FINITE]] := composite[id[omega], CARD]
```

Lemma. The class of finite sets is contained in the domain of the function **CARD**.

```
In[28]:= equal[intersection[FINITE, image[Q, OMEGA]], FINITE]

Out[28]= True

In[29]:= intersection[FINITE, image[Q, OMEGA]] := FINITE
```

Equipollence for finite sets is equivalent to equality of their cardinalities:

```
In[30]:= Assoc[Q, id[image[Q, OMEGA]], id[FINITE]]

Out[30]= composite[Q, id[FINITE]] == composite[inverse[CARD], id[omega], CARD]
```

```
In[31]:= composite[Q, id[FINITE]] := composite[inverse[CARD], id[omega], CARD]

In[32]:= composite[id[FINITE], Q] // DoubleInverse

Out[32]= composite[id[FINITE], Q] == composite[inverse[CARD], id[omega], CARD]

In[33]:= composite[id[FINITE], Q] := composite[inverse[CARD], id[omega], CARD]
```

Lemma. The cardinality of a set is a nonzero natural number if and only if the set is finite and nonempty.

```
In[34]:= image[inverse[CARD], intersection[omega, complement[singleton[0]]]] // Normality

Out[34]= image[inverse[CARD], intersection[omega, complement[singleton[0]]]] ==
          intersection[FINITE, complement[singleton[0]]]

In[35]:= image[inverse[CARD], intersection[omega, complement[singleton[0]]]] :=
          intersection[FINITE, complement[singleton[0]]]
```

proof of the main theorem

An explicit formula for the function **composite[id[omega], CARD, inverse[K]]** follows:

```
In[36]:= SubstTest[implies, and[subclass[x, y], FUNCTION[y]],
          equal[x, composite[y, id[domain[x]]]]],
             {x -> composite[inverse[SUCC], id[omega], CARD],
           y -> composite[id[omega], CARD, inverse[K]]}]

Out[36]= equal[composite[id[omega], CARD, inverse[K]],
           composite[inverse[SUCC], id[omega], CARD]] == True

In[37]:= composite[id[omega], CARD, inverse[K]] := composite[inverse[SUCC], id[omega], CARD]
```

The next step requires the inverse formula.

```
In[38]:= composite[K, inverse[CARD], id[omega]] // DoubleInverse

Out[38]= composite[K, inverse[CARD], id[omega]] == composite[inverse[CARD], id[omega], SUCC]

In[39]:= composite[K, inverse[CARD], id[omega]] := composite[inverse[CARD], id[omega], SUCC]
```

The final step is to use the uniqueness theorem for **iterate**.

```
In[40]:= SubstTest[implies, and[equal[composite[w, SUCC], composite[u, w]],
                 equal[image[w, singleton[0]], v]],
             equal[composite[w, id[omega]], iterate[u, v]],
             {u -> K, v -> singleton[0], w -> composite[Q, id[omega]]}]

Out[40]= equal[composite[inverse[CARD], id[omega]], iterate[K, singleton[0]]] == True
```

This formula implies that for each natural number **n**, the class of sets with cardinality **succ[n]** is the image under **K** of the class of sets with cardinality **n**.

```
In[41]:= iterate[K, singleton[0]] := composite[inverse[CARD], id[omega]]
```

a related theorem

In this section, a formula is derived for another function obtained by replacing **inverse[K]** with **K**.

```
In[42]:= SubstTest[subclass, composite[x, inverse[x]], id[omega],
             x -> composite[id[omega], CARD, K]] // Reverse

Out[42]= FUNCTION[composite[id[omega], CARD, K]] == True

In[43]:= FUNCTION[composite[id[omega], CARD, K]] := True

In[44]:= SubstTest[implies, subclass[u, v], subclass[composite[u, w], composite[v, w]],
             {u -> restrict[K, omega, omega], v -> K, w -> CARD}]

Out[44]= subclass[composite[id[omega], SUCC, CARD], composite[K, CARD]] == True

In[45]:= subclass[composite[id[omega], SUCC, CARD], composite[K, CARD]] := True

In[46]:= SubstTest[implies, subclass[u, v], subclass[composite[w, u], composite[w, v]],
             {u -> CARD, v -> Q, w -> composite[id[omega], K]}]

Out[46]= subclass[composite[id[omega], K, CARD], composite[CARD, K]] == True

In[47]:= subclass[composite[id[omega], K, CARD], composite[CARD, K]] := True

In[48]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
             {u -> composite[id[omega], SUCC, CARD], v -> composite[id[omega], K, CARD],
                 w -> composite[CARD, K]}]

Out[48]= subclass[composite[id[omega], SUCC, CARD], composite[CARD, K]] == True

In[49]:= subclass[composite[id[omega], SUCC, CARD], composite[CARD, K]] := True
```

An explicit formula for the function **composite[id[omega], CARD, K]** follows:

```
In[50]:= SubstTest[implies, and[subclass[x, y], FUNCTION[y]],
          equal[x, composite[y, id[domain[x]]]],
             {x -> composite[id[omega], SUCC, CARD], y -> composite[id[omega], CARD, K]}]

Out[50]= equal[composite[id[omega], CARD, K], composite[id[omega], SUCC, CARD]] == True

In[51]:= composite[id[omega], CARD, K] := composite[id[omega], SUCC, CARD]
```

Corollary. The cover relation commutes with the restriction of the equipollence relation to finite sets.

```
In[52]:= commute[K, composite[inverse[CARD], id[omega], CARD]]

Out[52]= True
```