finite functions

Johan G. F. Belinfante
2012 February 22


```
In[1]:= SetDirectory["l:"]; << goedel.12feb21a

:Package Title: goedel.12feb21a            2012 February 21 at 11:20 a.m.

Loading takes about thirteen minutes, half that time due to builtin pauses.

It is now:  2012 Feb 22 at 7:57

Loading Simplification Rules

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

weightlimit = 40

Loading completed.

It is now:  2012 Feb 22 at 8:10
```

---

summary

It was shown 2005 November 20 in the posted notebook **acdorafu.nb** that when the axiom of choice holds one can obtain an explicit formula for **image[DORA, FUNS]** in terms of the equipollence relation **Q**. In particular, the axiom of choice implies that the upper bound **image[DORA, FUNS] ⊂ Q ∘ inverse[S]** holds.

```
In[4]:= implies[axch, subclass[image[DORA, FUNS], composite[Q, inverse[S]]]]

Out[4]= True
```

It is shown below that this upper bound implies that when **axch** holds, the cardinality of the domain of a function can not be less than that of its range. It was also shown in the same notebook that whether or not the axiom of choice holds, the following lower bound for **image[DORA, FUNS]** holds.

```
In[3]:= subclass[composite[id[complement[set[0]]], Q, inverse[S]], image[DORA, FUNS]]

Out[3]= True
```

For finite functions, the axiom of choice is not needed. The theorem of finite choice was used 2005 November 5 in the posted notebook **fin-fu.nb** to show that the cardinality of the domain of a finite function can not be less than that of its range.

```
In[5]:= implies[and[FUNCTION[x], member[x, FINITE]],
         not[member[card[domain[x]], card[range[x]]]]]

Out[5]= True
```

Some related results are rederived here using essentially the same technique.

---

axch results

Statements about what the axiom of choice implies are derived in this section.

Lemma. (Introducing a variable **x**. The variable **y** is also added for the sake of generality.)

```
In[6]:= Map[implies[member[x, y], #] &,
          SubstTest[implies, and[member[x, u], subclass[u, v]], member[x, v],
           {u → FUNS, v → image[inverse[DORA], composite[Q, inverse[S]]]}]] // Reverse

Out[6]= or[member[range[x], image[Q, P[domain[x]]]], not[FUNCTION[x]], not[member[x, y]],
          not[subclass[image[DORA, FUNS], composite[Q, inverse[S]]]]] == True

In[7]:= (% /. {x → x_, y → y}) /. Equal → SetDelayed
```

Theorem. The axiom of choice implies that if **x** is a small function, then **range[x] ∈ image[Q, domain[x]]**.

```
In[8]:= Map[not, SubstTest[and, implies[p0, p2], implies[and[p1, p2], p3],
           not[implies[and[p0, p1], p3]], {p0 → axch, p1 → and[member[x, y], FUNCTION[x]],
            p2 -> subclass[image[DORA, FUNS], composite[Q, inverse[S]]],
            p3 -> member[range[x], image[Q, P[domain[x]]]]}]] // Reverse

Out[8]= or[member[range[x], image[Q, P[domain[x]]]],
          not[axch], not[FUNCTION[x]], not[member[x, y]]] == True

In[9]:= or[member[range[x_], image[Q, P[domain[x_]]]],
          not[axch], not[FUNCTION[x_]], not[member[x_, y_]]] := True
```

Comment. When the axiom of choice is assumed, every set is equipollent to an ordinal. The above theorem can therefore be restated using cardinal numbers.

```
In[10]:= implies[axch, equal[image[Q, OMEGA], V]]

Out[10]= True
```

The following technical observation helps to explain why two steps in the derivation of the corollary below can be omitted.

```
In[11]:= or[member[range[x], image[Q, P[domain[x]]]],
           not[axch], not[FUNCTION[x]], not[member[domain[x], V]]] // not // not

Out[11]= True
```

In the derivation of the following corollary two proof steps are omitted, as indicated with **(* ⋯ *)**. The rewrite rules in the **GOEDEL** program make up for these missing steps.

Corollary. The axiom of choice implies that the cardinality of the domain of a function is not less than that of its range.

```
In[12]:= Map[not, SubstTest[and, (*implies[and[p0,p1],or[p2,p3]],*) implies[p2, p5],
            implies[p3, p4], implies[p4, p5], not[implies[and[p0, p1], p5]],
            {p0 → axch, p1 → FUNCTION[x], p2 → member[range[x], image[Q, P[domain[x]]]],
             p3 → not[member[domain[x], V]], p4 → not[member[domain[x], image[Q, OMEGA]]],
             p5 → not[member[card[domain[x]], card[range[x]]]]}]] // Reverse

Out[12]= or[not[axch], not[FUNCTION[x]], not[member[card[domain[x]], card[range[x]]]]] == True

In[13]:= or[not[axch], not[FUNCTION[x_]],
          not[member[card[domain[x_]], card[range[x_]]]]] := True
```

finite functions

In this section, the axiom of choice is not assumed, but the additional hypothesis that **x** be finite is added. The theorem of finite choice implies that any finite set admits a cross-section. In the following lemma, the proposition **p2** is a temporary hypothesis that says **y** is a cross-section of **inverse[x]**. (The requirement that **y** be a set has purposely been omitted from **p2** to avoid the intervention of rewrite rules about mappings.)

Lemma.

```
In[14]:= Map[not, SubstTest[and, (*implies[p2,p3],implies[and[p1,p2],p4],
             implies[and[p1,p2],p5],*) implies[and[p2, p4, p5], p6],
            not[implies[and[p1, p2], p6]], {p1 → and[member[x, FINITE], FUNCTION[x]],
             p2 -> and[FUNCTION[y], subclass[y, inverse[x]]],
             p3 → subclass[range[y], domain[x]], p4 → FUNCTION[inverse[y]],
             p5 → member[y, FINITE], p6 → member[pair[domain[y], range[y]], Q]}]] // Reverse

Out[14]= or[member[pair[domain[y], range[y]], Q], not[FUNCTION[x]],
          not[FUNCTION[y]], not[member[x, FINITE]], not[subclass[y, inverse[x]]]] == True

In[15]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

The execution time for the following theorem was reduced to 13.5 seconds by omitting four proof steps, as indicated. Note that the variable **y** occurs only in the hypothesis, and not in the conclusion.

Theorem. If **y** is a cross-section of the inverse of a finite function **x**, then **range[x] ∈ image[Q, P[domain[x]]**.

```
In[16]:= Map[not, SubstTest[and,
           (*implies[and[p1,p2],p3],implies[and[p1,p2],p4],implies[and[p1,p2],p5],*)
           implies[and[p2, p4, p5], p6], implies[p2, p7], (*implies[and[p2,p6,p7],p8],*)
           not[implies[and[p1, p2], p8]], {p1 → and[member[x, FINITE], FUNCTION[x]],
            p2 -> and[FUNCTION[y], subclass[y, inverse[x]], equal[domain[y], range[x]]],
            p3 → member[pair[domain[y], range[y]], Q], p4 -> FUNCTION[inverse[y]],
            p5 -> member[y, FINITE], p6 -> member[pair[domain[y], range[y]], Q],
            p7 -> subclass[range[y], domain[x]],
            p8 → member[range[x], image[Q, P[domain[x]]]]}]] // Reverse

Out[16]= or[member[range[x], image[Q, P[domain[x]]]],
           not[equal[domain[y], range[x]]], not[FUNCTION[x]], not[FUNCTION[y]],
           not[member[x, FINITE]], not[subclass[y, inverse[x]]]] == True

In[17]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

At this point, the variable **y** can be eliminated using a combination of **reify** and **case**.

Lemma. (Eliminating the variable **y**. A redundant literal remains.)

```
In[18]:= Map[equal[V, domain[#]] &,
          SubstTest[reify, y, case[or[member[range[x], image[Q, P[domain[x]]]],
            not[equal[domain[y], range[x]]], not[FUNCTION[x]], not[FUNCTION[y]],
            not[member[x, t]], not[subclass[y, inverse[x]]]]]], t → FINITE]]

Out[18]= or[equal[0, X[inverse[x]]], member[range[x], image[Q, P[domain[x]]]],
           not[FUNCTION[x]], not[member[x, FINITE]]] == True

In[19]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Since **x** is finite, the hypothesis that **inverse[x]** admits a cross-section follows from the theorem of finite choice. The final step is to eliminate this redundant literal.

Main Theorem. If **x** is a finite function, then **range[x] ∈ image[Q, P[domain[x]]]**.

```
In[20]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3], not[implies[p1, p3]],
           {p1 → and[FUNCTION[x], member[x, FINITE]], p2 → not[equal[0, X[inverse[x]]]],
            p3 → member[range[x], image[Q, P[domain[x]]]]}]] // Reverse

Out[20]= or[member[range[x], image[Q, P[domain[x]]]],
           not[FUNCTION[x]], not[member[x, FINITE]]] == True

In[21]:= or[member[range[x_], image[Q, P[domain[x_]]]],
           not[FUNCTION[x_]], not[member[x_, FINITE]]] := True
```

Corollary. A variable-free reformulation.

```
In[22]:= Map[equal[V, #] &, complement[intersection[FINITE, FUNS,
            complement[image[inverse[DORA], composite[Q, inverse[S]]]]]]] // Normality]

Out[22]= subclass[image[DORA, intersection[FINITE, FUNS]], composite[Q, inverse[S]]] == True

In[23]:= subclass[image[DORA, intersection[FINITE, FUNS]], composite[Q, inverse[S]]] := True
```

Observation. Any finite set is equipollent to a natural number. (Comment. This is a theorem, not the definition of finiteness.)

```
In[24]:= implies[member[x, FINITE], member[x, image[Q, omega]]]

Out[24]= True
```

The main theorem can therefore be restated in terms of cardinal numbers. This can be done automatically by introducing the **fin** wrapper. A restatement as a single literal can be achieved by simultanuously introducing the **funpart** wrapper.

Lemma. Restatement of the theorem using the compound wrapper **funpart[fin[x]]**.

```
In[25]:= Map[and[member[card[funpart[fin[x]]], card[range[funpart[fin[x]]]]], not[#]] &,
           SubstTest[implies, and[FUNCTION[t], member[t, FINITE]],
            member[range[t], image[Q, P[domain[t]]]]], t → funpart[fin[x]]]] // Reverse

Out[25]= member[card[funpart[fin[x]]], card[range[funpart[fin[x]]]]] == False

In[26]:= member[card[funpart[fin[x_]]], card[range[funpart[fin[x_]]]]] := False
```

Note that the cardinality of the domain of a function here is rewritten to the cardinality of the function itself by the following rewrite rule.

```
In[27]:= card[domain[funpart[x]]]

Out[27]= card[funpart[x]]
```

Removing both wrappers, one can of course obtain the statement that is already in the **GOEDEL** program. The following statement also holds.

Theorem. The cardinality of a finite function can not be less than that of its range.

```
In[35]:= SubstTest[implies, equal[x, funpart[fin[t]]],
           not[member[card[x], card[range[x]]]], t → x] // Reverse

Out[35]= or[not[FUNCTION[x]], not[member[x, FINITE]],
           not[member[card[x], card[range[x]]]]] == True

In[36]:= or[not[FUNCTION[x_]], not[member[x_, FINITE]],
           not[member[card[x_], card[range[x_]]]]] := True
```