equicardinality

Johan G. F. Belinfante
2012 February 27

```
In[1]:= SetDirectory["l:"]; << goedel.12feb24a

        :Package Title: goedel.12feb24a              2012 February 24 at 8:45 p.m.

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

        It is now:  2012 Feb 27 at 14:50

        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 27 at 15:3
```

summary

If the axiom of choice holds, there is no difference between the equipollence relation **Q** and the equicardinality relation **inverse[CARD] ∘ CARD**. By default, **axch** is not assumed in the **GOEDEL** program, so one needs to distinguish these two relations. The subset relation **S** commutes with **Q**, but when the axiom of choice is not assumed, the subset relation **S** need not commute with the equicardinality relation **inverse[CARD] ∘ CARD**. However, the equicardinality relation does subcommute with the subset relation. This is because the domain of the cardinality function is invariant under **inverse[S]**.

```
In[2]:= invariant[inverse[S], domain[CARD]]

Out[2]= True
```

In other words, if a set is equipollent to an ordinal, then so is any subset.

```
In[3]:= implies[and[subclass[x, y], member[y, image[Q, OMEGA]]], member[x, image[Q, OMEGA]]]

Out[3]= True
```

The statements that **Q** commutes with **S** and with **inverse[S]** are formulated in the **GOEDEL** program in an unsymmetric fashion in that the following two rewrite rules move both **S** and **inverse[S]** from left to right:

```
In[4]:= composite[S, Q]

Out[4]= composite[Q, S]
```

```
In[5]:= composite[inverse[S], Q]

Out[5]= composite[Q, inverse[S]]
```

For the equicardinality relation, in addition to putting **S** or its inverse on the left or right of **inverse[CARD] ∘ CARD** one also has the superior option of placing it in between **inverse[CARD]** and **CARD**. In this notebook precise statements about all these matters are derived, as well as various new simplification rules that hold independently of **axch**.

a statement equivalent to axch

The axiom of choice equivalent to the statement that every set is equipollent to an ordinal.

```
In[6]:= equiv[axch, equal[V, image[Q, OMEGA]]]

Out[6]= True
```

Lemma.

```
In[7]:= SubstTest[implies, equal[u, v], equal[fix[u], fix[v]],
          {u → Q, v -> composite[inverse[CARD], CARD]}] // Reverse

Out[7]= or[axch, not[equal[Q, composite[inverse[CARD], CARD]]]] == True

In[8]:= % /. Equal → SetDelayed
```

Lemma. The equipollence relation and the equicardinality relation are equal if **axch** holds.

```
In[9]:= implies[axch, equal[Q, composite[inverse[CARD], CARD]]] // AssertTest

Out[9]= or[equal[Q, composite[inverse[CARD], CARD]], not[axch]] == True

In[10]:= % /. Equal → SetDelayed
```

Theorem. The statement that the equipollence and equicardinality relations are equal is equivalent to **axch**.

```
In[11]:= equiv[equal[Q, composite[inverse[CARD], CARD]], axch]

Out[11]= True

In[12]:= equal[Q, composite[inverse[CARD], CARD]] := axch
```

a key simplification rule

Lemma. An inclusion.

```
In[13]:= SubstTest[subclass, x, composite[id[y], z], {x -> composite[inverse[CARD], S, CARD],
           y -> domain[CARD], z → composite[Q, S]}] // Reverse

Out[13]= subclass[composite[inverse[CARD], S, CARD], composite[inverse[CARD], CARD, S]] == True
```

```
In[14]:= % /. Equal → SetDelayed
```

Lemma. An equation.

```
In[15]:= equal[composite[inverse[CARD], S, CARD],
           composite[inverse[CARD], CARD, S]] // AssertTest

Out[15]= equal[composite[inverse[CARD], CARD, S], composite[inverse[CARD], S, CARD]] == True

In[16]:= % /. Equal → SetDelayed
```

A better equation can be derived.

Theorem. A simplification rule for moving **CARD** past **S**.

```
In[17]:= SubstTest[implies, equal[u, v], equal[composite[t, u], composite[t, v]],
           {t → CARD, u -> composite[inverse[CARD], CARD, S],
            v -> composite[inverse[CARD], S, CARD]}] // Reverse

Out[17]= equal[composite[CARD, S], composite[id[fix[CARD]], S, CARD]] == True

In[18]:= composite[CARD, S] := composite[id[fix[CARD]], S, CARD]
```

Corollary.

```
In[19]:= composite[inverse[S], inverse[CARD]] // DoubleInverse

Out[19]= composite[inverse[S], inverse[CARD]] ==
          composite[inverse[CARD], inverse[S], id[fix[CARD]]]

In[20]:= composite[inverse[S], inverse[CARD]] :=
          composite[inverse[CARD], inverse[S], id[fix[CARD]]]
```

Theorem. A rewrite rule that moves the inverse of the cardinality function to the left.

```
In[21]:= Map[composite[#, inverse[CARD]] &,
          Assoc[composite[id[image[Q, OMEGA]], S], id[domain[CARD]], Q]]

Out[21]= composite[id[image[Q, OMEGA]], S, inverse[CARD]] ==
          composite[inverse[CARD], S, id[fix[CARD]]]

In[22]:= composite[id[image[Q, OMEGA]], S, inverse[CARD]] :=
          composite[inverse[CARD], S, id[fix[CARD]]]
```

Corollary. A rewrite rule that moves the cardinality function to the right.

```
In[26]:= composite[CARD, inverse[S], id[image[Q, OMEGA]]] // DoubleInverse

Out[26]= composite[CARD, inverse[S], id[image[Q, OMEGA]]] ==
          composite[id[fix[CARD]], inverse[S], CARD]

In[27]:= composite[CARD, inverse[S], id[image[Q, OMEGA]]] :=
          composite[id[fix[CARD]], inverse[S], CARD]
```

Corollary. This special result is not needed in the rest of this notebook.

```
In[29]:= Map[composite[CARD, #] &,
          Assoc[id[FINITE], id[image[Q, OMEGA]], composite[S, inverse[CARD]]]]

Out[29]= composite[id[omega], S, id[fix[CARD]]] == composite[id[omega], S, id[omega]]

In[30]:= composite[id[omega], S, id[fix[CARD]]] := composite[id[omega], S, id[omega]]
```

subcommute property

Corollary. The equipollence relation subcommutes with the subset relation.

```
In[31]:= SubstTest[subclass, composite[id[x], y], y,
           {x → domain[CARD], y → composite[S, inverse[CARD], CARD]}] // Reverse

Out[31]= subclass[composite[inverse[CARD], S, CARD], composite[S, inverse[CARD], CARD]] == True

In[32]:= subclass[composite[inverse[CARD], S, CARD], composite[S, inverse[CARD], CARD]] := True
```

Restatement.

```
In[33]:= subcommute[composite[inverse[CARD], CARD], S]

Out[33]= True
```

Corollary. The relation **inverse[S]** subcommutes with the equicardinality relation.

```
In[34]:= SubstTest[subcommute, inverse[x], inverse[y],
          {x -> composite[inverse[CARD], CARD], y -> inverse[S]}]

Out[34]= subclass[composite[inverse[CARD], inverse[S], CARD],
           composite[inverse[CARD], CARD, inverse[S]]] == True

In[35]:= subclass[composite[inverse[CARD], inverse[S], CARD],
           composite[inverse[CARD], CARD, inverse[S]]] := True
```

Restatement.

```
In[36]:= subcommute[inverse[S], composite[inverse[CARD], CARD]]

Out[36]= True
```

another statement equivalent to axch

Lemma. If **axch** holds, then the equicardinality relation commutes with the subset relation.

```
In[37]:= SubstTest[implies, and[equal[u, v], commute[u, w]], commute[v, w],
           {u → Q, v -> composite[inverse[CARD], CARD], w → S}] // Reverse

Out[37]= or[equal[composite[S, inverse[CARD], CARD], composite[inverse[CARD], S, CARD]],
           not[axch]] == True

In[38]:= % /. Equal → SetDelayed
```

Lemma. If the equicardinality relation commutes with the subset relation, then **axch** holds.

```
In[39]:= SubstTest[implies, equal[u, v],
           equal[range[u], range[v]], {u -> composite[S, inverse[CARD], CARD],
            v -> composite[inverse[CARD], CARD, S]}] // Reverse

Out[39]= or[axch, not[
            equal[composite[S, inverse[CARD], CARD], composite[inverse[CARD], S, CARD]]]] == True

In[40]:= % /. Equal → SetDelayed
```

Theorem. The statement that the equicardinality relation commutes with the subset relation is equivalent to **axch**.

```
In[41]:= equiv[
          equal[composite[S, inverse[CARD], CARD], composite[inverse[CARD], S, CARD]], axch]

Out[41]= True

In[42]:= equal[composite[S, inverse[CARD], CARD], composite[inverse[CARD], S, CARD]] := axch
```

Restatement.

```
In[43]:= commute[S, composite[inverse[CARD], CARD]]

Out[43]= axch
```

rules that convert Q to inverse[CARD] ∘ CARD

By adding a factor of **id[image[Q, Ω]]**, one can use double inversion to convert an expression involving the equipollence relation to one involving the equicardinality relation

Theorem.

```
In[44]:= composite[Q, S, id[image[Q, OMEGA]]] // DoubleInverse

Out[44]= composite[Q, S, id[image[Q, OMEGA]]] == composite[S, inverse[CARD], CARD]

In[45]:= composite[Q, S, id[image[Q, OMEGA]]] := composite[S, inverse[CARD], CARD]
```

Theorem.

```
In[46]:= composite[Q, inverse[S], id[image[Q, OMEGA]]] // DoubleInverse

Out[46]= composite[Q, inverse[S], id[image[Q, OMEGA]]] ==
          composite[inverse[CARD], inverse[S], CARD]

In[47]:= composite[Q, inverse[S], id[image[Q, OMEGA]]] :=
          composite[inverse[CARD], inverse[S], CARD]
```

rules that remove a factor of S or its inverse

Theorem.

```
In[48]:= Assoc[composite[Q, S], id[image[Q, OMEGA]], S] // Reverse

Out[48]= composite[S, inverse[CARD], S, CARD] == composite[S, inverse[CARD], CARD]

In[49]:= composite[S, inverse[CARD], S, CARD] := composite[S, inverse[CARD], CARD]
```

Corollary.

```
In[50]:= composite[inverse[CARD], inverse[S], CARD, inverse[S]] // DoubleInverse

Out[50]= composite[inverse[CARD], inverse[S], CARD, inverse[S]] ==
          composite[inverse[CARD], CARD, inverse[S]]

In[51]:= composite[inverse[CARD], inverse[S], CARD, inverse[S]] :=
          composite[inverse[CARD], CARD, inverse[S]]
```

rules that remove a factor of CARD or its inverse

Theorem.

```
In[52]:= Map[inverse, Assoc[inverse[CARD], composite[Q, S], inverse[CARD]]]

Out[52]= composite[CARD, inverse[S], CARD] == composite[id[fix[CARD]], inverse[S], CARD]

In[53]:= composite[CARD, inverse[S], CARD] := composite[id[fix[CARD]], inverse[S], CARD]
```

Corollary.

```
In[54]:= composite[inverse[CARD], S, inverse[CARD]] // DoubleInverse

Out[54]= composite[inverse[CARD], S, inverse[CARD]] == composite[inverse[CARD], S, id[fix[CARD]]]

In[55]:= composite[inverse[CARD], S, inverse[CARD]] :=
          composite[inverse[CARD], S, id[fix[CARD]]]
```

rules that remove an identity factor

Theorem.

```
In[56]:= Assoc[inverse[CARD], composite[id[image[Q, OMEGA]], S], id[image[Q, OMEGA]]] // Reverse

Out[56]= composite[inverse[CARD], S, id[image[Q, OMEGA]]] == composite[inverse[CARD], S]

In[57]:= composite[inverse[CARD], S, id[image[Q, OMEGA]]] := composite[inverse[CARD], S]
```

Corollary.

```
In[58]:= composite[id[image[Q, OMEGA]], inverse[S], CARD] // DoubleInverse

Out[58]= composite[id[image[Q, OMEGA]], inverse[S], CARD] == composite[inverse[S], CARD]

In[59]:= composite[id[image[Q, OMEGA]], inverse[S], CARD] := composite[inverse[S], CARD]
```