inverse elements in a category

Johan G. F. Belinfante
2009 January 29

```
In[1]:= SetDirectory["l:"]; << goedel.09jan28a;<< tools.m

        :Package Title: goedel.09jan28a            2009 January 28 at 9:50 p.m.

        It is now:  2009 Jan 29 at 14:12

        Loading Simplification Rules

        TOOLS.M                                    Revised 2009 January 20

        weightlimit = 40
```

summary

In general, a morphism in a category need not have an inverse, but if it does, it is unique. In this notebook, some results about inverses in categories are derived. These results also apply to monoids and groups, although one can generally obtain sharper results for groups.

invertibility

Various conditions for invertibility can be given. The simplest of these follows directly from the definition of **inv[x]**.

Theorem. A general formula for the class of invertible elements.

```
In[50]:= SubstTest[domain, intersection[t, inverse[t]], t → image[inverse[x], ids[x]]]

Out[50]= fix[composite[image[inverse[x], ids[x]], image[inverse[x], ids[x]]]] == domain[inv[x]]

In[56]:= fix[composite[image[inverse[x_], ids[x_]], image[inverse[x_], ids[x_]]]] :=
          domain[inv[x]]
```

Theorem. Morphisms **u** and **v** in a category **cat[x]** are inverses of each other if both **u · v** and **v · u** are identity morphisms.

```
In[2]:= Map[implies[#, member[pair[u, v], inv[cat[x]]]] &, SubstTest[member, pair[u, v],
          intersection[t, inverse[t]], t -> image[inverse[cat[x]], ids[cat[x]]]]]

Out[2]= or[member[pair[u, v], inv[cat[x]]], not[member[APPLY[cat[x], PAIR[u, v]], ids[cat[x]]]],
          not[member[APPLY[cat[x], PAIR[v, u]], ids[cat[x]]]]] == True

In[3]:= or[member[pair[u_, v_], inv[cat[x_]]],
          not[member[APPLY[cat[x_], PAIR[u_, v_]], ids[cat[x_]]]],
          not[member[APPLY[cat[x_], PAIR[v_, u_]], ids[cat[x_]]]]] := True
```

Lemma. This is needed to cope with a rewrite rule for the associative law.

```
In[5]:= SubstTest[implies, and[member[pair[u, t], domain[cat[x]]], member[t, ids[cat[x]]]],
          equal[u, APPLY[cat[x], PAIR[u, t]]], t → APPLY[cat[x], PAIR[v, w]]] // Reverse

Out[5]= or[equal[u, APPLY[cat[x], PAIR[APPLY[cat[x], PAIR[u, v]], w]]],
          not[member[APPLY[cat[x], PAIR[v, w]], ids[cat[x]]]],
          not[member[pair[u, APPLY[cat[x], PAIR[v, w]]], domain[cat[x]]]]] == True

In[6]:= (% /. {x → x_, u → u_, v → v_, w → w_}) /. Equal → SetDelayed
```

Theorem. If a morphism **u** is a left-inverse of a morphism **v** and if **w** is a right-inverse of **v**, then **u = w**. (Comment. This derivation takes 42 seconds. To speed things up, the conclusion **u = w** has been bundled with the hypotheses in **p1**.)

```
In[9]:= Map[not, SubstTest[and, implies[p2, p3],
          implies[p2, p4], implies[and[p1, p3], p5], implies[and[p1, p4], p6],
          p1, {p1 → and[member[APPLY[cat[x], PAIR[u, v]], ids[cat[x]]],
             member[APPLY[cat[x], PAIR[v, w]], ids[cat[x]]], not[equal[u, w]]],
           p2 → and[member[pair[u, v], domain[cat[x]]], member[pair[v, w], domain[cat[x]]]],
           p3 → member[pair[u, APPLY[cat[x], PAIR[v, w]]], domain[cat[x]]],
           p4 → member[pair[APPLY[cat[x], PAIR[u, v]], w], domain[cat[x]]],
           p5 → equal[u, APPLY[cat[x], PAIR[APPLY[cat[x], PAIR[u, v]], w]]],
           p6 → equal[w, APPLY[cat[x], PAIR[APPLY[cat[x], PAIR[u, v]], w]]]}]] // Reverse

Out[9]= or[equal[u, w], not[member[APPLY[cat[x], PAIR[u, v]], ids[cat[x]]]],
          not[member[APPLY[cat[x], PAIR[v, w]], ids[cat[x]]]]] == True

In[10]:= or[equal[u_, w_], not[member[APPLY[cat[x_], PAIR[u_, v_]], ids[cat[x_]]]],
          not[member[APPLY[cat[x_], PAIR[v_, w_]], ids[cat[x_]]]]] := True
```

Lemma. A morphism **u** is invertible if there exists a morphism **v** such that **pair[u,v] ∈ inv[x]**.

```
In[17]:= SubstTest[implies, and[member[pair[u, v], t], subclass[t, cart[V, V]]],
          member[u, domain[t]], t → inv[x]] // Reverse

Out[17]= or[member[u, domain[inv[x]]], not[member[pair[u, v], inv[x]]]] == True

In[18]:= or[member[u_, domain[inv[x_]]], not[member[pair[u_, v_], inv[x_]]]] := True
```

Corollary. A morphism is invertible if it is left and right invertible.

```
In[22]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3], implies[p3, p4],
           not[implies[p1, p4]], {p1 → and[member[APPLY[cat[x], PAIR[u, v]], ids[cat[x]]],
              member[APPLY[cat[x], PAIR[v, w]], ids[cat[x]]]], p2 → equal[u, w], p3 →
             member[pair[v, w], inv[cat[x]]], p4 → member[v, domain[inv[cat[x]]]]}]] // Reverse

Out[22]= or[member[v, domain[inv[cat[x]]]], not[member[APPLY[cat[x], PAIR[u, v]], ids[cat[x]]]],
          not[member[APPLY[cat[x], PAIR[v, w]], ids[cat[x]]]]] == True

In[23]:= or[member[v_, domain[inv[cat[x_]]]],
          not[member[APPLY[cat[x_], PAIR[u_, v_]], ids[cat[x_]]]],
          not[member[APPLY[cat[x_], PAIR[v_, w_]], ids[cat[x_]]]]] := True
```

Corollary. Obtained by eliminating the variables **u**, **v** and **w**.

```
In[26]:= Map[empty[composite[complement[#], id[cart[V, V]]]] &, SubstTest[class,
           pair[pair[u, v], w], or[member[v, z], not[member[APPLY[funpart[t], PAIR[u, v]], y]],
            not[member[APPLY[funpart[t], PAIR[v, w]], y]]],
           {t → cat[x], y → ids[cat[x]], z -> domain[inv[cat[x]]]}]]

Out[26]= subclass[intersection[domain[image[inverse[cat[x]], ids[cat[x]]]],
            range[image[inverse[cat[x]], ids[cat[x]]]]], domain[inv[cat[x]]]] == True

In[27]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[34]:= SubstTest[implies, subclass[u, v], subclass[domain[u], domain[v]],
           {u → inv[x], v → inverse[image[inverse[x], ids[x]]]}] // Reverse

Out[34]= subclass[domain[inv[x]], range[image[inverse[x], ids[x]]]] == True

In[35]:= subclass[domain[inv[x_]], range[image[inverse[x_], ids[x_]]]] := True
```

Theorem. Invertibility criterion for categories.

```
In[37]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> intersection[domain[image[inverse[cat[x]], ids[cat[x]]]],
             range[image[inverse[cat[x]], ids[cat[x]]]]], v -> domain[inv[cat[x]]]}]

Out[37]= equal[domain[inv[cat[x]]], intersection[domain[image[inverse[cat[x]], ids[cat[x]]]],
            range[image[inverse[cat[x]], ids[cat[x]]]]]] == True

In[39]:= intersection[domain[image[inverse[cat[x_]], ids[cat[x_]]]],
           range[image[inverse[cat[x_]], ids[cat[x_]]]]] := domain[inv[cat[x]]]
```

---

corollary for monoids

Lemma. (Remove the **cat** wrapper.)

```
In[40]:= SubstTest[implies, equal[x, cat[t]],
           equal[domain[inv[x]], intersection[domain[image[inverse[x], ids[x]]],
             range[image[inverse[x], ids[x]]]]], t → x] // Reverse

Out[40]= or[equal[domain[inv[x]], intersection[domain[image[inverse[x], ids[x]]],
             range[image[inverse[x], ids[x]]]]], not[category[x]]] == True

In[41]:= or[equal[domain[inv[x_]], intersection[domain[image[inverse[x_], ids[x_]]],
             range[image[inverse[x_], ids[x_]]]]], not[category[x_]]] := True
```

Theorem. An element in a monoid is invertible if it has left and right inverses.

```
In[44]:= Map[not, SubstTest[and, implies[p1, p2],
           implies[p1, p3], implies[and[p2, p3], p4], not[implies[p1, p4]],
           {p1 → member[x, MONOIDS], p2 → category[x], p3 → equal[ids[x], set[e[x]]],
            p4 -> equal[domain[inv[x]], intersection[domain[image[inverse[x], set[e[x]]]],
               range[image[inverse[x], set[e[x]]]]]]}]] // Reverse

Out[44]= or[equal[domain[inv[x]], intersection[domain[image[inverse[x], set[e[x]]]],
            range[image[inverse[x], set[e[x]]]]]], not[member[x, MONOIDS]]] == True

In[45]:= or[equal[domain[inv[x_]], intersection[domain[image[inverse[x_], set[e[x_]]]],
            range[image[inverse[x_], set[e[x_]]]]]], not[member[x_, MONOIDS]]] := True
```

the commutative case

If **x** is commutative, that is, if **flip[x] = x**, the results obtained above become trivial because left and right invertibility coincide in this case.

```
In[66]:= SubstTest[implies, equal[x, z],
          equal[image[inverse[x], y], image[inverse[z], y]], z → flip[x]] // Reverse

Out[66]= or[equal[image[inverse[x], y], inverse[image[inverse[x], y]]],
          not[equal[x, composite[x, SWAP]]]] == True

In[67]:= or[equal[image[inverse[x_], y_], inverse[image[inverse[x_], y_]]],
          not[equal[x_, composite[x_, SWAP]]]] := True
```

Lemma.

```
In[61]:= SubstTest[implies, and[equal[t, u], equal[t, v]],
          equal[t, intersection[u, v]], {t → image[inverse[x], ids[x]],
           u -> image[inverse[x], ids[x]], v → inverse[image[inverse[x], ids[x]]]}] // Reverse

Out[61]= or[equal[image[inverse[x], ids[x]], inv[x]],
          not[equal[image[inverse[x], ids[x]], inverse[image[inverse[x], ids[x]]]]]] == True

In[62]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. Invertibility in the commutative case.

```
In[64]:= Map[not, SubstTest[and, implies[p1, p2],
           implies[p2, p3], not[implies[p1, p3]], {p1 → equal[x, flip[x]],
            p2 -> equal[image[inverse[x], ids[x]], inverse[image[inverse[x], ids[x]]]],
            p3 -> equal[image[inverse[x], ids[x]], inv[x]]}]] // Reverse

Out[64]= or[equal[image[inverse[x], ids[x]], inv[x]], not[equal[x, composite[x, SWAP]]]] == True

In[65]:= or[equal[image[inverse[x_], ids[x_]], inv[x_]],
          not[equal[x_, composite[x_, SWAP]]]] := True
```