onto morphisms in CATORELN

Johan G. F. Belinfante
2009 August 24

```
In[1]:= SetDirectory["l:"]; << goedel.09aug23a;<< tools.m

        :Package Title: goedel.09aug23a              2009 August 23 at 9:50 p.m.

        It is now:  2009 Aug 24 at 4:46

        Loading Simplification Rules

        TOOLS.M                                    Revised 2009 July 2

        weightlimit = 40
```

summary

A typical morphism of the category **CATORELN** of relations has the form **PAIR[x, y]** where **x** is a relation, **y** is a set and **range[x] ⊂ y**. Such a morphism is said to be **onto** if **range[x] = y**. In this notebook it is shown that the restriction of **CATORELN** to the cartesian square of the class of onto morphisms is a subcategory.

derivation

The first task will be to derive a variable-free formulation of the following fact.

```
In[2]:= implies[equal[domain[x], range[y]], equal[range[composite[x, y]], range[x]]]

Out[2]= True
```

Lemma. (Elimination of the variables.)

```
In[3]:= Map[composite[complement[#], id[cart[V, V]]] &,
         SubstTest[class, pair[pair[x, y], z], implies[and[member[pair[pair[x, y], z], t],
            equal[domain[x], range[y]]], equal[range[z], range[x]]], t → COMPOSE]]

Out[3]= composite[intersection[COMPOSE,
           composite[complement[inverse[IMAGE[SECOND]]], IMAGE[SECOND], FIRST]],
          id[composite[inverse[IMAGE[SECOND]], IMAGE[FIRST]]]] == 0

In[4]:= % /. Equal→ SetDelayed
```

Theorem. A simpler variable-free statement.

```
In[5]:= SubstTest[empty, dif[u, v],
         {u -> composite[COMPOSE, id[composite[inverse[IMAGE[SECOND]], IMAGE[FIRST]]]],
          v -> composite[inverse[IMAGE[SECOND]], IMAGE[SECOND], FIRST]}]

Out[5]= subclass[composite[COMPOSE, id[composite[inverse[IMAGE[SECOND]], IMAGE[FIRST]]]],
          composite[inverse[IMAGE[SECOND]], IMAGE[SECOND], FIRST]] == True

In[6]:= subclass[composite[COMPOSE, id[composite[inverse[IMAGE[SECOND]], IMAGE[FIRST]]]],
          composite[inverse[IMAGE[SECOND]], IMAGE[SECOND], FIRST]] := True
```

Corollary.

```
In[7]:= SubstTest[implies, subclass[u, v], subclass[composite[u, w], composite[v, w]],
          {u -> composite[COMPOSE, id[composite[inverse[IMAGE[SECOND]], IMAGE[FIRST]]]],
           v -> composite[inverse[IMAGE[SECOND]], IMAGE[SECOND], FIRST],
           w → composite[inverse[FIRST], inverse[IMAGE[SECOND]]]}] // Reverse

Out[7]= subclass[
          composite[IMAGE[SECOND], FIRST, id[composite[inverse[IMAGE[SECOND]], IMAGE[FIRST]]],
           inverse[COMPOSE]], IMAGE[SECOND]] == True

In[8]:= subclass[
          composite[IMAGE[SECOND], FIRST, id[composite[inverse[IMAGE[SECOND]], IMAGE[FIRST]]],
           inverse[COMPOSE]], IMAGE[SECOND]] := True
```

Corollary.

```
In[9]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
          {u -> composite[IMAGE[SECOND], FIRST, id[composite[id[P[cart[V, V]]],
               inverse[IMAGE[SECOND]], IMAGE[FIRST], id[P[cart[V, V]]]]], inverse[COMPOSE]],
           v → composite[IMAGE[SECOND], FIRST, id[composite[inverse[IMAGE[SECOND]],
               IMAGE[FIRST]]], inverse[COMPOSE]], w -> IMAGE[SECOND]}] // Reverse

Out[9]= subclass[composite[IMAGE[SECOND], FIRST,
           id[composite[id[P[cart[V, V]]], inverse[IMAGE[SECOND]], IMAGE[FIRST],
             id[P[cart[V, V]]]], inverse[COMPOSE]], IMAGE[SECOND]] == True

In[10]:= % /. Equal→ SetDelayed
```

Lemma.

```
In[11]:= Assoc[direct[COMPOSE, FIRST], id[domain[CATORELN]], id[cartsq[IMAGE[SECOND]]]]

Out[11]= composite[cross[COMPOSE, FIRST], TWIST,
           id[composite[id[composite[IMAGE[SECOND], id[P[cart[V, V]]]]], inverse[SECOND],
             IMAGE[FIRST], FIRST, id[composite[IMAGE[SECOND], id[P[cart[V, V]]]]]]] ==
          composite[CATORELN, id[cart[IMAGE[SECOND], IMAGE[SECOND]]]]

In[12]:= % /. Equal→ SetDelayed
```

Theorem.

```
In[13]:= Assoc[direct[COMPOSE, FIRST], id[domain[CATORELN]],
           id[cartsq[composite[IMAGE[SECOND], id[P[cart[V, V]]]]]]] // Reverse

Out[13]= composite[CATORELN, id[cart[composite[IMAGE[SECOND], id[P[cart[V, V]]]],
            composite[IMAGE[SECOND], id[P[cart[V, V]]]]]] ==
          composite[CATORELN, id[cart[IMAGE[SECOND], IMAGE[SECOND]]]]

In[14]:= composite[CATORELN, id[cart[composite[IMAGE[SECOND], id[P[cart[V, V]]]],
            composite[IMAGE[SECOND], id[P[cart[V, V]]]]]] :=
          composite[CATORELN, id[cart[IMAGE[SECOND], IMAGE[SECOND]]]]
```

Lemma.

```
In[16]:= ImageComp[direct[COMPOSE, FIRST], id[domain[CATORELN]],
          cartsq[composite[IMAGE[SECOND], id[P[cart[V, V]]]]]]

Out[16]= image[CATORELN, cart[composite[IMAGE[SECOND], id[P[cart[V, V]]]],
            composite[IMAGE[SECOND], id[P[cart[V, V]]]]] ==
          composite[IMAGE[SECOND], FIRST, id[composite[id[P[cart[V, V]]],
            inverse[IMAGE[SECOND]], IMAGE[FIRST], id[P[cart[V, V]]]], inverse[COMPOSE]]]

In[17]:= % /. Equal→ SetDelayed
```

Theorem. The restriction of **CATORELN** to the cartesian square of **IMAGE[SECOND]** is a category.

```
In[18]:= SubstTest[implies,
           and[subclass[image[cat[x], cart[y, y]], y], invariant[cod[cat[x]], y],
            invariant[dom[cat[x]], y]], category[composite[cat[x], id[cart[y, y]]]],
           {x→ CATORELN, y-> composite[IMAGE[SECOND], id[P[cart[V, V]]]]}] // Reverse

Out[18]= category[composite[CATORELN, id[cart[IMAGE[SECOND], IMAGE[SECOND]]]]] == True

In[19]:= category[composite[CATORELN, id[cart[IMAGE[SECOND], IMAGE[SECOND]]]]] := True
```

Theorem. The category of onto relations is a subcategory of **CATORELN**.

```
In[20]:= SubstTest[implies,
           and[subclass[y, range[cat[x]]], subclass[image[cat[x], cart[y, y]], y],
            invariant[cod[cat[x]], y], invariant[dom[cat[x]], y]],
           functor[id[y], composite[cat[x], id[cart[y, y]]], cat[x]],
           {x→ CATORELN, y-> composite[IMAGE[SECOND], id[P[cart[V, V]]]]}] // Reverse

Out[20]= functor[id[composite[IMAGE[SECOND], id[P[cart[V, V]]]]],
           composite[CATORELN, id[cart[IMAGE[SECOND], IMAGE[SECOND]]]], CATORELN] == True

In[21]:= functor[id[composite[IMAGE[SECOND], id[P[cart[V, V]]]]],
           composite[CATORELN, id[cart[IMAGE[SECOND], IMAGE[SECOND]]]], CATORELN] := True
```