zero bands

Johan G. F. Belinfante
2009 July 15

In[1]:=
```
SetDirectory["l:"]; << goedel.09jul14a; << tools.m
```

```
:Package Title: goedel.09jul14a                2009 July 14 at 3:55 p.m.

It is now:  2009 Jul 15 at 5:11

Loading Simplification Rules

TOOLS.M                                      Revised 2009 July 2

weightlimit = 40
```

summary

Special examples of bands are considered in this notebook. The direct product of two bands is a band. A **left-zero binary operation** is one in which the identity **x y = y** holds. A **right-zero binary operation** is one in which the identity **x y = x** holds. A left-zero binary operation is one that is a restriction of **FIRST**. Similarly, a right-zero binary operation is one that is a restriction of **SECOND**. Both left-zero and right-zero binary operations are bands. A formula for **image[DORA, BANDS]** is derived.

direct products of bands

The wrapper **band[x]** for a generic band makes it easy to derive the main result in this section.

Theorem. The direct product of **band[x]** and **band[y]** is a band.

In[2]:=
```
((member[t, BANDS] // AssertTest) /. t → direct[semigp[u], semigp[v]]) /.
  {u → band[x], v → band[y]}
```

Out[2]=
```
member[composite[cross[band[x], band[y]], TWIST], BANDS] == True
```

In[3]:=
```
member[composite[cross[band[x_], band[y_]], TWIST], BANDS] := True
```

The **band** wrapper can be removed.

Corollary. The direct product of bands is a band.

```
In[4]:= SubstTest[implies, and[equal[x, band[u]], equal[y, band[v]]],
          member[direct[x, y], BANDS], {u → x, v → y}] // Reverse

Out[4]= or[member[composite[cross[x, y], TWIST], BANDS],
          not[member[x, BANDS]], not[member[y, BANDS]]] == True

In[5]:= or[member[composite[cross[x_, y_], TWIST], BANDS],
          not[member[x_, BANDS]], not[member[y_, BANDS]]] := True
```

A variable-free restatement is possible.

Corollary. The class of bands is binary-closed under direct products.

```
In[6]:= Map[empty[composite[Id, complement[#]]] &,
         dif[cart[BANDS, BANDS], image[inverse[composite[IMAGE[cross[TWIST, Id]], CROSS]],
             BANDS]] // complement // RelnNormality]

Out[6]= subclass[image[IMAGE[cross[TWIST, Id]], image[CROSS, cart[BANDS, BANDS]]], BANDS] == True

In[7]:= subclass[image[IMAGE[cross[TWIST, Id]], image[CROSS, cart[BANDS, BANDS]]], BANDS] := True
```

---

left and right zero binary operations

Observation. The class of functions of the form **FIRST ∘ id[cart[x, x]]** is given by:

```
In[8]:= range[VERTSECT[reify[x, composite[FIRST, id[cart[x, x]]]]]]

Out[8]= image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]]
```

Similarly, the class of functions of the form **SECOND ∘ id[cart[x, x]]** is given by:

```
In[9]:= range[VERTSECT[reify[x, composite[SECOND, id[cart[x, x]]]]]]

Out[9]= image[IMAGE[composite[id[SECOND], inverse[FIRST]]], image[CART, Id]]
```

Theorem. (The class of left-zero binary operations.)

```
In[10]:= SubstTest[subclass, intersection[u, v], v, {u → P[FIRST], v → BINOPS}] // Reverse

Out[10]= subclass[image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]], BINOPS] ==
          True

In[11]:= subclass[
           image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]], BINOPS] := True
```

Theorem. (The class of right-zero binary operations.)

```
In[12]:= SubstTest[subclass, intersection[u, v], v, {u → P[SECOND], v → BINOPS}] // Reverse

Out[12]= subclass[
           image[IMAGE[composite[id[SECOND], inverse[FIRST]]], image[CART, Id]], BINOPS] == True
```

```
In[13]:= subclass[
           image[IMAGE[composite[id[SECOND], inverse[FIRST]]], image[CART, Id]], BINOPS] := True
```

left and right zero binary operations are bands

Theorem. The restriction of **FIRST** to the cartesian square of any set is a band.

```
In[14]:= member[composite[FIRST, id[cart[x, x]]], BANDS] // AssertTest

Out[14]= member[composite[FIRST, id[cart[x, x]]], BANDS] == member[x, V]

In[15]:= member[composite[FIRST, id[cart[x_, x_]]], BANDS] := member[x, V]
```

Theorem. The restriction of **SECOND** to the cartesian square of any set is a band.

```
In[16]:= member[composite[SECOND, id[cart[x, x]]], BANDS] // AssertTest

Out[16]= member[composite[SECOND, id[cart[x, x]]], BANDS] == member[x, V]

In[17]:= member[composite[SECOND, id[cart[x_, x_]]], BANDS] := member[x, V]
```

Technical Lemma.

```
In[18]:= image[inverse[composite[IMAGE[cross[Id, FIRST]], CROSS, DUP, IMAGE[DUP]]], BANDS] //
          Normality

Out[18]= image[inverse[IMAGE[DUP]],
           fix[image[inverse[CROSS], image[inverse[IMAGE[cross[Id, FIRST]]], BANDS]]]] == V

In[19]:= % /. Equal → SetDelayed
```

Theorem. Every left-zero binary operation is a band.

```
In[20]:= Map[subclass[#, BANDS] &,
           ImageComp[composite[IMAGE[cross[Id, FIRST]], CROSS, DUP, IMAGE[DUP]], inverse[
             composite[IMAGE[cross[Id, FIRST]], CROSS, DUP, IMAGE[DUP]]], BANDS]] // Reverse

Out[20]= subclass[image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]], BANDS] ==
          True

In[21]:= subclass[
           image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]], BANDS] := True
```

The opposite of a band is a band. The opposite of a left-zero binary operation is a right-zero binary operation. These observations yield the following corollary.

Corollary. Every right-zero binary operation is a band.

```
In[22]:= SubstTest[implies, subclass[u, v],
           subclass[image[t, u], image[t, v]], {t → IMAGE[cross[SWAP, Id]],
            u -> image[IMAGE[cross[Id, FIRST]], image[IMAGE[DUP], image[CART, Id]]],
            v -> BANDS}] // Reverse

Out[22]= subclass[image[IMAGE[composite[id[SECOND], inverse[FIRST]]], image[CART, Id]], BANDS] ==
          True

In[23]:= subclass[
           image[IMAGE[composite[id[SECOND], inverse[FIRST]]], image[CART, Id]], BANDS] := True
```

Lemma. An inclusion.

```
In[24]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
           {t → id[P[FIRST]], u → BANDS, v → BINOPS}] // Reverse

Out[24]= subclass[intersection[BANDS, P[FIRST]],
           image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]]] == True

In[25]:= % /. Equal → SetDelayed
```

This inclusion can be strengthened to an equation.

Theorem. The class of left-zero binary operations is the intersection of the class of bands and the class of restrictions of **FIRST**.

```
In[26]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u → intersection[BANDS, P[FIRST]], v → intersection[BINOPS, P[FIRST]]}]

Out[26]= equal[image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]],
           intersection[BANDS, P[FIRST]]] == True

In[27]:= intersection[BANDS, P[FIRST]] :=
          image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]]
```

Corollary. The class of right-zero binary operations is the intersection of the class of bands and the class of restrictions of **SECOND**.

```
In[28]:= SubstTest[intersection, image[oopart[t], u], image[oopart[t], v],
           {t → composite[IMAGE[cross[SWAP, Id]], id[P[cart[cart[V, V], V]]]],
            u → BANDS, v → P[FIRST]}] // Reverse

Out[28]= intersection[BANDS, P[SECOND]] ==
          image[IMAGE[composite[id[SECOND], inverse[FIRST]]], image[CART, Id]]

In[29]:= intersection[BANDS, P[SECOND]] :=
          image[IMAGE[composite[id[SECOND], inverse[FIRST]]], image[CART, Id]]
```

image[DORA, BANDS]

In this section a formula for **image[DORA, BANDS]** is derived. This formula says that any set can be the range of a band, and that the domain of a band is the cartesian square of its range.

Theorem. An inclusion.

```
In[30]:= Map[equal[V, #] &,
          dif[BANDS, image[inverse[DORA], composite[inverse[DUP], inverse[CART]]]] //
            complement // Normality]

Out[30]= subclass[image[DORA, BANDS], composite[inverse[DUP], inverse[CART]]] == True

In[31]:= % /. Equal → SetDelayed
```

Lemma.

```
In[33]:= composite[IMAGE[FIRST], id[image[CART, Id]]] // ReifNormality

Out[33]= composite[IMAGE[FIRST], id[image[CART, Id]]] == composite[inverse[DUP], inverse[CART]]

In[34]:= composite[IMAGE[FIRST], id[image[CART, Id]]] := composite[inverse[DUP], inverse[CART]]
```

Lemma.

```
In[35]:= Map[image[#, image[CART, Id]] &,
          composite[DORA, IMAGE[composite[id[FIRST], inverse[FIRST]]]] // ReifNormality]

Out[35]= image[DORA, image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]]] ==
          composite[inverse[DUP], inverse[CART]]

In[36]:= image[DORA, image[IMAGE[composite[id[FIRST], inverse[FIRST]]], image[CART, Id]]] :=
          composite[inverse[DUP], inverse[CART]]
```

Lemma. Inclusion in the opposite direction.

```
In[38]:= SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
            {t → DORA, u -> image[IMAGE[cross[Id, FIRST]], image[IMAGE[DUP], image[CART, Id]]],
             v → BANDS}] // Reverse

Out[38]= subclass[composite[inverse[DUP], inverse[CART]], image[DORA, BANDS]] == True

In[39]:= % /. Equal → SetDelayed
```

Theorem. A formula for **image[DORA, BANDS]**.

```
In[40]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> composite[inverse[DUP], inverse[CART]], v -> image[DORA, BANDS]}]

Out[40]= equal[composite[inverse[DUP], inverse[CART]], image[DORA, BANDS]] == True
```

```
In[41]:= image[DORA, BANDS] := composite[inverse[DUP], inverse[CART]]
```