vertical sections of INTDIV

Johan G. F. Belinfante
2011 November 8

```
In[1]:= SetDirectory["l:"]; << goedel.11nov05a

:Package Title: goedel.11nov05a                2011 November 5 at 1:00 p.m.

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

It is now:  2011 Nov 8 at 16:13

Loading Simplification Rules

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

weightlimit = 40

Loading completed.

It is now:  2011 Nov 8 at 16:26
```

summary

In this notebook it is shown that each vertical section of the divisibility relation **INTDIV** for integers is the range of a subgroup of **INTADD**. Explicit formulas are derived for the neutral element and the inversion function of these subgroups.

derivation

The binary operation **INTADD** of integer addition is (the composition law of) a group. The vertical section **image[-INTDIV, {int[x]}]** of the divisibility relation **INTDIV** at any integer **int[x]** is the set of all integer multiples of **int[x]**. Each such vertical section is the range of a subgroup of **INTADD**.

Lemma.

```
In[2]:= Map[member[image[INTDIV, set[int[x]]], #] &, SubstTest[intersection,
          binclosed[gp[x]], complement[set[0]], fix[IMAGE[inv[gp[x]]]], x → INTADD]]

Out[2]= member[image[INTDIV, set[int[x]]],
          image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]] == True

In[3]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma. Elimination of the **int** wrapper.

```
In[4]:= SubstTest[implies, equal[x, int[t]], member[image[INTDIV, set[x]],
           image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]], t → x] // Reverse

Out[4]= or[member[image[INTDIV, set[x]],
           image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]], not[member[x, Z]]] == True

In[5]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma. Converse result.

```
In[6]:= SubstTest[implies, member[t, image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]],
         not[empty[t]], t -> image[INTDIV, set[x]]] // Reverse

Out[6]= or[member[x, Z], not[member[image[INTDIV, set[x]],
           image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]]]] == True

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

Theorem. The vertical sections of **INTDIV** at integers are ranges of subgroups of **INTADD**.

```
In[8]:= equiv[member[image[INTDIV, set[x]],
         image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]], member[x, Z]]

Out[8]= True

In[9]:= member[image[INTDIV, set[x_]],
         image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]] := member[x, Z]
```

A variable-free restatement can be obtained using **reify** and **case**.

Theorem.

```
In[10]:= Map[
          subclass[image[VERTSECT[INTDIV], Z], image[VERTSECT[INTDIV], domain[reify[x, #]]]] &,
          SubstTest[case, member[image[t, set[x]],
            image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]], t → INTDIV]]

Out[10]= subclass[image[VERTSECT[INTDIV], Z],
           image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]] == True

In[11]:= subclass[image[VERTSECT[INTDIV], Z],
           image[IMAGE[SECOND], intersection[GROUPS, P[INTADD]]]] := True
```

Comment. A converse result also holds, but will not be derived in this notebook.

Theorem. The restriction of integer addition to the set of integer multiples of **int[x]** is a group.

```
In[15]:= SubstTest[implies, member[t, image[IMAGE[SECOND], intersection[GROUPS, P[gp[x]]]]],
           member[composite[gp[x], id[cart[t, t]]], GROUPS],
           {t -> image[INTDIV, set[int[x]]], x → INTADD}] // Reverse

Out[15]= member[composite[INTADD,
            id[cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]], GROUPS] == True
```

```
In[16]:= member[composite[INTADD,
           id[cart[image[INTDIV, set[int[x_]]], image[INTDIV, set[int[x_]]]]]], GROUPS] := True
```

Theorem. The identity element of the subgroup is the same as that of the group.

```
In[19]:= SubstTest[implies, and[member[u, GROUPS], subclass[u, v], member[v, GROUPS]],
          equal[e[u], e[v]], {u -> composite[INTADD, id[cart[image[INTDIV, set[int[x]]],
               image[INTDIV, set[int[x]]]]]], v → INTADD}] // Reverse

Out[19]= equal[e[composite[INTADD, id[
              cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]]], id[omega]] == True

In[21]:= e[composite[INTADD,
           id[cart[image[INTDIV, set[int[x_]]], image[INTDIV, set[int[x_]]]]]]] := id[omega]
```

Theorem. The set of neutral elements for the subgroup is a singleton.

```
In[40]:= SubstTest[ids, gp[t], t -> composite[INTADD,
            id[cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]]] // Reverse

Out[40]= ids[composite[INTADD,
           id[cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]]] == set[id[omega]]

In[41]:= ids[composite[INTADD, id[
            cart[image[INTDIV, set[int[x_]]], image[INTDIV, set[int[x_]]]]]]] := set[id[omega]]
```

Lemma.

```
In[26]:= Map[or[#, subclass[inv[composite[INTADD,
               id[cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]]], INVERSE]] &,
          SubstTest[implies, and[member[u, GROUPS], subclass[u, v], member[v, GROUPS]],
           subclass[inv[u], inv[v]],
           {u -> composite[INTADD, id[cart[image[INTDIV, set[int[x]]],
                image[INTDIV, set[int[x]]]]]], v → INTADD}]] // Reverse

Out[26]= subclass[inv[composite[INTADD,
            id[cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]]], INVERSE] == True

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

Lemma.

```
In[31]:= SubstTest[implies, member[t, GROUPS],
          equal[domain[inv[t]], range[t]], t -> composite[INTADD,
            id[cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]]] // Reverse

Out[31]= equal[domain[inv[composite[INTADD,
             id[cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]]]],
          image[INTDIV, set[int[x]]]] == True

In[33]:= domain[inv[composite[INTADD, id[cart[image[INTDIV, set[int[x_]]],
              image[INTDIV, set[int[x_]]]]]]]] := image[INTDIV, set[int[x]]]
```

Theorem. An explicit formula for the function that takes each member of the subgroup to its additive inverse.

```
In[36]:= SubstTest[implies, and[subclass[u, v], FUNCTION[v]],
           equal[u, composite[v, id[domain[u]]]],
           {u -> inv[composite[INTADD, id[cart[image[INTDIV, set[int[x]]],
                 image[INTDIV, set[int[x]]]]]]], v → INVERSE}] // Reverse

Out[36]= equal[composite[INVERSE, id[image[INTDIV, set[int[x]]]]], inv[composite[INTADD,
             id[cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]]]] == True

In[38]:= inv[composite[INTADD,
             id[cart[image[INTDIV, set[int[x_]]], image[INTDIV, set[int[x_]]]]]]] :=
          composite[INVERSE, id[image[INTDIV, set[int[x]]]]]
```

Theorem. The inversion function is an involution.

```
In[43]:= SubstTest[inverse, inv[gp[t]], t -> composite[INTADD,
             id[cart[image[INTDIV, set[int[x]]], image[INTDIV, set[int[x]]]]]]] // Reverse

Out[43]= composite[id[image[INTDIV, set[int[x]]]], INVERSE] ==
          composite[INVERSE, id[image[INTDIV, set[int[x]]]]]

In[44]:= composite[id[image[INTDIV, set[int[x_]]]], INVERSE] :=
          composite[INVERSE, id[image[INTDIV, set[int[x]]]]]
```