linear differences, part 4. closure properties

Johan G. F. Belinfante
2007 April 12

```
In[1]:= SetDirectory["l:"]; << goedel92.11a; << tools.m

        :Package Title: goedel92.11a        2007 April 11 at 11:30 p.m.

        It is now:  2007 Apr 13 at 7:17

        Loading Simplification Rules

        TOOLS.M                      Revised 2007 March 25

        weightlimit = 40
```

summary

Although the set **omega** of natural numbers is not a ring, sets of numbers of the form **image[DIV,set[t]]** are analogous to principal ideals of a ring, and will be called **PI sets** for short. In this notebook it is shown that the set **ld[x, y]** of linear differences is closed under addition, subtraction and multiplication. These facts and others are derived from the fact that **ld[x,y]** is a PI set.

simplification rules

Simplification rule for PI sets.

```
In[2]:= ImageComp[times[x], DIV, set[y]] // Reverse

Out[2]= image[times[x], image[DIV, set[y]]] == image[DIV, set[natmul[x, y]]]

In[3]:= image[times[x_], image[DIV, set[y_]]] := image[DIV, set[natmul[x, y]]]
```

Simplification rule for sets of linear differences.

```
In[4]:= equal[intersection[omega, ld[x, y]], ld[x, y]]

Out[4]= True

In[5]:= intersection[omega, ld[x_, y_]] := ld[x, y]
```

closure under addition

Lemma.

```
In[6]:= image[inverse[VERTSECT[DIV]], fix[composite[IMAGE[NATADD], CART, DUP]]] // Normality

Out[6]= image[inverse[VERTSECT[DIV]], fix[composite[IMAGE[NATADD], CART, DUP]]] == V

In[7]:= % /. Equal → SetDelayed
```

Theorem. PI sets are closed under addition.

```
In[8]:= Map[subclass[range[VERTSECT[DIV]], #] &, ImageComp[VERTSECT[DIV],
          inverse[VERTSECT[DIV]], fix[composite[IMAGE[NATADD], CART, DUP]]]]

Out[8]= subclass[range[VERTSECT[DIV]], fix[composite[IMAGE[NATADD], CART, DUP]]] == True

In[9]:= subclass[range[VERTSECT[DIV]], fix[composite[IMAGE[NATADD], CART, DUP]]] := True
```

Corollary. The set **ld[x,y]** is closed under addition.

```
In[10]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u → ld[x, y], v → range[VERTSECT[DIV]],
            w → fix[composite[IMAGE[NATADD], CART, DUP]]}] // Reverse

Out[10]= equal[image[NATADD, cart[ld[x, y], ld[x, y]]], ld[x, y]] == True

In[11]:= image[NATADD, cart[ld[x_, y_], ld[x_, y_]]] := ld[x, y]
```

closure under subtraction

Closure under subtraction is derived in a fashion similar to that used to derive closure under addition, except that the function **NATADD** is replaced with the function **rotate[NATADD]**.

Lemma.

```
In[12]:= SubstTest[member, x, fix[composite[IMAGE[t], CART, DUP]], t → rotate[NATADD]] // Reverse

Out[12]= member[x, fix[composite[IMG, id[inverse[IMAGE[inverse[NATADD]]]], inverse[SECOND]]]] ==
          and[equal[x, image[image[inverse[NATADD], x], x]], member[x, V]]

In[13]:= member[x_,
           fix[composite[IMG, id[inverse[IMAGE[inverse[NATADD]]]], inverse[SECOND]]]] :=
          and[equal[x, image[image[inverse[NATADD], x], x]], member[x, V]]
```

Lemma.

```
In[14]:= image[inverse[VERTSECT[DIV]],
           fix[composite[IMAGE[rotate[NATADD]], CART, DUP]]] // Normality

Out[14]= image[inverse[VERTSECT[DIV]],
           fix[composite[IMG, id[inverse[IMAGE[inverse[NATADD]]]], inverse[SECOND]]]] == V

In[15]:= % /. Equal → SetDelayed
```

Theorem. PI sets are closed under subtraction.

```
In[16]:= Map[subclass[range[VERTSECT[DIV]], #] &, ImageComp[VERTSECT[DIV],
           inverse[VERTSECT[DIV]], fix[composite[IMAGE[rotate[NATADD]], CART, DUP]]]]

Out[16]= subclass[range[VERTSECT[DIV]],
           fix[composite[IMG, id[inverse[IMAGE[inverse[NATADD]]]], inverse[SECOND]]]] == True

In[17]:= subclass[range[VERTSECT[DIV]],
           fix[composite[IMG, id[inverse[IMAGE[inverse[NATADD]]]], inverse[SECOND]]]] := True
```

Corollary. The set **ld[x, y]** is closed under subtraction.

```
In[18]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u → ld[x, y], v → range[VERTSECT[DIV]],
            w → fix[composite[IMAGE[rotate[NATADD]], CART, DUP]]}] // Reverse

Out[18]= equal[image[image[inverse[NATADD], ld[x, y]], ld[x, y]], ld[x, y]] == True

In[19]:= image[image[inverse[NATADD], ld[x_, y_]], ld[x_, y_]] := ld[x, y]
```

closure under multiplication

Theorem. PI sets are closed under multiplication by any natural number.

```
In[20]:= SubstTest[subclass, range[VERTSECT[x]], range[IMAGE[x]], x → DIV] // Reverse

Out[20]= subclass[range[VERTSECT[DIV]], fix[IMAGE[DIV]]] == True

In[21]:= subclass[range[VERTSECT[DIV]], fix[IMAGE[DIV]]] := True
```

Corollary. The set **ld[x,y]** is closed under multiplication by any natural number.

```
In[22]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u → ld[x, y], v → range[VERTSECT[DIV]], w → fix[IMAGE[DIV]]}] // Reverse

Out[22]= equal[image[DIV, ld[x, y]], ld[x, y]] == True

In[23]:= image[DIV, ld[x_, y_]] := ld[x, y]
```

technical lemma

Temporary rule needed later:

```
In[24]:= ImageComp[times[x], rotate[NATADD], cart[image[DIV, set[y]], image[DIV, set[z]]]]

Out[24]= range[fix[composite[inverse[FIRST], times[x], id[image[DIV, set[z]]], inverse[S],
           id[image[DIV, set[y]]], inverse[times[x]], NATADD]]] == image[times[x], ld[y, z]]

In[25]:= range[fix[composite[inverse[FIRST], times[x_], id[image[DIV, set[z_]]], inverse[S],
           id[image[DIV, set[y_]]], inverse[times[x_]], NATADD]]] := image[times[x], ld[y, z]]
```

Temporary lemma, with a redundant literal.

```
In[26]:= SubstTest[implies, equal[u, v],
           equal[image[u, w], image[v, w]], {u -> composite[times[x], rotate[NATADD]],
            v → composite[rotate[NATADD], cross[times[x], times[x]]],
            w → cart[image[DIV, set[y]], image[DIV, set[z]]]}] // Reverse

Out[26]= or[equal[0, x], equal[image[times[x], ld[y, z]], ld[natmul[x, y], natmul[x, z]]]] == True

In[27]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Technical lemma.

```
In[28]:= ld[complement[image[V, intersection[omega, set[x]]]],
           complement[image[V, intersection[omega, set[y]]]]] // Normality

Out[28]= ld[complement[image[V, intersection[omega, set[x]]]],
           complement[image[V, intersection[omega, set[y]]]]] == intersection[
           image[V, intersection[omega, set[x]]], image[V, intersection[omega, set[y]]], set[0]]

In[29]:= ld[complement[image[V, intersection[omega, set[x_]]]],
           complement[image[V, intersection[omega, set[y_]]]]] := intersection[
           image[V, intersection[omega, set[x]]], image[V, intersection[omega, set[y]]], set[0]]
```

Theorem.

```
In[30]:= SubstTest[and, or[p, q], implies[p, q],
          {p → equal[0, x], q -> equal[image[times[x], ld[y, z]], ld[natmul[x, y], natmul[x, z]]]}]

Out[30]= equal[image[times[x], ld[y, z]], ld[natmul[x, y], natmul[x, z]]] == True

In[31]:= image[times[x_], ld[y_, z_]] := ld[natmul[x, y], natmul[x, z]]
```

Theorem.

```
In[32]:= SubstTest[implies, subclass[s, t], subclass[image[r, s], image[r, t]],
           {r → rotate[NATADD],
            s → cart[image[DIV, set[natmul[u, x]]], image[DIV, set[natmul[v, y]]]],
            t -> cart[image[DIV, set[x]], image[DIV, set[y]]]}] // Reverse

Out[32]= subclass[ld[natmul[u, x], natmul[v, y]], ld[x, y]] == True

In[33]:= subclass[ld[natmul[u_, x_], natmul[v_, y_]], ld[x_, y_]] := True
```

Theorem.

```
In[34]:= Map[or[not[member[pair[w, x], DIV]], not[member[pair[y, z], DIV]], #] &,
            SubstTest[implies, subclass[u, v], subclass[image[t, u], image[t, v]],
             {t → rotate[NATADD], u → cart[image[DIV, set[x]], image[DIV, set[z]]],
              v -> cart[image[DIV, set[w]], image[DIV, set[y]]]}]] // Reverse // MapNotNot

Out[34]= or[not[member[pair[w, x], DIV]],
          not[member[pair[y, z], DIV]], subclass[ld[x, z], ld[w, y]]] == True

In[35]:= or[not[member[pair[w_, x_], DIV]],
          not[member[pair[y_, z_], DIV]], subclass[ld[x_, z_], ld[w_, y_]]] := True
```

closure under NATMOD

Lemma. PI sets are closed under **NATMOD**.

```
In[36]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]], subclass[u, w],
           {u → range[VERTSECT[DIV]], v → binclosed[rotate[NATADD]], w → binclosed[NATMOD]}] //
          Reverse

Out[36]= subclass[range[VERTSECT[DIV]], binclosed[NATMOD]] == True

In[37]:= subclass[range[VERTSECT[DIV]], binclosed[NATMOD]] := True
```

Theorem.

```
In[38]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u -> image[DIV, set[x]], v -> range[VERTSECT[DIV]], w -> binclosed[NATMOD]}] // Reverse

Out[38]= subclass[image[NATMOD, cart[image[DIV, set[x]], image[DIV, set[x]]]],
          image[DIV, set[x]]] == True

In[39]:= (% /. x → x_) /. Equal → SetDelayed
```

The reverse inclusion also holds.

```
In[40]:= SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
           {u → cart[image[DIV, set[x]], set[0]], v → cartsq[image[DIV, set[x]]], w → NATMOD}] //
          Reverse

Out[40]= subclass[image[DIV, set[x]],
           image[NATMOD, cart[image[DIV, set[x]], image[DIV, set[x]]]]] == True

In[41]:= (% /. x → x_) /. Equal → SetDelayed
```

Combining these results yields an equation.

```
In[42]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> image[NATMOD, cart[image[DIV, set[x]], image[DIV, set[x]]]],
           v -> image[DIV, set[x]]}]

Out[42]= equal[image[DIV, set[x]],
           image[NATMOD, cart[image[DIV, set[x]], image[DIV, set[x]]]]] == True

In[43]:= image[NATMOD, cart[image[DIV, set[x_]], image[DIV, set[x_]]]] := image[DIV, set[x]]
```

Lemma.

```
In[44]:= image[inverse[VERTSECT[DIV]], fix[composite[IMAGE[NATMOD], CART, DUP]]] // Normality

Out[44]= image[inverse[VERTSECT[DIV]], fix[composite[IMAGE[NATMOD], CART, DUP]]] == V

In[45]:= % /. Equal → SetDelayed
```

Theorem. A sharper statement of the fact that PI sets are closed under modulo reduction.

```
In[46]:= Map[subclass[range[VERTSECT[DIV]], #] &, ImageComp[VERTSECT[DIV],
           inverse[VERTSECT[DIV]], fix[composite[IMAGE[NATMOD], CART, DUP]]]]

Out[46]= subclass[range[VERTSECT[DIV]], fix[composite[IMAGE[NATMOD], CART, DUP]]] == True

In[47]:= subclass[range[VERTSECT[DIV]], fix[composite[IMAGE[NATMOD], CART, DUP]]] := True
```

Corollary. The set **ld[x,y]** is closed under **NATMOD**.

```
In[48]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u → ld[x, y], v → range[VERTSECT[DIV]],
            w → fix[composite[IMAGE[NATMOD], CART, DUP]]}] // Reverse

Out[48]= equal[image[NATMOD, cart[ld[x, y], ld[x, y]]], ld[x, y]] == True

In[49]:= image[NATMOD, cart[ld[x_, y_], ld[x_, y_]]] := ld[x, y]
```