MIXDIV

Johan G. F. Belinfante
2006 December 3

```
In[1]:= SetDirectory["l:"]; << goedel87.30b; << tools.m

        :Package Title: goedel87.30b          2006 November 30 at 2:05 p.m.

        It is now:  2006 Dec 3 at 4:1

        Loading Simplification Rules

        TOOLS.M                      Revised 2006 November 22

        weightlimit = 40
```

summary

This notebook is concerned with the mixed divisibility relation **MIXDIV** relating natural numbers and integers. A **class**-wrapped membership rule has been introduced to define this relation:

```
In[2]:= Begin["Goedel`Private`"];

In[3]:= FirstMatch[class[t_, member[w_, HoldPattern[MIXDIV]]]]

Out[3]= class[x_, member[y_, MIXDIV]] := ReleaseHold[Module[{z = Unique[]}, class[
          x, exists[z, and[member[z, binhom[NATADD, INTADD]]], member[y, z]]]]]]
```

Various general results from the theory of binary homomorphisms are specialized to the case of **MIXDIV** and in some cases are made more precise.

normalization

Theorem.

```
In[4]:= MIXDIV // Normality // Reverse

Out[4]= U[binhom[NATADD, INTADD]] == MIXDIV

In[5]:= U[binhom[NATADD, INTADD]] := MIXDIV
```

an upper bound

Theorem.

```
In[6]:= SubstTest[subclass, binhom[x, y],
          map[fix[domain[x]], fix[domain[y]]], {x → NATADD, y → INTADD}] // Reverse

Out[6]= subclass[binhom[NATADD, INTADD], map[omega, Z]] == True

In[7]:= subclass[binhom[NATADD, INTADD], map[omega, Z]] := True
```

An upper bound for this mixed divisibility relation is:

```
In[8]:= SubstTest[implies, subclass[x, y], subclass[U[x], U[y]],
          {x -> binhom[NATADD, INTADD], y -> map[omega, Z]}] // Reverse

Out[8]= subclass[MIXDIV, cart[omega, Z]] == True

In[9]:= subclass[MIXDIV, cart[omega, Z]] := True
```

Corollary.

```
In[10]:= SubstTest[subclass, U[binhom[x, y]], cart[V, V], {x → NATADD, y → INTADD}] // Reverse

Out[10]= subclass[MIXDIV, cart[V, V]] == True

In[11]:= subclass[MIXDIV, cart[V, V]] := True
```

Corollary.

```
In[12]:= SubstTest[composite, Id, U[binhom[x, y]], {x → NATADD, y → INTADD}] // Reverse

Out[12]= composite[Id, MIXDIV] == MIXDIV

In[13]:= composite[Id, MIXDIV] := MIXDIV
```

domain

Lemma.

```
In[14]:= Map[not, SubstTest[implies, member[u, v], not[empty[v]],
            {u → cart[omega, set[id[omega]]], v → binhom[NATADD, INTADD]}]] // Reverse

Out[14]= equal[0, binhom[NATADD, INTADD]] == False

In[15]:= equal[0, binhom[NATADD, INTADD]] := False
```

Theorem.

```
In[16]:= SubstTest[or, empty[binhom[x, y]],
           equal[domain[U[binhom[x, y]]], fix[domain[x]]], {x → NATADD, y → INTADD}] // Reverse

Out[16]= equal[omega, domain[MIXDIV]] == True

In[17]:= domain[MIXDIV] := omega
```

range

From the general theory of binary homomorphisms, one only has this inclusion:

```
In[18]:= SubstTest[subclass, range[U[binhom[x, y]]],
           fix[domain[y]], {x → NATADD, y → INTADD}] // Reverse

Out[18]= subclass[range[MIXDIV], Z] == True

In[19]:= % /. Equal → SetDelayed
```

Lemma.

```
In[20]:= SubstTest[implies, member[u, v], subclass[u, U[v]],
           {u → PLUS, v → binhom[NATADD, INTADD]}] // Reverse

Out[20]= subclass[PLUS, MIXDIV] == True

In[21]:= subclass[PLUS, MIXDIV] := True
```

Theorem. A particular "negative" binary hom.

```
In[22]:= SubstTest[implies, and[member[u, binhom[y, z]], member[v, binhom[x, y]]],
           member[composite[u, v], binhom[x, z]],
           {u → composite[id[Z], INVERSE], v → PLUS, x → NATADD, y → INTADD, z → INTADD}] // Reverse

Out[22]= member[composite[INVERSE, PLUS], binhom[NATADD, INTADD]] == True

In[23]:= member[composite[INVERSE, PLUS], binhom[NATADD, INTADD]] := True
```

Corollary.

```
In[24]:= SubstTest[implies, member[x, y], subclass[x, U[y]],
           {x → composite[INVERSE, PLUS], y → binhom[NATADD, INTADD]}] // Reverse

Out[24]= subclass[composite[INVERSE, PLUS], MIXDIV] == True

In[25]:= subclass[composite[INVERSE, PLUS], MIXDIV] := True

In[26]:= SubstTest[implies, subclass[u, v], subclass[range[u], range[v]],
           {u -> composite[INVERSE, PLUS], v -> MIXDIV}] // Reverse

Out[26]= subclass[image[INVERSE, range[PLUS]], range[MIXDIV]] == True

In[27]:= % /. Equal → SetDelayed
```

Corollary.

```
In[28]:= SubstTest[subclass, union[u, v], w,
           {u → range[PLUS], v -> image[INVERSE, range[PLUS]], w → range[MIXDIV]}] // Reverse

Out[28]= subclass[Z, range[MIXDIV]] == True

In[29]:= % /. Equal → SetDelayed
```

Theorem.

```
In[30]:= SubstTest[and, subclass[u, v], subclass[v, u], {u -> Z, v -> range[MIXDIV]}]

Out[30]= equal[Z, range[MIXDIV]] == True

In[31]:= range[MIXDIV] := Z
```

divisibility by 0

Lemma: each binary hom can only take the idempotent **0** of **NATADD** to the idempotent **+0 = id[omega]** of **INTADD**.

```
In[32]:= SubstTest[subclass,
           image[U[binhom[x, y]], fix[composite[x, DUP]]], fix[composite[y, DUP]],
           {x → NATADD, y → INTADD}] // Reverse

Out[32]= subclass[image[MIXDIV, set[0]], set[id[omega]]] == True

In[33]:= subclass[image[MIXDIV, set[0]], set[id[omega]]] := True
```

Lemma. An inclusion provided by the constant homomorphism that takes all natural numbers to the integer plus zero.

```
In[34]:= SubstTest[implies, member[u, v], subclass[u, U[v]],
           {u -> cart[omega, set[id[omega]]], v -> binhom[NATADD, INTADD]}] // Reverse

Out[34]= subclass[omega, image[inverse[MIXDIV], set[id[omega]]]] == True

In[35]:= % /. Equal → SetDelayed
```

This can be made into an equation:

```
In[36]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> omega, v -> image[inverse[MIXDIV], set[id[omega]]]}]

Out[36]= equal[omega, image[inverse[MIXDIV], set[id[omega]]]] == True

In[37]:= image[inverse[MIXDIV], set[id[omega]]] := omega
```

Lemma.

```
In[38]:= SubstTest[implies, and[member[x, y], subclass[y, z]], member[x, z],
           {x -> pair[0, id[omega]], y → cart[omega, set[id[omega]]], z → MIXDIV}] // Reverse

Out[38]= member[pair[0, id[omega]], MIXDIV] == True

In[39]:= % /. Equal → SetDelayed
```

Corollary. The only integer divisible by the natural number **0** is the integer **+0 = id[omega]**.

```
In[40]:= equal[image[MIXDIV, set[0]], set[id[omega]]]

Out[40]= True

In[41]:= image[MIXDIV, set[0]] := set[id[omega]]
```

relation of MIXDIV to DIV

Theorem.

```
In[42]:= SubstTest[subclass, composite[U[binhom[y, z]], U[binhom[x, y]]],
           U[binhom[x, z]], {x → NATADD, y → NATADD, z → INTADD}] // Reverse

Out[42]= subclass[composite[MIXDIV, DIV], MIXDIV] == True

In[43]:= % /. Equal → SetDelayed

In[44]:= SubstTest[implies, subclass[u, v], subclass[composite[t, u], composite[t, v]],
           {t → MIXDIV, u → id[omega], v → DIV}] // Reverse

Out[44]= subclass[MIXDIV, composite[MIXDIV, DIV]] == True

In[45]:= % /. Equal → SetDelayed

In[46]:= SubstTest[and, subclass[u, v], subclass[v, u], {u -> MIXDIV, v -> composite[MIXDIV, DIV]}]

Out[46]= equal[MIXDIV, composite[MIXDIV, DIV]] == True

In[47]:= composite[MIXDIV, DIV] := MIXDIV
```

relation of MIXDIV to INTDIV

```
In[48]:= SubstTest[subclass, composite[U[binhom[y, z]], U[binhom[x, y]]],
           U[binhom[x, z]], {x → NATADD, y → INTADD, z → INTADD}] // Reverse

Out[48]= subclass[composite[INTDIV, MIXDIV], MIXDIV] == True

In[49]:= % /. Equal → SetDelayed
```

```
In[50]:= SubstTest[implies, subclass[u, v], subclass[composite[u, w], composite[v, w]],
           {u → id[Z], v → INTDIV, w → MIXDIV}] // Reverse

Out[50]= subclass[MIXDIV, composite[INTDIV, MIXDIV]] == True

In[51]:= % /. Equal → SetDelayed

In[52]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> MIXDIV, v -> composite[INTDIV, MIXDIV]}]

Out[52]= equal[MIXDIV, composite[INTDIV, MIXDIV]] == True

In[53]:= composite[INTDIV, MIXDIV] := MIXDIV
```

every integer is divisible by 1 = set[0]

Lemma.

```
In[54]:= SubstTest[implies, subclass[u, v], subclass[composite[u, w], composite[v, w]],
           {u → PLUS, v -> MIXDIV, w → DIV}] // Reverse

Out[54]= subclass[composite[PLUS, DIV], MIXDIV] == True

In[55]:= subclass[composite[PLUS, DIV], MIXDIV] := True
```

Theorem.

```
In[56]:= SubstTest[implies, subclass[u, v], subclass[image[u, w], image[v, w]],
           {u -> composite[PLUS, DIV], v -> MIXDIV, w → set[set[0]]}] // Reverse

Out[56]= subclass[range[PLUS], image[MIXDIV, set[set[0]]]] == True

In[57]:= (% /. Equal → SetDelayed)
```

Lemma.

```
In[58]:= SubstTest[implies, subclass[u, v], subclass[composite[u, w], composite[v, w]],
           {u -> composite[INVERSE, PLUS], v -> MIXDIV, w → DIV}] // Reverse

Out[58]= subclass[composite[INVERSE, PLUS, DIV], MIXDIV] == True

In[59]:= subclass[composite[INVERSE, PLUS, DIV], MIXDIV] := True
```

Theorem.

```
In[60]:= SubstTest[implies, subclass[u, v], subclass[image[u, w], image[v, w]],
           {u -> composite[INVERSE, PLUS, DIV], v -> MIXDIV, w → set[set[0]]}] // Reverse

Out[60]= subclass[image[INVERSE, range[PLUS]], image[MIXDIV, set[set[0]]]] == True

In[61]:= (% /. Equal → SetDelayed)
```

Combining the two theorems, one obtains:

```
In[62]:= SubstTest[subclass, union[u, v], w, {u → range[PLUS],
             v -> image[INVERSE, range[PLUS]], w -> image[MIXDIV, set[set[0]]]}] // Reverse

Out[62]= subclass[Z, image[MIXDIV, set[set[0]]]] == True

In[63]:= (% /. Equal → SetDelayed)
```

Theorem.

```
In[64]:= SubstTest[and, subclass[u, v], subclass[v, u], {u -> Z, v -> image[MIXDIV, set[set[0]]]}]

Out[64]= equal[Z, image[MIXDIV, set[set[0]]]] == True

In[65]:= image[MIXDIV, set[set[0]]] := Z
```

In general, one may regard **image[MIXDIV, set[nat[x]]]** as the set of integers divisible by the natural number **nat[x]**. What has been shown here is that every integer is divisible by the natural number **1 = set[0]**.

composites of MIXDIV with its inverse

Lemma.

```
In[72]:= SubstTest[implies, and[subclass[u, v], subclass[x, y]],
           subclass[composite[u, x], composite[v, y]],
           {u -> cart[set[id[omega]], omega], v -> inverse[MIXDIV],
            x → cart[omega, set[id[omega]]], y -> MIXDIV}] // Reverse

Out[72]= subclass[cart[omega, omega], composite[inverse[MIXDIV], MIXDIV]] == True

In[73]:= % /. Equal → SetDelayed
```

Theorem.

```
In[76]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> cart[omega, omega], v -> composite[inverse[MIXDIV], MIXDIV]}]

Out[76]= equal[cart[omega, omega], composite[inverse[MIXDIV], MIXDIV]] == True

In[78]:= composite[inverse[MIXDIV], MIXDIV] := cart[omega, omega]
```

Lemma.

```
In[82]:= SubstTest[implies, and[subclass[u, v], subclass[x, y]],
           subclass[composite[u, x], composite[v, y]],
           {u -> cart[set[set[0]], Z], v -> MIXDIV, x → cart[Z, set[set[0]]],
            y → inverse[MIXDIV]}] // Reverse

Out[82]= subclass[cart[Z, Z], composite[MIXDIV, inverse[MIXDIV]]] == True
```

```
In[83]:= % /. Equal → SetDelayed
```

Theorem.

```
In[84]:= SubstTest[and, subclass[u, v], subclass[v, u],
           {u -> cart[Z, Z], v -> composite[MIXDIV, inverse[MIXDIV]]}]

Out[84]= equal[cart[Z, Z], composite[MIXDIV, inverse[MIXDIV]]] == True

In[86]:= composite[MIXDIV, inverse[MIXDIV]] := cart[Z, Z]
```