differences of multiples are multiples

Johan G. F. Belinfante
2005 March 29

```
In[1]:= SetDirectory["i:"]; << goedel67.26b; << tools.m

:Package Title: goedel67.26b          2005 March 26 at 6:50 p.m.

It is now:  2005 Mar 29 at 9:50

Loading Simplification Rules

TOOLS.M                        Revised 2005 February 22

weightlimit = 40
```

summary

It is shown that the set of multiples of **x** is closed under subtraction. A corollary is that if two consecutive numbers are both divisible by **x**, then **x = 1**.

a distributive law

The following temporary abbreviation is convenient.

```
In[2]:= multiply[x_] := composite[NATMUL, LEFT[x]]
```

The following almost says that the difference of multiples is a multiple, but the occurrence of the subset relation **S** mars the result. Its appearance is a consequence of the fact that subtraction is not defined for an arbitrary pair of natural numbers. One can only subtract a smaller number from a larger one, and not the other way around.

```
In[3]:= ImageComp[multiply[x], rotate[NATADD], V] // InvertFix

Out[3]= range[fix[composite[inverse[NATADD], NATMUL, LEFT[x], S,
          inverse[LEFT[x]], inverse[NATMUL], FIRST]]] == image[DIV, set[x]]
```

This is made into a temporary rewrite rule.

```
In[4]:= range[fix[composite[inverse[NATADD], NATMUL, LEFT[x_], S,
          inverse[LEFT[x_]], inverse[NATMUL], FIRST]]] := image[DIV, set[x]]
```

a monotonicity property

If the product **natmul[x,y]** is less than or equal to **natmul[x,z]**, then **y** is less than or equal to **z**, or **x** is zero, assuming these are all natural numbers. A formulation of this observation can be derived in which the variables **y** and **z** have been eliminated:

```
In[5]:= composite[inverse[multiply[x]], S, multiply[x]] // VSRenormality

Out[5]= composite[inverse[LEFT[x]], inverse[NATMUL], S, NATMUL, LEFT[x]] ==
         union[cart[omega, intersection[omega, complement[image[V, x]]]],
          composite[id[intersection[omega, image[V, intersection[omega, set[x]]]]],
           S, id[omega]]]

In[6]:= composite[inverse[LEFT[x_]], inverse[NATMUL], S, NATMUL, LEFT[x_]] :=
         union[cart[omega, intersection[omega, complement[image[V, x]]]],
          composite[id[intersection[omega, image[V, intersection[omega, set[x]]]]],
           S, id[omega]]]
```

differences of multiples

The following observation can be made into a rewrite rule:

```
In[7]:= equal[union[image[DIV, set[x]],
          intersection[complement[image[V, x]], image[image[inverse[NATADD],
             image[DIV, set[x]]], image[DIV, set[x]]]]], image[DIV, set[x]]]

Out[7]= True

In[8]:= union[image[DIV, set[x_]], intersection[complement[image[V, x_]],
           image[image[inverse[NATADD], image[DIV, set[x_]]],
            image[DIV, set[x_]]]]] := image[DIV, set[x]]
```

The results of the preceding two sections can be combined:

```
In[9]:= Map[range[fix[composite[inverse[NATADD], #, FIRST]]] &,
         ImageComp[cross[multiply[x], multiply[x]],
          cross[inverse[multiply[x]], inverse[multiply[x]]], S]]

Out[9]= image[image[inverse[NATADD], image[DIV, set[x]]], image[DIV, set[x]]] ==
         image[DIV, set[x]]

In[10]:= image[image[inverse[NATADD], image[DIV, set[x_]]], image[DIV, set[x_]]] :=
          image[DIV, set[x]]
```

This says that the set of multiples of **x** is closed under subtraction.

DIV membership rules

```
In[11]:= SubstTest[member, z, composite[v, id[w]],
           {v → DIV, w → omega, z → pair[x, y]}] // Reverse

Out[11]= and[member[x, omega], member[pair[x, y], DIV]] == member[pair[x, y], DIV]

In[12]:= and[member[x_, omega], member[pair[x_, y_], DIV]] := member[pair[x, y], DIV]

In[13]:= SubstTest[member, z, composite[id[w], v],
           {v → DIV, w → omega, z → pair[x, y]}] // Reverse

Out[13]= and[member[y, omega], member[pair[x, y], DIV]] == member[pair[x, y], DIV]

In[14]:= and[member[y_, omega], member[pair[x_, y_], DIV_]] := member[pair[x, y], DIV]
```

consecutive multiples

Corollary. If two conecutive numbers are both divisible by **x**, then **x = 1**.

```
In[15]:= SubstTest[implies, subclass[u, v], subclass[image[w, u], image[w, v]],
           {u → set[PAIR[succ[y], y]], v → cart[image[DIV, set[x]], image[DIV, set[x]]],
            w → rotate[NATADD]}] // MapNotNot

Out[15]= or[equal[x, set[0]], not[member[pair[x, y], DIV]],
           not[member[pair[x, succ[y]], DIV]]] == True

In[16]:= or[equal[x_, set[0]], not[member[pair[x_, y_], DIV]],
           not[member[pair[x_, succ[y_]], DIV]]] := True
```

A variable-free formulation of this result can be derived.

```
In[17]:= Map[equal[0, composite[Id, complement[#]]] &, SubstTest[class,
              pair[x, y], or[equal[x, set[0]], not[member[pair[x, y], v]],
               not[member[pair[x, succ[y]], v]]], v → DIV]] // Reverse // InvertFix

Out[17]= subclass[fix[composite[inverse[DIV], SUCC, DIV]], set[set[0]]] == True

In[18]:= % /. Equal → SetDelayed

In[19]:= equal[fix[composite[inverse[DIV], SUCC, DIV]], set[set[0]]] // AssertTest

Out[19]= equal[fix[composite[inverse[DIV], SUCC, DIV]], set[set[0]]] == True
```

```
In[20]:= Equal[fix[composite[inverse[DIV], SUCC, DIV]], set[set[0]]]

Out[20]= fix[composite[inverse[DIV], SUCC, DIV]] == set[set[0]]

In[21]:= fix[composite[inverse[DIV], SUCC, DIV]] := set[set[0]]
```