division by zero for natural numbers

Johan G. F. Belinfante
2007 January 30

```
In[1]:= SetDirectory["l:"]; << goedel89.27a; << tools.m

        :Package Title: goedel89.27a        2007 January 27 at 4:10 p.m.

        It is now:  2007 Jan 30 at 22:32

        Loading Simplification Rules

        TOOLS.M                        Revised 2007 January 7

        weightlimit = 40
```

summary

The function **times[x]** is one-to-one unless **x = 0**. Nonetheless, the case **x = 0** need not always be automatically excluded in formulating theorems involving division of natural numbers. By not excluding zero divisors, one can sometimes derive converses of theorems, leading to better rewrite rules. In this notebook, for example, a rewrite rule is derived that says that **y** is the product of **x** and **APPLY[inverse[times[x]],y]** if and only if either **x** is a divisor of **y** or **y = V**.

derivation

Lemma.

```
In[2]:= SubstTest[image, funpart[w], set[PAIR[y, x]], w → rotate[NATMUL]] // Reverse

Out[2]= intersection[image[V, x], image[inverse[times[x]], set[y]]] ==
         intersection[image[V, x], set[APPLY[inverse[times[x]], y]]]

In[3]:= intersection[image[V, x_], image[inverse[times[x_]], set[y_]]] :=
         intersection[image[V, x], set[APPLY[inverse[times[x]], y]]]
```

Lemma.

```
In[4]:= equal[intersection[complement[image[V, x]], image[inverse[times[x]], set[y]]],
         intersection[omega, complement[image[V, x]], complement[image[V, y]]]]

Out[4]= True

In[5]:= intersection[complement[image[V, x_]], image[inverse[times[x_]], set[y_]]] :=
         intersection[omega, complement[image[V, x]], complement[image[V, y]]]
```

Lemma.

```
In[6]:= Map[subclass[#, omega] &,
         SubstTest[union, intersection[u, v], intersection[u, complement[v]],
          {u -> image[inverse[times[x]], set[y]], v → image[V, x]}]] // Reverse

Out[6]= or[equal[0, x], member[APPLY[inverse[times[x]], y], omega],
         not[member[pair[x, y], DIV]]] == True

In[7]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

The hypothesis that **x** be nonzero is in fact not needed. The following stronger result is true:

```
In[8]:= SubstTest[and, implies[p, q], or[p, q], {p -> equal[0, x],
         q -> or[member[APPLY[inverse[times[x]], y], omega], not[member[pair[x, y], DIV]]]}]

Out[8]= or[member[APPLY[inverse[times[x]], y], omega], not[member[pair[x, y], DIV]]] == True

In[9]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

The converse also holds:

```
In[10]:= SubstTest[implies, member[z, omega],
          member[z, V], z -> APPLY[inverse[times[x]], y]] // Reverse

Out[10]= or[member[pair[x, y], DIV], not[member[APPLY[inverse[times[x]], y], omega]]] == True

In[11]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem.

```
In[12]:= equiv[member[APPLY[inverse[times[x]], y], omega], member[pair[x, y], DIV]]

Out[12]= True

In[13]:= member[APPLY[inverse[times[x_]], y_], omega] := member[pair[x, y], DIV]
```

Comment. When **x = 0**, the condition **member[pair[x,y], DIV]** implies that **y = 0** as well, and in this case one has:

```
In[14]:= APPLY[inverse[times[x]], y] /. {x → 0, y → 0}

Out[14]= 0
```

Lemma.

```
In[15]:= Map[implies[equal[y, #], equal[y, natmul[APPLY[inverse[times[x]], y], x]]] &,
           ApComp[times[x], inverse[times[x]], y]] // Reverse // MapNotNot

Out[15]= or[equal[0, x], equal[y, natmul[x, APPLY[inverse[times[x]], y]]],
          not[member[pair[x, y], DIV]]] == True

In[16]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Here again one need not exclude the case **x = 0**. In that case the equation **y = natmul[x, APPLY[inverse[times[x]], y]]** reduces to **0 = 0 · 0**.

```
In[17]:= SubstTest[and, implies[p, q], or[p, q], {p -> equal[0, x], q ->
            or[equal[y, natmul[x, APPLY[inverse[times[x]], y]]], not[member[pair[x, y], DIV]]]}]

Out[17]= or[equal[y, natmul[x, APPLY[inverse[times[x]], y]]],
          not[member[pair[x, y], DIV]]] == True

In[18]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Note that the equation **y = natmul[x, APPLY[inverse[times[x]], y]]** also holds when **y = V**. This possibility must be taken into account in formulating the converse. Combining the above result with its converse yields the following theorem:

```
In[19]:= equiv[equal[y, natmul[x, APPLY[inverse[times[x]], y]]],
            or[member[pair[x, y], DIV], equal[V, y]]] // not // not

Out[19]= True

In[20]:= equal[y_, natmul[x_, APPLY[inverse[times[x_]], y_]]] :=
          or[equal[V, y], member[pair[x, y], DIV]]
```

---

vertical section rule

The formulas for vertical sections of **inverse[times[x]]** derived above can be combined and simplified as follows:

```
In[21]:= Map[equal[#, union[set[APPLY[inverse[times[x]], y]],
              intersection[omega, complement[image[V, x]], complement[image[V, y]]]]] &,
          SubstTest[union, intersection[u, v], intersection[u, complement[v]],
           {u -> image[inverse[times[x]], set[y]], v → image[V, x]}]]

Out[21]= equal[image[inverse[times[x]], set[y]],
          union[intersection[omega, complement[image[V, x]], complement[image[V, y]]],
           set[APPLY[inverse[times[x]], y]]]] == True

In[22]:= image[inverse[times[x_]], set[y_]] :=
          union[intersection[omega, complement[image[V, x]], complement[image[V, y]]],
           set[APPLY[inverse[times[x]], y]]]
```