ratio of negatives of rationals

Johan G. F. Belinfante
2012 October 18

```
In[1]:= SetDirectory["l:"]; << goedel.12oct17a

        :Package Title: goedel.12oct17a              2012 October 17 at 3:30 a.m.

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

        It is now:  2012 Oct 18 at 10:40

        Loading Simplification Rules

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

        weightlimit = 40

        Loading completed.

        It is now:  2012 Oct 18 at 10:56
```

summary

Fractions are unchanged if the signs of both numerator and denominator are changed: **(– d) \ (– n) = d \ n**.

derivation

The following observation is key.

```
In[9]:= implies[member[t, Z],
          or[equal[t, id[omega]], equal[composite[inverse[inttimes[u]], inttimes[v]],
            composite[inverse[inttimes[intmul[t, u]]], inttimes[intmul[t, v]]]]]]

Out[9]= True
```

Lemma. The integer minus one is not equal to the integer zero.

```
In[10]:= equal[composite[inverse[SUCC], id[omega]], id[omega]] // AssertTest

Out[10]= equal[composite[inverse[SUCC], id[omega]], id[omega]] == False

In[11]:= equal[composite[inverse[SUCC], id[omega]], id[omega]] := False
```

Theorem. Simplification rule: **(– int[x]) \ (– int[y]) = int[x] \ int[y]**.

```
In[13]:= SubstTest[implies, member[t, Z],
           or[equal[t, id[omega]], equal[composite[inverse[inttimes[u]], inttimes[v]],
             composite[inverse[inttimes[intmul[u, t]]], inttimes[intmul[v, t]]]]],
           {t → inverse[plus[set[0]]], u → int[x], v → int[y]}] // Reverse

Out[13]= equal[composite[inverse[inttimes[int[x]]], inttimes[int[y]]],
           composite[inverse[inttimes[inverse[int[x]]]], inttimes[inverse[int[y]]]]] == True

In[15]:= composite[inverse[inttimes[inverse[int[x_]]]], inttimes[inverse[int[y_]]]] :=
          composite[inverse[inttimes[int[x]]], inttimes[int[y]]]
```

Theorem. If **x** and **y** are integers, then **(– x) \ (– y) = x \ y**.

```
In[16]:= SubstTest[implies, and[equal[x, int[u]], equal[y, int[v]]],
           equal[frac[x, y], frac[inverse[x], inverse[y]]], {u → x, v → y}] // Reverse

Out[16]= or[equal[composite[inverse[inttimes[x]], inttimes[y]],
            composite[inverse[inttimes[inverse[x]]], inttimes[inverse[y]]]],
           not[member[x, Z]], not[member[y, Z]]] == True

In[18]:= or[equal[composite[inverse[inttimes[inverse[x_]]], inttimes[inverse[y_]]],
            composite[inverse[inttimes[x_]], inttimes[y_]]],
           not[member[x_, Z]], not[member[y_, Z]]] := True
```

---

a variable-free formulation

Lemma.

```
In[21]:= SubstTest[equal, inverse[u], inverse[v], {u → id[omega], v → int[x]}] // Reverse

Out[21]= equal[id[omega], inverse[int[x]]] == equal[id[omega], int[x]]

In[22]:= equal[id[omega], inverse[int[x_]]] := equal[id[omega], int[x]]
```

Theorem.

```
In[23]:= SubstTest[implies, equal[x, int[t]],
           or[equal[x, id[omega]], not[equal[id[omega], inverse[x]]]], t → x] // Reverse

Out[23]= or[equal[x, id[omega]], not[equal[id[omega], inverse[x]]], not[member[x, Z]]] == True

In[24]:= or[equal[x_, id[omega]], not[equal[id[omega], inverse[x_]]], not[member[x_, Z]]] := True
```

Theorem.

```
In[27]:= (Map[implies[member[x, domain[RATIO]], #] &,
           (member[t, fix[composite[inverse[funpart[u]], funpart[v]]]] // AssertTest) /.
            {t → PAIR[first[x], second[x]], u → RATIO,
             v → composite[RATIO, cross[INVERSE, INVERSE]]}]) // MapNotNot

Out[27]= or[equal[first[x], id[omega]],
           member[x, fix[composite[inverse[RATIO], RATIO, cross[INVERSE, INVERSE]]]],
           not[member[first[x], Z]], not[member[second[x], Z]]] == True

In[28]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. (Eliminate the variable **x**.)

```
In[29]:= Map[equal[V, domain[#]] &,
          SubstTest[reify, x, case[implies[member[x, u], member[x, v]]], {u → domain[RATIO],
            v -> fix[composite[inverse[RATIO], RATIO, cross[INVERSE, INVERSE]]]}]]

Out[29]= subclass[cart[intersection[Z, complement[set[id[omega]]]], Z],
           fix[composite[inverse[RATIO], RATIO, cross[INVERSE, INVERSE]]]] == True

In[30]:= % /. Equal → SetDelayed
```

Corollary. An inclusion.

```
In[31]:= SubstTest[subclass, domain[funpart[u]],
          fix[composite[inverse[funpart[u]], funpart[v]]],
          {u → RATIO, v -> composite[RATIO, cross[INVERSE, INVERSE]]}]

Out[31]= subclass[RATIO, composite[RATIO, cross[INVERSE, INVERSE]]] == True

In[32]:= % /. Equal → SetDelayed
```

Lemma.

```
In[35]:= ImageComp[INVERSE, id[Z], complement[set[id[omega]]]] // Reverse

Out[35]= image[INVERSE, intersection[Z, complement[set[id[omega]]]]] ==
          intersection[Z, complement[set[id[omega]]]]

In[36]:= image[INVERSE, intersection[Z, complement[set[id[omega]]]]] :=
          intersection[Z, complement[set[id[omega]]]]
```

Lemma.

```
In[39]:= SubstTest[composite, t, id[domain[t]],
           t -> composite[RATIO, cross[INVERSE, INVERSE]]] // Reverse

Out[39]= composite[RATIO,
           cross[composite[INVERSE, id[intersection[Z, complement[set[id[omega]]]]]],
            composite[id[Z], INVERSE]]] == composite[RATIO, cross[INVERSE, INVERSE]]

In[40]:= % /. Equal → SetDelayed
```

Theorem.

```
In[41]:= SubstTest[implies, and[subclass[u, v], FUNCTION[v]],
           equal[u, composite[v, id[domain[u]]]],
           {u → RATIO, v -> composite[RATIO, cross[INVERSE, INVERSE]]}] // Reverse

Out[41]= equal[RATIO, composite[RATIO, cross[INVERSE, INVERSE]]] == True

In[42]:= composite[RATIO, cross[INVERSE, INVERSE]] := RATIO
```

One application of this result will be derived.

Lemma.

```
In[44]:= ImageComp[RATIO, cross[INVERSE, INVERSE], cart[x, Z]] // Reverse

Out[44]= image[RATIO, cart[image[INVERSE, x], Z]] == image[RATIO, cart[x, Z]]

In[45]:= image[RATIO, cart[image[INVERSE, x_], Z]] := image[RATIO, cart[x, Z]]
```

Theorem. Every rational number can be written as a fraction with a positive denominator.

```
In[46]:= SubstTest[image, RATIO, union[u, v],
           {u → cart[range[PLUS], Z], v → cart[image[INVERSE, range[PLUS]], Z]}]

Out[46]= image[RATIO, cart[range[PLUS], Z]] == RATS

In[47]:= image[RATIO, cart[range[PLUS], Z]] := RATS
```