multiplying positive rationals

Johan G. F. Belinfante
2012 October 24

```
In[1]:= SetDirectory["l:"]; << goedel.12oct23a

        :Package Title: goedel.12oct23a              2012 October 23 at 8:30 a.m.

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

        It is now:  2012 Oct 24 at 18:37

        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 24 at 18:53
```

summary

The product of non-negative rationals is non-negative.

temporary abbreviations

The following are temporary abbreviations for the sets of non-negative and non-positive rationals.

```
In[8]:= POSRATS := intersection[RATS, P[complement[cart[range[PLUS], complement[range[PLUS]]]]]]

In[9]:= NEGRATS := intersection[RATS,
          P[complement[cart[image[INVERSE, range[PLUS]], complement[range[PLUS]]]]]]
```

simplification rules

Lemma.

```
In[20]:= equiv[and[equal[x, cart[Z, set[id[omega]]]], member[x, V]],
          equal[x, cart[Z, set[id[omega]]]]]

Out[20]= True
```

```
In[21]:= and[equal[x_, cart[Z, set[id[omega]]]], member[x_, V]] :=
          equal[x, cart[Z, set[id[omega]]]]
```

Lemma.

```
In[23]:= SubstTest[subclass, x, intersection[u, v],
          {u -> range[PLUS], v → image[INVERSE, range[PLUS]]}]

Out[23]= and[subclass[x, image[INVERSE, range[PLUS]]], subclass[x, range[PLUS]]] ==
          subclass[x, set[id[omega]]]

In[24]:= and[subclass[x_, image[INVERSE, range[PLUS]]], subclass[x_, range[PLUS]]] :=
          subclass[x, set[id[omega]]]
```

Lemma.

```
In[25]:= SubstTest[member, rat[x], intersection[u, v], {u → NEGRATS, v→ POSRATS}]

Out[25]= subclass[image[rat[x], range[PLUS]], set[id[omega]]] ==
          equal[cart[Z, set[id[omega]]], rat[x]]

In[26]:= subclass[image[rat[x_], range[PLUS]], set[id[omega]]] :=
          equal[cart[Z, set[id[omega]]], rat[x]]
```

Lemma. If a rational number is both non-negative and non-positive, then it is zero.

```
In[28]:= SubstTest[member, x, intersection[u, v], {u → NEGRATS, v→ POSRATS}]

Out[28]= and[member[x, RATS], subclass[image[x, image[INVERSE, range[PLUS]]], range[PLUS]],
            subclass[image[x, range[PLUS]], range[PLUS]]] == equal[x, cart[Z, set[id[omega]]]]

In[29]:= and[member[x_, RATS], subclass[image[x_, image[INVERSE, range[PLUS]]], range[PLUS]],
            subclass[image[x_, range[PLUS]], range[PLUS]]] := equal[x, cart[Z, set[id[omega]]]]
```

Theorem.

```
In[32]:= SubstTest[composite, INVERSE, rat[t], t → ratmul[rat[x], rat[y]]] // Reverse

Out[32]= composite[INVERSE, ratmul[rat[x], rat[y]]] == composite[ratmul[rat[x], rat[y]], INVERSE]

In[33]:= composite[INVERSE, ratmul[rat[x_], rat[y_]]] :=
          composite[ratmul[rat[x], rat[y]], INVERSE]
```

Theorem.

```
In[34]:= ImageComp[INVERSE, ratmul[rat[x], rat[y]], z]

Out[34]= image[ratmul[rat[x], rat[y]], image[INVERSE, z]] ==
          image[INVERSE, image[ratmul[rat[x], rat[y]], z]]

In[35]:= image[ratmul[rat[x_], rat[y_]], image[INVERSE, z_]] :=
          image[INVERSE, image[ratmul[rat[x], rat[y]], z]]
```

Theorem.

```
In[36]:= SubstTest[subclass, image[INVERSE, image[rat[t], x]],
           y, t → ratmul[rat[u], rat[v]]] // Reverse

Out[36]= subclass[image[INVERSE, image[ratmul[rat[u], rat[v]], x]], y] ==
          subclass[image[ratmul[rat[u], rat[v]], x], image[INVERSE, y]]

In[37]:= subclass[image[INVERSE, image[ratmul[rat[u_], rat[v_]], x_]], y_] :=
          subclass[image[ratmul[rat[u], rat[v]], x], image[INVERSE, y]]
```

---

main theorem

Theorem. The product of rationals is either non-negative or non-positive.

```
In[38]:= SubstTest[implies, member[t, RATS],
           or[invariant[t, range[PLUS]], invariant[composite[t, INVERSE], range[PLUS]]],
           t → ratmul[rat[x], rat[y]]] // Reverse

Out[38]= or[subclass[image[ratmul[rat[x], rat[y]], range[PLUS]], image[INVERSE, range[PLUS]]],
           subclass[image[ratmul[rat[x], rat[y]], range[PLUS]], range[PLUS]]] == True

In[39]:= (% /. {x → x_, y → y_}) /. Equal→ SetDelayed
```

Lemma. If the product of rationals is non-negative, then their composite leaves **range[PLUS]** invariant.

```
In[40]:= SubstTest[implies,
           and[subclass[u, v], invariant[v, range[PLUS]]], invariant[u, range[PLUS]],
           {u → composite[rat[x], rat[y]], v → ratmul[rat[x], rat[y]]}] // Reverse

Out[40]= or[not[subclass[image[ratmul[rat[x], rat[y]], range[PLUS]], range[PLUS]]],
           subclass[image[rat[x], image[rat[y], range[PLUS]]], range[PLUS]]] == True

In[41]:= (% /. {x → x_, y → y_}) /. Equal→ SetDelayed
```

Theorem.

```
In[43]:= SubstTest[subclass, image[rat[t], range[PLUS]],
           set[id[omega]], t → ratmul[rat[x], rat[y]]] // Reverse

Out[43]= subclass[image[ratmul[rat[x], rat[y]], range[PLUS]], set[id[omega]]] ==
          or[equal[cart[Z, set[id[omega]]], rat[x]], equal[cart[Z, set[id[omega]]], rat[y]]]

In[44]:= subclass[image[ratmul[rat[x_], rat[y_]], range[PLUS]], set[id[omega]]] :=
          or[equal[cart[Z, set[id[omega]]], rat[x]], equal[cart[Z, set[id[omega]]], rat[y]]]
```

Lemma.

```
In[45]:= SubstTest[implies,
           and[subclass[u, v], subclass[image[v, w], z]], subclass[image[u, w], z],
           {u → composite[rat[x], rat[y]], v → ratmul[rat[x], rat[y]]}] // Reverse

Out[45]= or[not[subclass[image[ratmul[rat[x], rat[y]], w], z]],
           subclass[image[rat[x], image[rat[y], w]], z]] == True

In[46]:= (% /. {w → w_, x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Theorem.

```
In[47]:= or[not[subclass[x, range[PLUS]]], not[subclass[x, image[INVERSE, range[PLUS]]]],
           subclass[x, set[id[omega]]]] // NotNotTest

Out[47]= or[not[subclass[x, image[INVERSE, range[PLUS]]]],
           not[subclass[x, range[PLUS]]], subclass[x, set[id[omega]]]] == True

In[48]:= or[not[subclass[x_, image[INVERSE, range[PLUS]]]],
           not[subclass[x_, range[PLUS]]], subclass[x_, set[id[omega]]]] := True
```

Lemma.

```
In[49]:= Map[subclass[#, set[id[omega]]] &, SubstTest[image, composite[rat[x], rat[y]],
           union[u, v], {u → range[PLUS], v → image[INVERSE, range[PLUS]]}]]

Out[49]= subclass[image[rat[x], image[rat[y], range[PLUS]]], set[id[omega]]] ==
          subclass[image[rat[x], range[rat[y]]], set[id[omega]]]

In[50]:= subclass[image[rat[x_], image[rat[y_], range[PLUS]]], set[id[omega]]] :=
          subclass[image[rat[x], range[rat[y]]], set[id[omega]]]
```

Theorem.

```
In[51]:= SubstTest[implies, and[not[subclass[domain[t], set[id[omega]]]], subclass[t, rat[w]]],
           equal[hull[RATS, t], rat[w]],
           {t → composite[rat[x], rat[y]], w → cart[Z, set[id[omega]]]}] // Reverse

Out[51]= or[equal[cart[Z, set[id[omega]]], rat[x]], equal[cart[Z, set[id[omega]]], rat[y]],
           not[subclass[image[rat[x], range[rat[y]]], set[id[omega]]]]] == True

In[52]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[53]:= or[not[
            subclass[image[rat[x], image[rat[y], range[PLUS]]], image[INVERSE, range[PLUS]]]],
           not[subclass[image[rat[x], image[rat[y], range[PLUS]]], range[PLUS]]],
           subclass[image[rat[x], range[rat[y]]], set[id[omega]]]] // NotNotTest

Out[53]= or[not[
            subclass[image[rat[x], image[rat[y], range[PLUS]]], image[INVERSE, range[PLUS]]]],
           not[subclass[image[rat[x], image[rat[y], range[PLUS]]], range[PLUS]]],
           subclass[image[rat[x], range[rat[y]]], set[id[omega]]]] == True
```

```
In[54]:= (% /. {x → x_, y → y_}) /. Equal→ SetDelayed
```

Lemma.

```
In[55]:= or[and[not[equal[cart[Z, set[id[omega]]], rat[x]]],
           not[equal[cart[Z, set[id[omega]]], rat[y]]]],
         subclass[image[ratmul[rat[x], rat[y]], range[PLUS]], range[PLUS]]] // NotNotTest

Out[55]= or[and[not[equal[cart[Z, set[id[omega]]], rat[x]]],
           not[equal[cart[Z, set[id[omega]]], rat[y]]]],
         subclass[image[ratmul[rat[x], rat[y]], range[PLUS]], range[PLUS]]] == True

In[56]:= (% /. {x → x_, y → y_}) /. Equal→ SetDelayed
```

Theorem. The product of two non-negative rationals is non-negative.

```
In[57]:= Map[not, SubstTest[and, implies[p1, p4],
           implies[p2, p3], implies[p3, p5], (*implies[and[p4,p5],p6],*)
           implies[p6, p7], implies[p7, not[p2]], not[implies[p1, not[p2]]],
           {p1 → and[subclass[image[rat[x], range[PLUS]], range[PLUS]],
              subclass[image[rat[y], range[PLUS]], range[PLUS]]],
            p2 → not[subclass[image[ratmul[rat[x], rat[y]], range[PLUS]], range[PLUS]]], p3 →
             subclass[image[ratmul[rat[x], rat[y]], range[PLUS]], image[INVERSE, range[PLUS]]],
            p4 → subclass[image[rat[x], image[rat[y], range[PLUS]]], range[PLUS]], p5 →
             subclass[image[rat[x], image[rat[y], range[PLUS]]], image[INVERSE, range[PLUS]]],
            p6 → subclass[image[rat[x], image[rat[y], range[PLUS]]], set[id[omega]]],
            p7 -> or[equal[cart[Z, set[id[omega]]], rat[x]],
              equal[cart[Z, set[id[omega]]], rat[y]]]}]] // Reverse

Out[57]= or[not[subclass[image[rat[x], range[PLUS]], range[PLUS]]],
         not[subclass[image[rat[y], range[PLUS]], range[PLUS]]],
         subclass[image[ratmul[rat[x], rat[y]], range[PLUS]], range[PLUS]]] == True

In[58]:= or[not[subclass[image[rat[x_], range[PLUS]], range[PLUS]]],
         not[subclass[image[rat[y_], range[PLUS]], range[PLUS]]],
         subclass[image[ratmul[rat[x_], rat[y_]], range[PLUS]], range[PLUS]]] := True
```

Corollary. (Eliminate the **rat** wrappers.)

```
In[59]:= SubstTest[implies, and[equal[x, rat[u]], equal[y, rat[v]]],
         or[not[subclass[image[x, range[PLUS]], range[PLUS]]],
          not[subclass[image[y, range[PLUS]], range[PLUS]]],
          subclass[image[ratmul[x, y], range[PLUS]], range[PLUS]]], {u → x, v → y}] // Reverse

Out[59]= or[not[member[x, RATS]], not[member[y, RATS]],
         not[subclass[image[x, range[PLUS]], range[PLUS]]],
         not[subclass[image[y, range[PLUS]], range[PLUS]]],
         subclass[image[ratmul[x, y], range[PLUS]], range[PLUS]]] == True

In[60]:= or[not[member[x_, RATS]], not[member[y_, RATS]],
         not[subclass[image[x_, range[PLUS]], range[PLUS]]],
         not[subclass[image[y_, range[PLUS]], range[PLUS]]],
         subclass[image[ratmul[x_, y_], range[PLUS]], range[PLUS]]] := True
```

Lemma.

```
In[61]:= Map[empty[composite[Id, complement[#]]] &,
          dif[cart[POSRATS, POSRATS], image[inverse[RATMUL], POSRATS]] // complement //
           RelnNormality]

Out[61]= subclass[image[
            U[image[RATMUL, cart[P[complement[cart[range[PLUS], complement[range[PLUS]]]]],
               P[complement[cart[range[PLUS], complement[range[PLUS]]]]]]]],
            range[PLUS]], range[PLUS]] == True

In[62]:= % /. Equal → SetDelayed
```

Lemma.

```
In[63]:= ImageComp[RATMUL, LEFT[id[Z]], x] // Reverse

Out[63]= image[RATMUL, cart[set[id[Z]], x]] == intersection[RATS, x]

In[64]:= image[RATMUL, cart[set[id[Z]], x_]] := intersection[RATS, x]
```

lemma.

```
In[65]:= SubstTest[implies, subclass[u, v], subclass[image[RATMUL, u], image[RATMUL, v]],
           {u → cart[set[id[Z]], POSRATS], v → cart[POSRATS, POSRATS]}] // Reverse

Out[65]= subclass[
           intersection[RATS, P[complement[cart[range[PLUS], complement[range[PLUS]]]]]],
           image[RATMUL, cart[P[complement[cart[range[PLUS], complement[range[PLUS]]]]],
             P[complement[cart[range[PLUS], complement[range[PLUS]]]]]]] == True

In[66]:= % /. Equal → SetDelayed
```

Theorem. The product of non-negative rationals is non-negative.

```
In[67]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u → image[RATMUL, cartsq[POSRATS]], v → POSRATS}]

Out[67]= equal[image[RATMUL, cart[P[complement[cart[range[PLUS], complement[range[PLUS]]]]],
             P[complement[cart[range[PLUS], complement[range[PLUS]]]]]], intersection[
            RATS, P[complement[cart[range[PLUS], complement[range[PLUS]]]]]] == True

In[69]:= image[RATMUL, cart[P[complement[cart[range[PLUS], complement[range[PLUS]]]]],
            P[complement[cart[range[PLUS], complement[range[PLUS]]]]]] :=
           intersection[RATS, P[complement[cart[range[PLUS], complement[range[PLUS]]]]]]
```

Restatement.

```
In[70]:= member[POSRATS, binclosed[RATMUL]]

Out[70]= True
```

Corollary. The non-negative rationals form a monoid under multiplication.

```
In[71]:= SubstTest[implies, and[member[x, MONOIDS], member[y, binclosed[x]], member[e[x], y]],
           member[composite[x, id[cartsq[y]]], MONOIDS], {x → RATMUL, y → POSRATS}] // Reverse

Out[71]= member[
           composite[RATMUL, id[cart[P[complement[cart[range[PLUS], complement[range[PLUS]]]]],
              P[complement[cart[range[PLUS], complement[range[PLUS]]]]]]], MONOIDS] == True

In[72]:= member[
           composite[RATMUL, id[cart[P[complement[cart[range[PLUS], complement[range[PLUS]]]]],
              P[complement[cart[range[PLUS], complement[range[PLUS]]]]]]], MONOIDS] := True
```