composites of rationals

Johan G. F. Belinfante
2012 August 3

```
In[1]:= SetDirectory["l:"]; << goedel.12jul29a

:Package Title: goedel.12jul29a                  2012 July 29 at 7:50 a.m.

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

It is now:  2012 Aug 3 at 10:4

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 Aug 3 at 10:19
```

summary

If **x** and **y** are rational numbers, then there exists a rational number **z** such that $\mathbf{x} \circ \mathbf{y} \subset \mathbf{z}$.

derivation

Lemma. An inclusion.

```
In[2]:= SubstTest[implies, subclass[r, s], subclass[composite[q, r, t], composite[q, s, t]],
          {q -> inverse[inttimes[u]], r → composite[inttimes[v], inverse[inttimes[x]]],
           s -> composite[inverse[inttimes[x]], inttimes[v]], t → inttimes[y]}] // Reverse

Out[2]= subclass[
          composite[inverse[inttimes[u]], inttimes[v], inverse[inttimes[x]], inttimes[y]],
          composite[inverse[inttimes[intmul[u, x]]], inttimes[intmul[v, y]]]] == True

In[3]:= subclass[
          composite[inverse[inttimes[u_]], inttimes[v_], inverse[inttimes[x_]], inttimes[y_]],
          composite[inverse[inttimes[intmul[u_, x_]]], inttimes[intmul[v_, y_]]]] := True
```

Lemma.

```
In[4]:= SubstTest[implies, and[subclass[u, v], member[v, w]],
          member[u, image[inverse[S], w]], {u -> composite[APPLY[RATIO, x], APPLY[RATIO, y]],
           v -> composite[inverse[inttimes[first[x]]], inverse[inttimes[first[y]]],
             inttimes[second[x]], inttimes[second[y]]], w → RATS}] // Reverse

Out[4]= or[equal[first[x], id[omega]], equal[first[y], id[omega]],
          member[composite[APPLY[RATIO, x], APPLY[RATIO, y]], image[inverse[S], RATS]],
          not[member[first[x], Z]], not[member[first[y], Z]], not[member[second[x], Z]],
          not[member[second[y], Z]], not[subclass[composite[APPLY[RATIO, x], APPLY[RATIO, y]],
            composite[inverse[inttimes[intmul[first[x], first[y]]]],
             inttimes[intmul[second[x], second[y]]]]]]] == True

In[5]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[6]:= Map[not, SubstTest[and, implies[p1, p2], implies[p1, p3], implies[and[p2, p3], p4],
           implies[p4, p5], implies[and[p1, p5], p6], not[implies[p1, p6]],
           {p1 → and[member[x, domain[RATIO]], member[y, domain[RATIO]]], p2 → equal[
              APPLY[RATIO, x], composite[inverse[inttimes[first[x]]], inttimes[second[x]]]],
            p3 → equal[APPLY[RATIO, y], composite[inverse[inttimes[first[y]]],
               inttimes[second[y]]]], p4 → subclass[composite[APPLY[RATIO, x], APPLY[RATIO, y]],
              composite[inverse[inttimes[first[x]]], inttimes[second[x]],
               inverse[inttimes[first[y]]], inttimes[second[y]]]],
            p5 -> subclass[composite[APPLY[RATIO, x], APPLY[RATIO, y]],
              composite[inverse[inttimes[first[x]]], inverse[inttimes[first[y]]],
               inttimes[second[x]], inttimes[second[y]]]], p6 → member[
              composite[APPLY[RATIO, x], APPLY[RATIO, y]], image[inverse[S], RATS]]}]] // Reverse

Out[6]= or[equal[first[x], id[omega]], equal[first[y], id[omega]],
          member[composite[APPLY[RATIO, x], APPLY[RATIO, y]], image[inverse[S], RATS]],
          not[member[first[x], Z]], not[member[first[y], Z]],
          not[member[second[x], Z]], not[member[second[y], Z]]] == True

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

Lemma. A membership rule.

```
In[8]:= (member[pair[x, y], composite[inverse[funpart[t]], z, funpart[t]]] // AssertTest) /.
         t → RATIO

Out[8]= member[pair[x, y], composite[inverse[RATIO], z, RATIO]] == and[member[first[x], Z],
          member[first[y], Z], member[pair[APPLY[RATIO, x], APPLY[RATIO, y]], z],
          member[second[x], Z], member[second[y], Z],
          not[equal[first[x], id[omega]]], not[equal[first[y], id[omega]]]]

In[9]:= member[pair[x_, y_], composite[inverse[RATIO], z_, RATIO]] := and[member[first[x], Z],
          member[first[y], Z], member[pair[APPLY[RATIO, x], APPLY[RATIO, y]], z],
          member[second[x], Z], member[second[y], Z],
          not[equal[first[x], id[omega]]], not[equal[first[y], id[omega]]]]
```

Main Theorem. The composite of two rational numbers is a subset of a rational number.

```
In[10]:= Map[empty[composite[Id, complement[#]]] &, SubstTest[class, pair[x, y],
           implies[member[pair[x, y], u], member[pair[x, y], v]], {u → cartsq[domain[RATIO]], v →
             image[inverse[composite[COMPOSE, cross[RATIO, RATIO]]], image[inverse[S], RATS]]}]]

Out[10]= subclass[image[COMPOSE, cart[RATS, RATS]], image[inverse[S], RATS]] == True

In[11]:= subclass[image[COMPOSE, cart[RATS, RATS]], image[inverse[S], RATS]] := True
```