intersections of rationals

Johan G. F. Belinfante
2012 July 24

```
In[1]:= SetDirectory["l:"]; << goedel.12jul23a

        :Package Title: goedel.12jul23a                 2012 July 23 at 8:55 a.m.

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

        It is now:  2012 Jul 24 at 10:43

        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 Jul 24 at 10:59
```

summary

If $\mathbf{x}$ and $\mathbf{y}$ are rational numbers, then either $\mathbf{x} = \mathbf{y}$ or $\mathbf{x} \bigcap \mathbf{y} = \{\mathbf{id}[\omega]\} \times \{\mathbf{id}[\omega]\}$.

general results

Theorem. A general result.

```
In[2]:= SubstTest[implies, equal[u, v], equal[union[u, w], union[v, w]],
          {u → dif[x, z], v → dif[y, z], w → z}] // Reverse

Out[2]= or[equal[union[x, z], union[y, z]],
          not[equal[intersection[x, complement[z]], intersection[y, complement[z]]]]] == True

In[3]:= or[equal[union[x_, z_], union[y_, z_]], not[
           equal[intersection[complement[z_], x_], intersection[complement[z_], y_]]]] := True
```

Theorem. Another general result.

```
In[4]:= SubstTest[implies, and[equal[t, u], equal[v, w]], equal[union[t, v], union[u, w]],
          {t → dif[x, z], u → dif[y, z], v → intersection[x, z], w → intersection[y, z]}] // Reverse

Out[4]= or[equal[x, y], not[equal[intersection[x, z], intersection[y, z]]],
          not[equal[intersection[x, complement[z]], intersection[y, complement[z]]]]] == True
```

```
In[5]:= or[equal[x_, y_],
          not[equal[intersection[complement[z_], x_], intersection[complement[z_], y_]]],
          not[equal[intersection[x_, z_], intersection[y_, z_]]]] := True
```

Corollary.

```
In[6]:= Map[not, SubstTest[and, implies[and[p2, p3], p4],
           implies[and[p1, p4], p5], not[implies[and[p1, p2, p3], p5]],
           {p1 -> equal[dif[x, z], dif[y, z]], p2 -> subclass[z, x], p3 -> subclass[z, y],
            p4 → equal[intersection[x, z], intersection[y, z]], p5 → equal[x, y]}]] // Reverse
```

```
Out[6]= or[equal[x, y],
          not[equal[intersection[x, complement[z]], intersection[y, complement[z]]]],
          not[subclass[z, x]], not[subclass[z, y]]] == True
```

```
In[7]:= or[equal[x_, y_],
          not[equal[intersection[complement[z_], x_], intersection[complement[z_], y_]]],
          not[subclass[z_, x_]], not[subclass[z_, y_]]] := True
```

Corollary.

```
In[8]:= Map[implies[and[member[w, x], member[w, y]], #] &,
         SubstTest[implies, and[equal[dif[x, t], dif[y, t]], subclass[t, x], subclass[t, y]],
           equal[x, y], t → set[w]] // Reverse]
```

```
Out[8]= or[equal[x, y],
          not[equal[intersection[x, complement[set[w]]], intersection[y, complement[set[w]]]]],
          not[member[w, x]], not[member[w, y]]] == True
```

```
In[9]:= or[equal[x_, y_], not[equal[
            intersection[complement[set[w_]], x_], intersection[complement[set[w_]], y_]]],
          not[member[w_, x_]], not[member[w_, y_]]] := True
```

---

derivation

Lemma. Introduce variables.

```
In[10]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u → pair[x, y], v -> cartsq[image[IMAGE[id[complement[id[set[id[omega]]]]]], RATS]],
            w -> union[Id, DISJOINT]}] // Reverse
```

```
Out[10]= or[equal[0, intersection[x, y]], equal[x, y],
          not[member[x, image[IMAGE[id[cart[complement[set[id[omega]]], V]]], RATS]]],
          not[member[y, image[IMAGE[id[cart[complement[set[id[omega]]], V]]], RATS]]]] == True
```

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

Lemma. For rationals, removing the point at the origin is equivalent to removing the vertical axis in the $\mathbf{Z} \times \mathbf{Z}$ plane.

```
In[12]:= SubstTest[implies, member[x, z], member[dif[x, y], image[IMAGE[id[complement[y]]], z]],
           {y -> cart[set[id[omega]], set[id[omega]]], z → RATS}] // Reverse

Out[12]= or[member[intersection[x, complement[cart[set[id[omega]], set[id[omega]]]]],
            image[IMAGE[id[cart[complement[set[id[omega]]], V]]], RATS]],
           not[member[x, RATS]]] == True

In[13]:= (% /. {x → x_, y → y_}) /. Equal→ SetDelayed
```

Lemma.

```
In[14]:= SubstTest[or, equal[0, intersection[u, v]], equal[u, v],
           not[member[u, image[IMAGE[id[cart[complement[set[id[omega]]], V]]], RATS]]],
           not[member[v, image[IMAGE[id[cart[complement[set[id[omega]]], V]]], RATS]]],
           {u → dif[x, cart[set[id[omega]], set[id[omega]]]],
            v → dif[y, cart[set[id[omega]], set[id[omega]]]]}] // Reverse

Out[14]= or[equal[intersection[x, complement[cart[set[id[omega]], set[id[omega]]]]],
            intersection[y, complement[cart[set[id[omega]], set[id[omega]]]]]],
           not[member[intersection[x, complement[cart[set[id[omega]], set[id[omega]]]]],
             image[IMAGE[id[cart[complement[set[id[omega]]], V]]], RATS]]],
           not[member[intersection[y, complement[cart[set[id[omega]], set[id[omega]]]]],
             image[IMAGE[id[cart[complement[set[id[omega]]], V]]], RATS]]],
           subclass[intersection[x, y], cart[set[id[omega]], set[id[omega]]]]] == True

In[15]:= (% /. {x → x_, y → y_}) /. Equal→ SetDelayed
```

Lemma.

```
In[16]:= Map[not,
           SubstTest[and, implies[p1, p2], implies[p1, p3], implies[and[p2, p3], or[p4, p5]],
            not[implies[p1, or[p4, p5]]], {p1 → member[pair[x, y], cart[RATS, RATS]],
             p2 → member[intersection[x, complement[cart[set[id[omega]], set[id[omega]]]]],
               image[IMAGE[id[cart[complement[set[id[omega]]], V]]], RATS]],
             p3 -> member[intersection[y, complement[cart[set[id[omega]], set[id[omega]]]]],
               image[IMAGE[id[cart[complement[set[id[omega]]], V]]], RATS]],
             p4 → equal[intersection[x, complement[cart[set[id[omega]], set[id[omega]]]]],
               intersection[y, complement[cart[set[id[omega]], set[id[omega]]]]]],
             p5 → disjoint[intersection[x, complement[cart[set[id[omega]], set[id[omega]]]]],
               intersection[y, complement[cart[set[id[omega]], set[id[omega]]]]]]}]] // Reverse

Out[16]= or[equal[intersection[x, complement[cart[set[id[omega]], set[id[omega]]]]],
            intersection[y, complement[cart[set[id[omega]], set[id[omega]]]]]],
           not[member[x, RATS]], not[member[y, RATS]],
           subclass[intersection[x, y], cart[set[id[omega]], set[id[omega]]]]] == True

In[17]:= (% /. {x → x_, y → y_}) /. Equal→ SetDelayed
```

Corollary. (A particular case of a general result that is needed here.)

```
In[18]:= SubstTest[or, equal[x, y], not[
           equal[intersection[x, complement[set[w]]], intersection[y, complement[set[w]]]],
          not[member[w, x]], not[member[w, y]], w → PAIR[id[omega], id[omega]]] // Reverse

Out[18]= or[equal[x, y],
          not[equal[intersection[x, complement[cart[set[id[omega]], set[id[omega]]]]],
            intersection[y, complement[cart[set[id[omega]], set[id[omega]]]]]],
          not[member[pair[id[omega], id[omega]], x]],
          not[member[pair[id[omega], id[omega]], y]]] == True

In[19]:= (% /. {x → x_, y → y_}) /. Equal→ SetDelayed
```

Lemma.

```
In[20]:= Map[implies[#, equal[cart[set[id[omega]], set[id[omega]]], intersection[x, y]]] &,
          SubstTest[and, subclass[u, v], subclass[v, u],
           {u -> intersection[x, y], v -> cart[set[id[omega]], set[id[omega]]]}]] // Reverse

Out[20]= or[equal[cart[set[id[omega]], set[id[omega]]], intersection[x, y]],
          not[member[pair[id[omega], id[omega]], x]],
          not[member[pair[id[omega], id[omega]], y]],
          not[subclass[intersection[x, y], cart[set[id[omega]], set[id[omega]]]]]] == True

In[21]:= (% /. {x → x_, y → y_}) /. Equal→ SetDelayed
```

Theorem.

```
In[22]:= Map[not, SubstTest[and, implies[p1, or[p2, p3]], implies[p1, p4],
           implies[p1, p5], (*implies[and[p2,p4,p5],p6],implies[and[p3,p4,p5],p7],*)
           not[implies[p1, or[p6, p7]]], {p1 -> member[pair[x, y], cart[RATS, RATS]],
            p2 -> equal[intersection[x, complement[cart[set[id[omega]], set[id[omega]]]]],
              intersection[y, complement[cart[set[id[omega]], set[id[omega]]]]]],
            p3 -> subclass[intersection[x, y], cart[set[id[omega]], set[id[omega]]]],
            p4 → subclass[cart[set[id[omega]], set[id[omega]]], x],
            p5 → subclass[cart[set[id[omega]], set[id[omega]]], y], p6 → equal[x, y],
            p7 → equal[intersection[x, y], cart[set[id[omega]], set[id[omega]]]]}]] // Reverse

Out[22]= or[equal[x, y], equal[cart[set[id[omega]], set[id[omega]]], intersection[x, y]],
          not[member[x, RATS]], not[member[y, RATS]]] == True

In[23]:= or[equal[cart[set[id[omega]], set[id[omega]]], intersection[x_, y_]],
          equal[x_, y_], not[member[x_, RATS]], not[member[y_, RATS]]] := True
```