gcd of a union

Johan G. F. Belinfante
2009 April 7

```
In[1]:= SetDirectory["l:"]; << goedel.09apr06a; << tools.m

        :Package Title: goedel.09apr06a            2009 April 6 at 9:45 a.m.

        It is now:  2009 Apr 7 at 3:42

        Loading Simplification Rules

        TOOLS.M                                    Revised 2009 April 6

        weightlimit = 40
```

summary

The gcd of a union of two subsets of **omega** can be computed in two stages:

```
In[2]:= equal[APPLY[GLB[DIV], set[APPLY[GLB[DIV], x], APPLY[GLB[DIV], y]]],
          APPLY[GLB[DIV], union[x, y]]]

Out[2]= and[subclass[x, omega], subclass[y, omega]]
```

This equation can be turned into a rewrite rule for the right-hand side. One can then use reify to obtain a variable-free expression of this fact. As an application this formula is used to show that **omega** is a semilattice under the binary operation **composite[GLB[DIV],PAIRSET]**.

derivation

Theorem. A rewrite rule.

```
In[3]:= equal[APPLY[GLB[DIV], union[x, y]],
          union[APPLY[GLB[DIV], set[APPLY[GLB[DIV], x], APPLY[GLB[DIV], y]]],
           image[V, intersection[x, complement[omega]]],
           image[V, intersection[y, complement[omega]]]]]

Out[3]= True

In[4]:= APPLY[GLB[DIV], union[x_, y_]] :=
          union[APPLY[GLB[DIV], set[APPLY[GLB[DIV], x], APPLY[GLB[DIV], y]]],
           image[V, intersection[x, complement[omega]]],
           image[V, intersection[y, complement[omega]]]]
```

Corollary. A variable-free restatement. (This takes quite a while.)

```
In[5]:= SubstTest[reify, x, image[t, set[union[first[x], second[x]]]], t -> GLB[DIV]] // Reverse

Out[5]= composite[GLB[DIV], PAIRSET, cross[GLB[DIV], GLB[DIV]]] == composite[GLB[DIV], CUP]

In[6]:= composite[GLB[DIV], PAIRSET, cross[GLB[DIV], GLB[DIV]]] := composite[GLB[DIV], CUP]
```

---

an associative law

Lemma. (Simplification rule.)

```
In[7]:= composite[GLB[DIV], PAIRSET,
          cross[composite[GLB[DIV], PAIRSET], id[omega]]] // TripleRotate

Out[7]= composite[GLB[DIV], PAIRSET, cross[composite[GLB[DIV], PAIRSET], id[omega]]] ==
         composite[GLB[DIV], PAIRSET, cross[composite[GLB[DIV], PAIRSET], Id]]

In[8]:= composite[GLB[DIV], PAIRSET, cross[composite[GLB[DIV], PAIRSET], id[omega]]] :=
         composite[GLB[DIV], PAIRSET, cross[composite[GLB[DIV], PAIRSET], Id]]
```

Theorem.

```
In[9]:= Assoc[composite[GLB[DIV], PAIRSET],
         cross[GLB[DIV], GLB[DIV]], cross[PAIRSET, SINGLETON]]

Out[9]= composite[GLB[DIV], PAIRSET, cross[composite[GLB[DIV], PAIRSET], Id]] ==
         composite[GLB[DIV], CUP, cross[PAIRSET, SINGLETON]]

In[10]:= composite[GLB[DIV], PAIRSET, cross[composite[GLB[DIV], PAIRSET], Id]] :=
          composite[GLB[DIV], CUP, cross[PAIRSET, SINGLETON]]
```

Corollary.

```
In[11]:= composite[GLB[DIV], PAIRSET, cross[Id, composite[GLB[DIV], PAIRSET]]] // TripleRotate

Out[11]= composite[GLB[DIV], PAIRSET, cross[Id, composite[GLB[DIV], PAIRSET]]] ==
          composite[GLB[DIV], CUP, cross[SINGLETON, PAIRSET]]

In[12]:= composite[GLB[DIV], PAIRSET, cross[Id, composite[GLB[DIV], PAIRSET]]] :=
          composite[GLB[DIV], CUP, cross[SINGLETON, PAIRSET]]
```

Theorem. An associative law.

```
In[13]:= SubstTest[implies, and[subclass[x, cart[cart[V, V], V]],
            equal[composite[x, cross[x, Id]], composite[x, cross[Id, x], ASSOC]]],
           associative[x], x → composite[GLB[DIV], PAIRSET]] // Reverse

Out[13]= associative[composite[GLB[DIV], PAIRSET]] == True

In[14]:= associative[composite[GLB[DIV], PAIRSET]] := True
```

semilattice property

The semilattice property is established here through a series of lemmas.

Lemma. A sethood result.

```
In[15]:= member[composite[GLB[DIV], PAIRSET], V] // AssertTest
Out[15]= member[composite[GLB[DIV], PAIRSET], V] == True

In[16]:= member[composite[GLB[DIV], PAIRSET], V] := True
```

Theorem. Mapping property.

```
In[17]:= member[composite[GLB[DIV], PAIRSET], map[cart[omega, omega], omega]] // AssertTest
Out[17]= member[composite[GLB[DIV], PAIRSET], map[cart[omega, omega], omega]] == True

In[18]:= member[composite[GLB[DIV], PAIRSET], map[cart[omega, omega], omega]] := True
```

Theorem. The gcd as a binary operation.

```
In[19]:= SubstTest[implies, and[member[u, v], subclass[v, w]], member[u, w],
           {u -> composite[GLB[DIV], PAIRSET], v → map[cart[omega, omega], omega], w -> BINOPS}] //
          Reverse
Out[19]= member[composite[GLB[DIV], PAIRSET], BINOPS] == True

In[20]:= member[composite[GLB[DIV], PAIRSET], BINOPS] := True
```

Corollary. An associative binary operation is a semigroup.

```
In[21]:= member[composite[GLB[DIV], PAIRSET], SEMIGPS] // AssertTest
Out[21]= member[composite[GLB[DIV], PAIRSET], SEMIGPS] == True

In[22]:= member[composite[GLB[DIV], PAIRSET], SEMIGPS] := True
```

Corollary. A band is a semigroup whose elements are all idempotent.

```
In[23]:= (member[t, BANDS] // AssertTest) /. t -> composite[GLB[DIV], PAIRSET]
Out[23]= member[composite[GLB[DIV], PAIRSET], BANDS] == True

In[24]:= member[composite[GLB[DIV], PAIRSET], BANDS] := True
```

Corollary. A semilattice is a commutative band.

```
In[25]:= member[composite[GLB[DIV], PAIRSET], SEMILATTICES] // AssertTest
Out[25]= member[composite[GLB[DIV], PAIRSET], SEMILATTICES] == True
```

```
In[26]:= member[composite[GLB[DIV], PAIRSET], SEMILATTICES] := True
```