pairsets in a complete lattice

Johan G. F. Belinfante
2009 March 1

```
In[1]:= SetDirectory["l:"]; << goedel.09feb26a; << tools.m

        :Package Title: goedel.09feb26a        2009 February 26 at 11:30 a.m.

        It is now:  2009 Mar 1 at 20:38

        Loading Simplification Rules

        TOOLS.M                                    Revised 2009 February 18

        weightlimit = 40
```

summary

If **x** is a complete lattice, then any subset of **fix[x]** has a glb and lub. In particular pairsets do. Simple rewrite rules about glb and lub for pairsets in a complete lattice are derived in this notebook.

In the notes accompanying the theorems, the notation **u ≤ v** means that **pair[u,v] ∈ x**, and the standard abbreviations **u ∧ v** and **u ∨ v** will be used for the glb and lub respectively of the pairset **set[u, v]**.

derivation

Theorem. If **x** is a complete lattice and **u, v ∈ fix[x]**, then **u ∧ v ∈ fix[x]**.

```
In[2]:= Map[not, SubstTest[and, implies[p2, p3], implies[p2, p4],
           implies[and[p1, p3, p4], p5], not[implies[and[p1, p2], p5]],
           {p1 → member[x, CL], p2 → and[member[u, fix[x]], member[v, fix[x]]],
            p3 → subclass[set[u, v], fix[x]], p4 → or[member[u, V], member[v, V]],
            p5 → member[APPLY[GLB[x], set[u, v]], fix[x]]}]] // Reverse

Out[2]= or[member[APPLY[GLB[x], set[u, v]], fix[x]],
          not[member[u, fix[x]]], not[member[v, fix[x]]], not[member[x, CL]]] == True

In[3]:= or[member[APPLY[GLB[x_], set[u_, v_]], fix[x_]],
          not[member[u_, fix[x_]]], not[member[v_, fix[x_]]], not[member[x_, CL]]] := True
```

Dual Theorem. If **x** is a complete lattice and **u, v ∈ fix[x]**, then **u ∨ v ∈ fix[x]**.

```
In[4]:= Map[implies[member[x, CL], #] &,
          SubstTest[or, member[APPLY[GLB[t], set[u, v]], fix[t]], not[member[u, fix[t]]],
           not[member[v, fix[t]]], not[member[t, CL]], t → inverse[x]]] // Reverse

Out[4]= or[member[APPLY[LUB[x], set[u, v]], fix[x]],
          not[member[u, fix[x]]], not[member[v, fix[x]]], not[member[x, CL]]] == True

In[5]:= or[member[APPLY[LUB[x_], set[u_, v_]], fix[x_]],
          not[member[u_, fix[x_]]], not[member[v_, fix[x_]]], not[member[x_, CL]]] := True
```

Lemma.

```
In[6]:= Map[not, SubstTest[and, implies[p2, p3],
           implies[and[p1, p2, p3], p4], not[implies[and[p1, p2], p4]], {p1 → member[x, CL],
            p2 → and[member[u, fix[x]], member[v, fix[x]]], p3 → subclass[set[u, v], fix[x]],
            p4 → subclass[set[u, v], image[x, set[APPLY[GLB[x], set[u, v]]]]]}]] // Reverse

Out[6]= or[not[member[u, fix[x]]], not[member[v, fix[x]]], not[member[x, CL]],
          subclass[set[u, v], image[x, set[APPLY[GLB[x], set[u, v]]]]]] == True

In[7]:= (% /. {u → u_, v → v_, x → x_}) /. Equal → SetDelayed
```

Lemma.

```
In[8]:= Map[or[#, not[member[u, z]], member[pair[APPLY[GLB[x], set[u, v]], u], x]] &,
          SubstTest[implies, and[member[u, w], subclass[w, y]], member[u, y],
           {w → set[u, v], y -> image[x, set[APPLY[GLB[x], set[u, v]]]]}]] // Reverse

Out[8]= or[member[pair[APPLY[GLB[x], set[u, v]], u], x], not[member[u, z]],
          not[subclass[set[u, v], image[x, set[APPLY[GLB[x], set[u, v]]]]]]] == True

In[9]:= (% /. {u → u_, v → v_, x → x_, z → z_}) /. Equal → SetDelayed
```

Theorem. If **x** is a complete lattice, and **u, v ∈ fix[x]**, then **u ∧ v ≤ u**. (By symmetry, also **u ∧ v ≤ v**. This requires no additional rewrite rule.)

```
In[10]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3], not[implies[p1, p3]],
            {p1 -> and[member[x, CL], member[u, fix[x]], member[v, fix[x]]],
             p2 -> subclass[set[u, v], image[x, set[APPLY[GLB[x], set[u, v]]]]],
             p3 -> member[pair[APPLY[GLB[x], set[u, v]], u], x]}]] // Reverse

Out[10]= or[member[pair[APPLY[GLB[x], set[u, v]], u], x],
           not[member[u, fix[x]]], not[member[v, fix[x]]], not[member[x, CL]]] == True

In[11]:= or[member[pair[APPLY[GLB[x_], set[u_, v_]], u_], x_],
           not[member[u_, fix[x_]]], not[member[v_, fix[x_]]], not[member[x_, CL]]] := True
```

Dual Theorem. If **x** is a complete lattice, and **u, v ∈ fix[x]**, then **u ≤ u ∨ v**.

```
In[12]:= Map[implies[member[x, CL], or[#, member[pair[u, APPLY[LUB[x], set[u, v]]], x]]] &,
           SubstTest[or, member[pair[APPLY[GLB[t], set[u, v]], u], t], not[member[u, fix[t]]],
            not[member[v, fix[t]]], not[member[t, CL]], t → inverse[x]]] // Reverse

Out[12]= or[member[pair[u, APPLY[LUB[x], set[u, v]]], x],
           not[member[u, fix[x]]], not[member[v, fix[x]]], not[member[x, CL]]] == True

In[13]:= or[member[pair[u_, APPLY[LUB[x_], set[u_, v_]]], x_],
           not[member[u_, fix[x_]]], not[member[v_, fix[x_]]], not[member[x_, CL]]] := True
```

Lemma. (This lemma is needed to bring pairsets into play. Introducing the **po[x]** wrapper automatically removes sethood literals.)

```
In[14]:= (Map[implies[and[member[pair[t, u], w], member[pair[t, v], w]], #] &,
            SubstTest[subclass, union[y, z], image[w, set[t]],
             {y → set[u], z → set[v]}]] /. w → po[x]) // Reverse

Out[14]= or[not[member[pair[t, u], po[x]]], not[member[pair[t, v], po[x]]],
           subclass[set[u, v], image[po[x], set[t]]]] == True

In[15]:= (% /. {u → u_, v → v_, x → x_}) /. Equal → SetDelayed
```

Lemma. (The **po** wrapper in the preceding lemma is eliminated.)

```
In[16]:= SubstTest[implies, equal[x, po[w]], or[not[member[pair[t, u], x]],
            not[member[pair[t, v], x]], subclass[set[u, v], image[x, set[t]]]], w → x] // Reverse

Out[16]= or[not[member[pair[t, u], x]], not[member[pair[t, v], x]],
           not[PARTIALORDER[x]], subclass[set[u, v], image[x, set[t]]]] == True

In[17]:= (% /. {t → t_, u → u_, v → v_, x → x_}) /. Equal → SetDelayed
```

Lemma. This lemma is also needed to bring pairsets into play.

```
In[18]:= Map[not, SubstTest[and, implies[p1, p2],
            implies[p1, p3], implies[and[p2, p3], p4], not[implies[p1, p4]],
            {p1 → and[member[x, CL], member[pair[t, u], x], member[pair[t, v], x]],
             p2 → member[u, fix[x]], p3 → member[v, fix[x]],
             p4 → subclass[set[u, v], fix[x]]}]] // Reverse

Out[18]= or[not[member[x, CL]], not[member[pair[t, u], x]],
           not[member[pair[t, v], x]], subclass[set[u, v], fix[x]]] == True

In[19]:= (% /. {t → t_, u → u_, v → v_, x → x_}) /. Equal → SetDelayed
```

Theorem. If **x** is a complete lattice and if **t ≤ u** and **t ≤ v**, then **t ≤ u ∧ v**. (Note that explicit literals stating that **t**, **u** and **v** belong to **fix[x]** are not needed here.)

```
In[20]:= Map[not, SubstTest[and, implies[p1, p2], implies[p1, p3],
            implies[and[p1, p3], p4], implies[and[p1, p2, p4], p5], not[implies[p1, p5]],
            {p1 → and[member[x, CL], member[pair[t, u], x], member[pair[t, v], x]],
             p2 → member[t, fix[x]], p3 → PARTIALORDER[x],
             p4 -> subclass[set[u, v], image[x, set[t]]],
             p5 -> member[pair[t, APPLY[GLB[x], set[u, v]]], x]}]] // Reverse

Out[20]= or[member[pair[t, APPLY[GLB[x], set[u, v]]], x], not[member[x, CL]],
          not[member[pair[t, u], x]], not[member[pair[t, v], x]]] == True

In[21]:= or[member[pair[t_, APPLY[GLB[x_], set[u_, v_]]], x_], not[member[pair[t_, u_], x_]],
          not[member[pair[t_, v_], x_]], not[member[x_, CL]]] := True
```

Lemma. Here again, the **po** wrapper is used to automatically remove redundant sethood literals.

```
In[22]:= Map[implies[member[po[x], CL],
              or[#, member[pair[APPLY[LUB[po[x]], set[u, v]], w], po[x]]]] &,
            SubstTest[or, member[pair[w, APPLY[GLB[t], set[u, v]]], t],
             not[member[t, CL]], not[member[pair[w, u], t]],
             not[member[pair[w, v], t]], t → inverse[po[x]]]] // Reverse // MapNotNot

Out[22]= or[member[pair[APPLY[LUB[po[x]], set[u, v]], w], po[x]], not[member[pair[u, w], po[x]]],
          not[member[pair[v, w], po[x]]], not[member[po[x], CL]]] == True

In[23]:= (% /. {u → u_, v → v_, w → w_, x → x_}) /. Equal → SetDelayed
```

Removing the **po** wrapper yields the dual theorem:

Corollary. If **x** is a complete lattice and if **u** ≤ **w** and **v** ≤ **w**, then **u** ∨ **v** ≤ **w**.

```
In[24]:= SubstTest[implies, equal[x, po[t]],
            or[member[pair[APPLY[LUB[x], set[u, v]], w], x], not[member[pair[u, w], x]],
             not[member[pair[v, w], x]], not[member[x, CL]]], t → x] // Reverse // MapNotNot

Out[24]= or[member[pair[APPLY[LUB[x], set[u, v]], w], x], not[member[x, CL]],
          not[member[pair[u, w], x]], not[member[pair[v, w], x]]] == True

In[25]:= or[member[pair[APPLY[LUB[x_], set[u_, v_]], w_], x_], not[member[x_, CL]],
          not[member[pair[u_, w_], x_]], not[member[pair[v_, w_], x_]]] := True
```