strict monotonicity of GLB[x] and LUB[x] for complete lattices

Johan G. F. Belinfante
2009 March 19

```
In[1]:= SetDirectory["l:"]; << goedel.09mar16a; << tools.m

        :Package Title: goedel.09mar16a            2009 March 16 at 4:35 p.m.

        It is now:  2009 Mar 19 at 12:30

        Loading Simplification Rules

        TOOLS.M                                    Revised 2009 February 18

        weightlimit = 40
```

summary

In this notebook the monotonicity properties of **GLB[x]** and **LUB[x]** for a complete lattice are strengthened to equations.

derivation

Lemma.

```
In[2]:= (Map[subclass[#, GLB[po[t]]] &,
           Assoc[GLB[po[t]], composite[VERTSECT[po[t]], id[fix[po[t]]]],
            inverse[composite[VERTSECT[po[t]], id[fix[po[t]]]]]] // Reverse) /. t → setpart[x]

Out[2]= subclass[composite[id[fix[po[setpart[x]]]], inverse[VERTSECT[po[setpart[x]]]]],
          GLB[po[setpart[x]]]] == True

In[3]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[4]:= SubstTest[implies, subclass[u, v],
          subclass[composite[u, w, inverse[u]], composite[v, w, inverse[v]]],
          {u → composite[id[fix[po[setpart[x]]]], inverse[VERTSECT[po[setpart[x]]]]],
           v → GLB[po[setpart[x]]], w → S}] // Reverse

Out[4]= subclass[inverse[po[setpart[x]]],
          composite[GLB[po[setpart[x]]], S, inverse[GLB[po[setpart[x]]]]]] == True

In[5]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[6]:= Map[implies[member[po[x], CL], #] &,
         (SubstTest[implies, equal[w, po[setpart[t]]], subclass[inverse[w],
              composite[GLB[w], S, inverse[GLB[w]]]], t → w] // Reverse) /. w → inverse[po[x]]]

Out[6]= or[not[member[po[x], CL]],
          subclass[po[x], composite[LUB[po[x]], S, inverse[LUB[po[x]]]]]] == True

In[7]:= (% /. x → x_) /. Equal→ SetDelayed

In[8]:= Map[not, SubstTest[and, implies[p1, p2],
           implies[p1, p3], not[implies[p1, p4]], {p1 → member[po[x], CL],
            p2 → subclass[po[x], composite[LUB[po[x]], S, inverse[LUB[po[x]]]]],
            p3 -> subclass[composite[LUB[po[x]], S, inverse[LUB[po[x]]]], po[x]],
            p4 → equal[composite[LUB[po[x]], S, inverse[LUB[po[x]]]], po[x]]}]] // Reverse

Out[8]= or[equal[composite[LUB[po[x]], S, inverse[LUB[po[x]]]], po[x]],
          not[member[po[x], CL]]] == True

In[9]:= (% /. x → x_) /. Equal→ SetDelayed
```

Theorem. Strict monotonicity property of **LUB[x]** for a complete lattice **x**.

```
In[10]:= SubstTest[implies, equal[x, po[t]], or[equal[composite[LUB[x], S, inverse[LUB[x]]], x],
             not[member[x, CL]]], t → x] // MapNotNot // Reverse

Out[10]= or[equal[x, composite[LUB[x], S, inverse[LUB[x]]]], not[member[x, CL]]] == True

In[11]:= or[equal[x_, composite[LUB[x_], S, inverse[LUB[x_]]]], not[member[x_, CL]]] := True
```

Corollary. Strict monotonicity property of **GLB[x]** for a complete lattice **x**. (Derived using duality.)

```
In[12]:= Map[implies[member[x, CL], #] &, SubstTest[implies, member[t, CL],
            equal[t, composite[LUB[t], S, inverse[LUB[t]]]], t → inverse[x]]] // Reverse

Out[12]= or[equal[composite[GLB[x], S, inverse[GLB[x]]], inverse[x]], not[member[x, CL]]] == True

In[13]:= or[equal[composite[GLB[x_], S, inverse[GLB[x_]]], inverse[x_]],
           not[member[x_, CL]]] := True
```