

finite chains and complete lattices

Johan G. F. Belinfante
2005 October 17

```
In[1]:= SetDirectory["1:"]; << goedel74.17b; << tools.m

:Package Title: goedel74.17b          2005 October 17 at 4:30 p.m.

It is now: 2005 Oct 17 at 20:9

Loading Simplification Rules

TOOLS.M                      Revised 2005 October 17

weightlimit = 40
```

summary

Any nonempty finite chain is a complete lattice.

derivation

Lemma 1.

```
In[3]:= Map[not, (SubstTest[member, fix[z], domain[LEAST[z]], z → wo[y]) /.
  y → inverse[to[fin[x]]] // Reverse

Out[3]= equal[0, intersection[fix[to[fin[x]]], ub[to[fin[x]], fix[to[fin[x]]]]] ==
  equal[0, to[fin[x]]]

In[4]:= equal[0, intersection[fix[to[fin[x_]]], ub[to[fin[x_]], fix[to[fin[x_]]]]] :=
  equal[0, to[fin[x]]]
```

Lemma 2.

```
In[5]:= ((member[0, domain[GLB[z]]] // AssertTest) /. z → wo[y]) /. y → to[fin[x]]

Out[5]= member[0, domain[GLB[to[fin[x]]]]] == not[equal[0, to[fin[x]]]]

In[6]:= member[0, domain[GLB[to[fin[x_]]]]] := not[equal[0, to[fin[x]]]
```

Theorem.

```
In[7]:= SubstTest[implies, and[member[y, WO], member[0, domain[GLB[y]]]],
  member[y, CL], y → to[fin[x]]]
```

```
Out[7]= or[equal[0, to[fin[x]]], member[to[fin[x]], CL]] == True
```

```
In[8]:= or[equal[0, to[fin[x_]]], member[to[fin[x_]], CL]] := True
```

The wrappers can be removed.

```
In[10]:= SubstTest[implies, equal[x, to[fin[y]]], member[x, union[set[0], CL]], y → x]
```

```
Out[10]= or[equal[0, x], member[x, CL], not[member[x, FINITE]], not[TOTALORDER[x]]] == True
```

```
In[11]:= or[equal[0, x_], member[x_, CL], not[member[x_, FINITE]], not[TOTALORDER[x_]]] := True
```

Variable-free formulation.

```
In[12]:= Map[equal[V, #] &, SubstTest[class, x,
  implies[member[x, u], member[x, v]], {u → intersection[FINITE, TO],
  v → union[set[0], CL]}]] // Reverse
```

```
Out[12]= subclass[intersection[FINITE, TO], union[CL, set[0]]] == True
```

```
In[13]:= subclass[intersection[FINITE, TO], union[CL, set[0]]] := True
```