complete lattice fixed point theorem

Johan G. F. Belinfante
2006 January 7

```
In[1]:= SetDirectory["l:"]; << goedel77.07a; << tools.m

        :Package Title: goedel77.07a                   2006 January 7 at 1:40 p.m.

        It is now:  2006 Jan 8 at 5:39

        Loading Simplification Rules

        TOOLS.M                      Revised 2006 January 2

        weightlimit = 40
```

summary

The Knaster-Tarski fixed-point theorem for complete lattices is derived in this notebook through a series of lemmas. For most lemmas, the hypotheses are that **x** is a complete lattice order and **y** is a monotone mapping from the fixed point set of **x** to itself. It is shown that **z = APPLY[GLB[x],fix[composite[x,y]]]** is a fixed point of the function **y**.

```
In[2]:= "A. Tarski, A lattice-theoretical theorem and its
           applications, Pacific Journal of Mathematics 5:285-309 (1955)";

In[3]:= "B.Knaster, Un theoreme sur les fonctions
          d'ensembles. Ann. Soc. Polon. Math. 6:133-134. (1928)"

Out[3]= B.Knaster, Un theoreme sur les fonctions
          d'ensembles. Ann. Soc. Polon. Math. 6:133-134. (1928)
```

lemmas

Lemma 1. The relation **x** is on the domain of the function **y**.

```
In[4]:= Map[not, SubstTest[and, implies[p1, p3], implies[p2, p4],
          implies[and[p3, p4], p5], not[implies[and[p1, p2], p5]],
          {p1 -> member[x, CL], p2 -> member[y, map[fix[x], fix[x]]],
           p3 -> PARTIALORDER[x], p4 -> equal[domain[y], fix[x]],
           p5 -> subclass[x, cart[domain[y], domain[y]]]}]]

Out[4]= or[not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
          subclass[x, cart[domain[y], domain[y]]]] == True

In[5]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma 2. The element **z** is an element of **fix[x]**.

```
In[6]:= Map[not, SubstTest[and, implies[p1, p3],
          implies[and[p1, p2, p3], p4], not[implies[and[p1, p2], p4]],
          {p1 → member[x, CL], p2 → equal[z, APPLY[GLB[x], fix[composite[x, y]]]],
           p3 → subclass[fix[composite[x, y]], fix[x]], p4 → member[z, fix[x]]}]] /.
        z -> APPLY[GLB[x], fix[composite[x, y]]]

Out[6]= or[member[APPLY[GLB[x], fix[composite[x, y]]], fix[x]], not[member[x, CL]]] == True

In[7]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma 3. The element **z** belongs to the domain of the function **y**.

```
In[8]:= Map[not, SubstTest[and, implies[p1, p3], implies[p2, p4],
         implies[and[p3, p4], p5], not[implies[and[p1, p2], p5]],
         {p1 → member[x, CL], p2 → member[y, map[fix[x], fix[x]]], p3 →
           member[APPLY[GLB[x], fix[composite[x, y]]], fix[x]], p4 → equal[domain[y], fix[x]],
          p5 → member[APPLY[GLB[x], fix[composite[x, y]]], domain[y]]}]]

Out[8]= or[member[APPLY[GLB[x], fix[composite[x, y]]], domain[y]],
         not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]]] == True

In[9]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma 4. The element **APPLY[y, z]** belongs to **fix[x]**.

```
In[10]:= Map[not, SubstTest[and, implies[p2, p3], implies[and[p1, p2], p4], implies[p2, p5],
          implies[and[p3, p4], p6], implies[and[p5, p6], p7], not[implies[and[p1, p2], p7]],
          {p1 → member[x, CL], p2 → member[y, map[fix[x], fix[x]]], p3 → FUNCTION[y],
           p4 → member[APPLY[GLB[x], fix[composite[x, y]]], domain[y]],
           p5 → subclass[range[y], fix[x]],
           p6 → member[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]], range[y]],
           p7 → member[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]], fix[x]]}]]

Out[10]= or[member[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]], fix[x]],
          not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]]] == True

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

Lemma 5. The element **z** is a lower bound for **fix[composite[x, y]]**.

```
In[12]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3], not[implies[p1, p3]],
          {p1 → member[x, CL], p2 → subclass[fix[composite[x, y]], fix[x]], p3 → subclass[
             fix[composite[x, y]], image[x, set[APPLY[GLB[x], fix[composite[x, y]]]]]]}]]

Out[12]= or[not[member[x, CL]], subclass[fix[composite[x, y]],
           image[x, set[APPLY[GLB[x], fix[composite[x, y]]]]]]] == True

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

The next step is to show that **APPLY[y, z]** is also a lower bound for **fix[composite[x, y]]**. This is the most difficult part of the proof in that a temporary variable **t** needs to be introduced, and then eliminated. This is done in the next section.

use of a temporary variable

Lemma. If **t** belongs to **fix[composite[x, y]]**, then **APPLY[y,z]** is less than **APPLY[y,t]**.

```
In[14]:= Map[not, SubstTest[and, implies[and[p1, p2, p3], p4], implies[and[p1, p4, p6], p7],
           implies[and[p5, p7], p8], not[implies[and[p1, p2, p3, p5, p6], p8]],
           {p1 → FUNCTION[y], p2 → subclass[fix[composite[x, y]], image[x, set[z]]],
            p3 → member[t, fix[composite[x, y]]], p4 → member[pair[z, t], x],
            p5 → monotone[y, x, x], p6 → subclass[x, cart[domain[y], domain[y]]],
            p7 → member[pair[APPLY[y, z], APPLY[y, t]], composite[y, x, inverse[y]]],
            p8 -> member[pair[APPLY[y, z], APPLY[y, t]], x]}]]

Out[14]= or[member[pair[APPLY[y, z], APPLY[y, t]], x], not[FUNCTION[y]],
           not[member[t, fix[composite[x, y]]]], not[subclass[x, cart[domain[y], domain[y]]]],
           not[subclass[composite[y, x, inverse[y]], x]],
           not[subclass[fix[composite[x, y]], image[x, set[z]]]]] == True

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

Lemma. The element **APPLY[y, z]** is less than **t**.

```
In[16]:= Map[not, SubstTest[and, implies[p2, p3],
           implies[and[p1, p4, p5, p6, p7], q1], implies[and[p4, p7], q2],
           implies[and[p3, q1, q2], q3], not[implies[and[p1, p2, p4, p5, p6, p7], q3]],
           {p1 → subclass[composite[y, x, inverse[y]], x], p2 → PARTIALORDER[x],
            p3 → TRANSITIVE[x], p4 → FUNCTION[y], p5 → subclass[x, cart[domain[y], domain[y]]],
            p6 → subclass[fix[composite[x, y]], image[x, set[z]]],
            p7 → member[t, fix[composite[x, y]]],
            p9 → member[pair[APPLY[y, z], APPLY[y, t]], composite[y, x, inverse[y]]],
            q1 → member[pair[APPLY[y, z], APPLY[y, t]], x],
            q2 → member[pair[APPLY[y, t], t], x], q3 → member[pair[APPLY[y, z], t], x]}]]

Out[16]= or[member[pair[APPLY[y, z], t], x],
           not[FUNCTION[y]], not[member[t, fix[composite[x, y]]]],
           not[PARTIALORDER[x]], not[subclass[x, cart[domain[y], domain[y]]]],
           not[subclass[composite[y, x, inverse[y]], x]],
           not[subclass[fix[composite[x, y]], image[x, set[z]]]]] == True

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

Introduce wrappers to simplify the statement.

```
In[19]:= SubstTest[or, member[pair[APPLY[y, z], t], x],
          not[FUNCTION[y]], not[member[t, fix[composite[x, y]]]],
          not[PARTIALORDER[x]], not[subclass[x, cart[domain[y], domain[y]]]],
          not[subclass[composite[y, x, inverse[y]], x]],
          not[subclass[fix[composite[x, y]], image[x, set[z]]]], {x → po[u], y → funpart[v]}]

Out[19]= or[member[pair[APPLY[funpart[v], z], t], po[u]],
           not[member[t, fix[composite[po[u], funpart[v]]]]],
           not[subclass[composite[funpart[v], po[u], inverse[funpart[v]]], po[u]]],
           not[subclass[fix[composite[po[u], funpart[v]]], image[po[u], set[z]]]],
           not[subclass[fix[po[u]], domain[funpart[v]]]]] == True

In[20]:= (% /. {t → t_, u → u_, v → v_, z → z_}) /. Equal → SetDelayed
```

The variable **t** is eliminated. This takes a couple of minutes.

```
In[21]:= Map[equal[V, #] &, SubstTest[class, t, or[member[t, r], not[member[t, s]], equal[V, w]],
            {r → image[po[u], set[APPLY[funpart[v], z]]], s -> fix[composite[po[u], funpart[v]]],
             w -> union[complement[image[V, intersection[domain[funpart[v]], set[z]]]],
               image[V, fix[composite[po[u], inverse[funpart[v]],
                  complement[inverse[po[u]]], funpart[v]]]],
               image[V, intersection[complement[domain[funpart[v]]], fix[po[u]]]],
               image[V, intersection[complement[image[po[u], set[z]]],
                 fix[composite[po[u], funpart[v]]]]]]}]] // Reverse

Out[21]= or[not[member[z, domain[funpart[v]]]],
           not[subclass[composite[funpart[v], po[u], inverse[funpart[v]]], po[u]]],
           not[subclass[fix[composite[po[u], funpart[v]]], image[po[u], set[z]]]],
           not[subclass[fix[po[u]], domain[funpart[v]]]], subclass[
            fix[composite[po[u], funpart[v]]], image[po[u], set[APPLY[funpart[v], z]]]]] == True

In[22]:= (% /. {u → u_, v → v_, z → z_}) /. Equal → SetDelayed
```

The **po** and **funpart** wrappers are now removed again.

```
In[23]:= SubstTest[implies, and[equal[x, po[u]], equal[y, funpart[v]]],
          or[not[member[z, domain[y]]], not[subclass[composite[y, x, inverse[y]], x]],
           not[subclass[fix[composite[x, y]], image[x, set[z]]]],
           not[subclass[fix[x], domain[y]]],
           subclass[fix[composite[x, y]], image[x, set[APPLY[y, z]]]]], {u -> x, v → y}]

Out[23]= or[not[FUNCTION[y]], not[member[z, domain[y]]], not[PARTIALORDER[x]],
           not[subclass[composite[y, x, inverse[y]], x]], not[subclass[fix[x], domain[y]]],
           not[subclass[fix[composite[x, y]], image[x, set[z]]]],
           subclass[fix[composite[x, y]], image[x, set[APPLY[y, z]]]]] == True

In[24]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

final steps

Lemma 6. The element **APPLY[y, z]** is a lower bound for **fix[composite[x, y]]**.

```
In[25]:= Map[not, SubstTest[and, implies[p1, p4], implies[p2, p5],
           implies[p2, p6a], implies[p6a, p6b], implies[and[p1, p2], p7], implies[p1, p8],
           implies[and[p3, p4, p5, p6b, p7, p8], p9], not[implies[and[p1, p2, p3], p9]],
           {p1 → member[x, CL], p2 → member[y, map[fix[x], fix[x]]],
            p3 → subclass[composite[y, x, inverse[y]], x], p4 → PARTIALORDER[x],
            p5 → FUNCTION[y], p6a → equal[fix[x], domain[y]], p6b -> subclass[fix[x], domain[y]],
            p7 → member[APPLY[GLB[x], fix[composite[x, y]]], domain[y]], p8 →
             subclass[fix[composite[x, y]], image[x, set[APPLY[GLB[x], fix[composite[x, y]]]]]],
            p9 → subclass[fix[composite[x, y]],
              image[x, set[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]]]]]}]]

Out[25]= or[not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
           not[subclass[composite[y, x, inverse[y]], x]], subclass[fix[composite[x, y]],
            image[x, set[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]]]]]] == True

In[26]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma 7. The lower bound **APPLY[y, z]** lies below the greatest lower bound **z**.

```
In[27]:= Map[not, SubstTest[and, implies[and[p1, p2], p4], implies[and[p1, p2, p3], p5],
           implies[and[p1, p4, p5], p6], not[implies[and[p1, p2, p3], p6]], {p1 → member[x, CL],
            p2 → member[y, map[fix[x], fix[x]]], p3 → subclass[composite[y, x, inverse[y]], x],
            p4 → member[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]], fix[x]],
            p5 → subclass[fix[composite[x, y]],
              image[x, set[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]]]]],
            p6 → member[pair[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]],
               APPLY[GLB[x], fix[composite[x, y]]]], x]}]]

Out[27]= or[member[pair[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]],
             APPLY[GLB[x], fix[composite[x, y]]]], x],
           not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
           not[subclass[composite[y, x, inverse[y]], x]]] == True

In[28]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma 8. The element **z** belongs to **fix[composite[x, y]]**.

```
In[29]:= Map[not, SubstTest[and, implies[and[p1, p2, p3], p5], implies[and[p1, p2], p4],
           implies[p2, p6], implies[and[p1, p2], p7], implies[and[p4, p5, p6, p7], p8],
           not[implies[and[p1, p2, p3], p8]], {p1 → member[x, CL],
            p2 → member[y, map[fix[x], fix[x]]], p3 → subclass[composite[y, x, inverse[y]], x],
            p4 → member[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]], fix[x]],
            p5 → member[pair[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]],
               APPLY[GLB[x], fix[composite[x, y]]]], x],
            p6 → FUNCTION[y], p7 → member[APPLY[GLB[x], fix[composite[x, y]]], domain[y]],
            p8 -> member[APPLY[GLB[x], fix[composite[x, y]]], fix[composite[x, y]]]}]]

Out[29]= or[member[APPLY[GLB[x], fix[composite[x, y]]], fix[composite[x, y]]],
           not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
           not[subclass[composite[y, x, inverse[y]], x]]] == True

In[30]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma 9.

```
In[31]:= SubstTest[implies, and[FUNCTION[y], member[z, w], subclass[w, domain[y]]],
          member[APPLY[y, z], image[y, w]],
          w → fix[composite[x, y]]]

Out[31]= or[member[APPLY[y, z], image[y, fix[composite[x, y]]]],
           not[FUNCTION[y]], not[member[z, fix[composite[x, y]]]]] == True

In[32]:= (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Lemma 10. The element **APPLY[y, z]** belongs to **fix[composite[x y]]**.

```
In[33]:= Map[not, SubstTest[and, implies[p2, p4], implies[and[p1, p2, p3], p5],
           implies[and[p1, p2, p3], p6], implies[and[p4, p6], p7],
           implies[and[p5, p7], p8], not[implies[and[p1, p2, p3], p8]],
           {p1 → member[x, CL], p2 → member[y, map[fix[x], fix[x]]],
            p3 → subclass[composite[y, x, inverse[y]], x], p4 → FUNCTION[y],
            p5 → subclass[image[y, fix[composite[x, y]]], fix[composite[x, y]]],
            p6 → member[APPLY[GLB[x], fix[composite[x, y]]], fix[composite[x, y]]], p7 → member[
              APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]], image[y, fix[composite[x, y]]]],
            p8 → member[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]], fix[composite[x, y]]]}]]

Out[33]= or[member[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]], fix[composite[x, y]]],
           not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
           not[subclass[composite[y, x, inverse[y]], x]]] == True

In[34]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma 11. The element **z** is below **APPLY[y, z]**.

```
In[35]:= Map[not, SubstTest[and, implies[p1, p4], implies[and[p1, p2, p3], p5],
           implies[and[p4, p5], p6], not[implies[and[p1, p2, p3], p6]],
           {p1 → member[x, CL], p2 → member[y, map[fix[x], fix[x]]],
            p3 → subclass[composite[y, x, inverse[y]], x], p4 →
             subclass[fix[composite[x, y]], image[x, set[APPLY[GLB[x], fix[composite[x, y]]]]]],
            p5 → member[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]], fix[composite[x, y]]],
            p6 → member[pair[APPLY[GLB[x], fix[composite[x, y]]],
               APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]]], x]}]]

Out[35]= or[member[pair[APPLY[GLB[x], fix[composite[x, y]]],
             APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]]], x],
           not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
           not[subclass[composite[y, x, inverse[y]], x]]] == True

In[36]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma 12. The element **APPLY[y, z]** is below **z**.

```
In[37]:= Map[not, SubstTest[and, implies[p1, p4],
           implies[and[p1, p2, p3], p5], implies[and[p1, p2, p3], p6],
           implies[and[p4, p5, p6], p7], not[implies[and[p1, p2, p3], p7]],
           {p1 → member[x, CL], p2 → member[y, map[fix[x], fix[x]]],
            p3 → subclass[composite[y, x, inverse[y]], x], p4 → PARTIALORDER[x],
            p5 → member[pair[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]],
               APPLY[GLB[x], fix[composite[x, y]]]], x], p6 → member[pair[APPLY[GLB[x],
                fix[composite[x, y]]], APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]]], x],
            p7 → equal[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]],
              APPLY[GLB[x], fix[composite[x, y]]]]}]]

Out[37]= or[equal[APPLY[y, APPLY[GLB[x], fix[composite[x, y]]]],
            APPLY[GLB[x], fix[composite[x, y]]]],
           not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
           not[subclass[composite[y, x, inverse[y]], x]]] == True

In[38]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

The Knaster-Tarski fixed point theorem.

```
In[39]:= Map[not, SubstTest[and, implies[p2, p4], implies[p1, p5], implies[and[p1, p2, p3], p6],
           implies[and[p4, p5, p6], p7], not[implies[and[p1, p2, p3], p7]],
           {p1 → member[x, CL], p2 → member[y, map[fix[x], fix[x]]],
            p3 → subclass[composite[y, x, inverse[y]], x], p4 → FUNCTION[y],
            p5 → member[APPLY[GLB[x], fix[composite[x, y]]], fix[x]], p6 → equal[APPLY[y,
               APPLY[GLB[x], fix[composite[x, y]]]], APPLY[GLB[x], fix[composite[x, y]]]],
            p7 → member[APPLY[GLB[x], fix[composite[x, y]]], fix[y]]}]]

Out[39]= or[member[APPLY[GLB[x], fix[composite[x, y]]], fix[y]],
           not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
           not[subclass[composite[y, x, inverse[y]], x]]] == True
```

```
In[40]:= or[member[APPLY[GLB[x_], fix[composite[x_, y_]]], fix[y_]],
           not[member[x_, CL]], not[member[y_, map[fix[x_], fix[x_]]]],
           not[subclass[composite[y_, x_, inverse[y_]], x_]]] := True
```

corollaries

The dual theorem:

```
In[41]:= Map[implies[member[x, CL], #] &,
          SubstTest[or, member[APPLY[GLB[z], fix[composite[z, y]]], fix[y]],
           not[member[z, CL]], not[member[y, map[fix[z], fix[z]]]],
           not[subclass[composite[y, z, inverse[y]], z]], z → inverse[x]]]

Out[41]= or[member[APPLY[LUB[x], fix[composite[inverse[x], y]]], fix[y]],
           not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
           not[subclass[composite[y, x, inverse[y]], x]]] == True

In[42]:= or[member[APPLY[LUB[x_], fix[composite[inverse[x_], y]]], fix[y_]],
           not[member[x_, CL]], not[member[y_, map[fix[x_], fix[x_]]]],
           not[subclass[composite[y_, x_, inverse[y_]], x_]]] := True
```

Corollary.

```
In[43]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], not[implies[p1, p3]],
           {p1 → and[member[x, CL], member[y, map[fix[x], fix[x]]], monotone[y, x, x]],
            p2 -> member[APPLY[GLB[x], fix[composite[x, y]]], fix[y]], p3 → not[empty[fix[y]]]}]]

Out[43]= or[not[equal[0, fix[y]]], not[member[x, CL]], not[member[y, map[fix[x], fix[x]]]],
           not[subclass[composite[y, x, inverse[y]], x]]] == True

In[44]:= or[not[equal[0, fix[y_]]], not[member[x_, CL]], not[member[y_, map[fix[x_], fix[x_]]]],
           not[subclass[composite[y_, x_, inverse[y_]], x_]]] := True
```

The variable **y** can be eliminated:

```
In[50]:= Map[equal[V, #] &,
            SubstTest[class, y, implies[and[member[x, u], member[y, v]], not[member[y, w]]],
             {u → CL, v -> intersection[map[fix[x], fix[x]], cliques[
                complement[cross[x, complement[x]]]]], w → P[Di]}]] // Reverse // MapNotNot

Out[50]= or[equal[0, intersection[cliques[complement[cross[x, complement[x]]]],
             map[fix[x], fix[x]], P[Di]]], not[member[x, CL]]] == True

In[51]:= or[equal[0, intersection[cliques[complement[cross[x_, complement[x_]]]],
             map[fix[x_], fix[x_]], P[Di]]], not[member[x_, CL]]] := True
```