transitive law for final segments

Johan G. F. Belinfante
2010 July 11

```
In[1]:= SetDirectory["l:"]; << goedel.10jul09a; << tools.m

        :Package Title: goedel.10jul09a              2010 July 8 at 1:10 p.m.

        It is now:  2010 Jul 11 at 15:9

        Loading Simplification Rules

        TOOLS.M                                    Revised 2010 February 26

        weightlimit = 40
```

summary

A theorem about transitivity of final segments is stated in this book:

```
In[3]:= "Egbert Harzheim, Ordered Sets, Advances in Mathematics, volume 7, Springer
         Science+Business Media, Inc., 2005.  ISBN 0387-24219-8.  QA171.48 .H37";
```

A class **y** is a **final segment** of a relation **x** if **image[x, y] = y**. That is, the class **y** is both invariant and subvariant under **x**. The class of final segments of a relation **x** is **fix[IMAGE[x]]**.

A variable-free formulation of this theorem is derived in this notebook.

introductory remarks

The function that takes a set **x** to the set **fix[IMAGE[x]]** of final segments is:

```
In[2]:= VERTSECT[reify[x, fix[IMAGE[x]]]]
Out[2]= VERTSECT[fix[composite[inverse[SECOND], IMG]]]
```

The following relation is also needed:

```
In[4]:= class[pair[x, v], member[v, fix[IMAGE[x]]]]
Out[4]= fix[composite[inverse[SECOND], IMG]]
```

A simple membership rule for this relation can be formulated in terms of the **setpart** wrapper:

```
In[5]:= member[pair[setpart[x], setpart[v]], fix[composite[inverse[SECOND], IMG]]]

Out[5]= equal[image[setpart[x], setpart[v]], setpart[v]]
```

derivation

In this section, a transitive law for final segments is derived. The theorem is concerned with a final segment **u** of a restriction of a relation **x** to a final segment **v** of **x**.

Lemma.

```
In[6]:= or[equal[u, image[x, u]], not[equal[u, intersection[v, image[x, u]]]],
          not[subclass[image[x, u], v]]] // NotNotTest

Out[6]= or[equal[u, image[x, u]],
          not[equal[u, intersection[v, image[x, u]]]], not[subclass[image[x, u], v]]] == True

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

Lemma.

```
In[8]:= (Map[not, SubstTest[and, implies[p1, p3],
             implies[and[p0, p1], p4], implies[and[p0, p3], p5], implies[and[p4, p5], p8],
             not[implies[and[p0, p1], p8]], {p0 → equal[t, intersection[u, v]],
              p1 -> equal[u, intersection[v, image[x, intersection[u, v]]]], p3 → subclass[u, v],
              p4 → equal[u, intersection[v, image[x, t]]], p5 → equal[t, u],
              p8 → equal[u, intersection[v, image[x, u]]]}]] /. t → intersection[u, v]) // Reverse

Out[8]= or[equal[u, intersection[v, image[x, u]]],
          not[equal[u, intersection[v, image[x, intersection[u, v]]]]]] == True

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

The following is the transitive law for final segments, as stated by Harzheim, page 32, Theorem 8.10.

Theorem. If **v** is a final segment of **x**, and if **u** is a final segment of the two-sided restriction of **x** to **v**, then **u** is a final segment of **x**.

```
In[10]:= Map[not, SubstTest[and, implies[p1, p8], implies[p1, p3], implies[p3, p6],
            implies[and[p2, p6], p7], implies[and[p7, p8], p9], not[implies[and[p1, p2], p9]],
            {p1 -> equal[u, intersection[v, image[x, intersection[u, v]]]],
             p2 -> equal[v, image[x, v]], p3 → subclass[u, v],
             p6 → subclass[image[x, u], image[x, v]], p7 → subclass[image[x, u], v],
             p8 → equal[u, intersection[v, image[x, u]]], p9 -> equal[u, image[x, u]]}]] // Reverse

Out[10]= or[equal[u, image[x, u]], not[equal[u, intersection[v, image[x, intersection[u, v]]]]],
           not[equal[v, image[x, v]]]] == True
```

```
In[11]:= or[equal[u_, image[x_, u_]],
           not[equal[u_, intersection[v_, image[x_, intersection[u_, v_]]]]],
           not[equal[v_, image[x_, v_]]]] := True
```

eliminating variables

Temporary Lemma.

```
In[12]:= member[u, fix[composite[IMAGE[composite[id[v], x]], IMAGE[id[v]]]]] // AssertTest

Out[12]= member[u, fix[composite[IMAGE[composite[id[v], x]], IMAGE[id[v]]]]] ==
          and[equal[u, intersection[v, image[x, intersection[u, v]]]], member[u, V]]

In[13]:= member[u_, fix[composite[IMAGE[composite[id[v_], x_]], IMAGE[id[v_]]]]] :=
          and[equal[u, intersection[v, image[x, intersection[u, v]]]], member[u, V]]
```

Lemma. Eliminating **u**.

```
In[14]:= Map[equal[V, #] &,
          SubstTest[class, u, implies[and[member[u, s], equal[v, image[x, v]]], member[u, t]],
           {s -> fix[IMAGE[restrict[x, v, v]]], t -> fix[IMAGE[x]]}]]

Out[14]= or[not[equal[v, image[x, v]]], subclass[
            fix[composite[IMAGE[composite[id[v], x]], IMAGE[id[v]]]], fix[IMAGE[x]]]] == True

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

It helps to introduce an additional relation variable **y** and reformulate the result found so far as follows.

```
In[16]:= implies[and[equal[v, image[x, v]], equal[y, restrict[x, v, v]]],
          subclass[fix[IMAGE[y]], fix[IMAGE[x]]]]

Out[16]= True
```

The following key observation requires introducing **setpart**.

```
In[17]:= member[pair[pair[setpart[x], setpart[v]], y],
          composite[CAP, cross[Id, composite[CART, DUP]]]]

Out[17]= equal[y, composite[id[setpart[v]], setpart[x], id[setpart[v]]]]
```

Lemma. Introducing **setpart**.

```
In[18]:= SubstTest[implies, and[equal[t, image[u, t]], equal[y, restrict[u, t, t]]],
           subclass[fix[IMAGE[y]], fix[IMAGE[u]]], {u → setpart[x], t → setpart[v]}] // Reverse

Out[18]= or[not[equal[y, composite[id[setpart[v]], setpart[x], id[setpart[v]]]]],
           not[equal[image[setpart[x], setpart[v]], setpart[v]]],
           subclass[fix[IMAGE[y]], fix[IMAGE[setpart[x]]]]] == True

In[19]:= (% /. {v → v_, x → x_, y → y_}) /. Equal → SetDelayed
```

Observation.

```
In[20]:= member[pair[setpart[y], setpart[x]], composite[inverse[s], S, s]] /.
          s -> VERTSECT[fix[composite[inverse[SECOND], IMG]]]

Out[20]= subclass[fix[IMAGE[setpart[y]]], fix[IMAGE[setpart[x]]]]
```

One can now eliminate all variables from the transitive law for final segments.

Theorem. Variable-free statement of the transitive laws for final segments.

```
In[21]:= Map[empty[range[#]] &, Map[composite[complement[#], id[cart[V, V]]] &, SubstTest[class,
            pair[pair[x, v], y], implies[and[member[pair[setpart[x], setpart[v]], z],
              member[pair[pair[setpart[x], setpart[v]], setpart[y]], w]],
             member[pair[setpart[y], setpart[x]], composite[inverse[s], S, s]]],
            {s -> VERTSECT[fix[composite[inverse[SECOND], IMG]]],
             w -> composite[CAP, cross[Id, composite[CART, DUP]]],
             z -> fix[composite[inverse[SECOND], IMG]]}]]]

Out[21]= subclass[composite[fix[composite[inverse[SECOND], IMG]], CAP,
            id[composite[CART, DUP, fix[composite[inverse[SECOND], IMG]]]], inverse[FIRST]],
           fix[composite[inverse[SECOND], IMG]]] == True

In[22]:= subclass[composite[fix[composite[inverse[SECOND], IMG]], CAP,
            id[composite[CART, DUP, fix[composite[inverse[SECOND], IMG]]]], inverse[FIRST]],
           fix[composite[inverse[SECOND], IMG]]] := True
```