least elements for a wellorder

Johan G. F. Belinfante
2009 October 21

```
In[1]:= SetDirectory["l:"]; << goedel.09oct19a;<< tools.m

        :Package Title: goedel.09oct19a            2009 October 19 at 3:40 p.m.

        It is now:  2009 Oct 21 at 11:4

        Loading Simplification Rules

        TOOLS.M                                    Revised 2009 September 15

        weightlimit = 40
```

summary

The definition of a well-order only implies the existence of least elements for non-empty subsets of its fixed point class. This result does not suffice for ordinal number theory because the class of ordinals is not a set. In general, when a well-order is not a set, one may wish to consider non-empty subclasses which need not be sets. In this notebook such a general result is derived by adding a thin-ness hypothesis. It is shown that if a well-order has a thin inverse, then every non-empty subclass of its fixed-point class has a least member.

derivation

If the fixed point class of a total order has a maximal element, then that element is the greatest element. If a well-founded relation has a thin inverse, then every nonepty class has a minimal element. Combining these results yields the following theorem.

Theorem.

```
In[9]:= SubstTest[implies, and[thin[inverse[wf[t]]], empty[minimal[wf[t], y]]],
          empty[y], {t → intersection[wo[x], Di], y → fix[wo[x]]}] // Reverse

Out[9]= or[equal[0, wo[x]], not[equal[0, funpart[inverse[wo[x]]]]],
          not[equal[V, domain[VERTSECT[inverse[wo[x]]]]]]] == True

In[10]:= or[equal[0, wo[x_]], not[equal[0, funpart[inverse[wo[x_]]]]],
           not[equal[V, domain[VERTSECT[inverse[wo[x_]]]]]]] := True
```

Restatement.

```
In[12]:= implies[thin[inverse[wo[x]]], or[empty[wo[x]], not[empty[least[wo[x], fix[wo[x]]]]]]]

Out[12]= True
```

Corollary. (Wrapper-free restatement.)

```
In[16]:= SubstTest[implies, equal[x, wo[t]], implies[thin[inverse[x]],
            or[empty[x], not[empty[least[x, fix[x]]]]]], t → x] // Reverse

Out[16]= or[equal[0, x], not[equal[0, intersection[fix[x], lb[x, fix[x]]]]],
           not[equal[V, domain[VERTSECT[inverse[x]]]]], not[WELLORDER[x]]] == True

In[17]:= or[equal[0, x_], not[equal[0, intersection[fix[x_], lb[x_, fix[x_]]]]],
           not[equal[V, domain[VERTSECT[inverse[x_]]]]], not[WELLORDER[x_]]] := True
```

Lemma. (Specialization to a restriction of a well-order.)

```
In[19]:= SubstTest[or, equal[0, t], not[equal[0, intersection[fix[t], lb[t, fix[t]]]]],
           not[equal[V, domain[VERTSECT[inverse[t]]]]],
           not[WELLORDER[t]], t → restrict[wo[x], y, y]] // Reverse

Out[19]= or[equal[0, intersection[y, image[wo[x], y]]],
           not[equal[0, intersection[y, fix[wo[x]], lb[wo[x], intersection[y, fix[wo[x]]]]]]],
           not[subclass[y, domain[VERTSECT[composite[id[y], inverse[wo[x]]]]]]]] == True

In[20]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma.

```
In[24]:= SubstTest[implies, and[subclass[t, x], thin[x]],
           thin[t], t → restrict[x, y, z]] // Reverse

Out[24]= or[not[equal[V, domain[VERTSECT[x]]]],
           subclass[y, domain[VERTSECT[composite[id[z], x]]]]] == True

In[25]:= or[not[equal[V, domain[VERTSECT[x_]]]],
           subclass[y_, domain[VERTSECT[composite[id[z_], x_]]]]] := True
```

Lemma.

```
In[32]:= SubstTest[implies, and[equal[u, y], not[empty[least[z, u]]]],
           not[empty[least[z, y]]], u → intersection[x, y]] // Reverse

Out[32]= or[equal[0, intersection[x, y, lb[z, intersection[x, y]]]],
           not[equal[0, intersection[y, lb[z, y]]]], not[subclass[y, x]]] == True

In[33]:= or[equal[0, intersection[x_, y_, lb[z_, intersection[x_, y_]]]],
           not[equal[0, intersection[y_, lb[z_, y_]]]], not[subclass[y_, x_]]] := True
```

Main Theorem. Existence of least elements for wellorders whose inverse is thin.

```
In[35]:= Map[not, SubstTest[and, implies[p1, p4], implies[and[p2, p3], p5],
            implies[and[p4, p5], p6], implies[and[p2, p6], p7], not[implies[and[p1, p2, p3], p7]],
            {p1 → thin[inverse[wo[x]]], p2 → subclass[y, fix[wo[x]]], p3 → not[empty[y]],
             p4 -> subclass[y, domain[VERTSECT[composite[id[y], inverse[wo[x]]]]]],
             p5 → not[disjoint[y, image[wo[x], y]]], p6 → not[
               equal[0, intersection[y, fix[wo[x]], lb[wo[x], intersection[y, fix[wo[x]]]]]]],
             p7 → not[empty[least[wo[x], y]]]}]] // Reverse

Out[35]= or[equal[0, y], not[equal[0, intersection[y, lb[wo[x], y]]]],
           not[equal[V, domain[VERTSECT[inverse[wo[x]]]]]], not[subclass[y, fix[wo[x]]]]] == True

In[36]:= or[equal[0, y_], not[equal[0, intersection[y_, lb[wo[x_], y_]]]],
           not[equal[V, domain[VERTSECT[inverse[wo[x_]]]]]],
           not[subclass[y_, fix[wo[x_]]]]] := True
```

Corollary. (Wrapper-free restatement.)

```
In[37]:= SubstTest[implies, equal[x, wo[t]],
           or[equal[0, y], not[equal[0, intersection[y, lb[x, y]]]],
            not[equal[V, domain[VERTSECT[inverse[x]]]]],
            not[subclass[y, fix[x]]]], t → x] // Reverse

Out[37]= or[equal[0, y], not[equal[0, intersection[y, lb[x, y]]]],
           not[equal[V, domain[VERTSECT[inverse[x]]]]],
           not[subclass[y, fix[x]]], not[WELLORDER[x]]] == True

In[38]:= or[equal[0, y_], not[equal[0, intersection[y_, lb[x_, y_]]]],
           not[equal[V, domain[VERTSECT[inverse[x_]]]]],
           not[subclass[y_, fix[x_]]], not[WELLORDER[x_]]] := True
```

Restatement. If **x** is a well-order whose inverse is thin, then every non-empty subclass of **fix[x]** has a least member.

```
In[39]:= implies[and[WELLORDER[x], thin[inverse[x]], subclass[y, fix[x]], not[empty[y]]],
          not[empty[least[x, y]]]]

Out[39]= True
```

examples

Example:

```
In[41]:= SubstTest[implies,
           and[WELLORDER[x], thin[inverse[x]], subclass[y, fix[x]], not[empty[y]]],
           not[empty[least[x, y]]], x → restrict[S, OMEGA, OMEGA]] // Reverse

Out[41]= or[and[member[A[y], OMEGA], member[A[y], y]],
           equal[0, y], not[subclass[y, OMEGA]]] == True
```

This result is known to the **GOEDEL** program. One need only apply a double negation.

```
In[42]:= % // MapNotNot

Out[42]= True
```

A more general result:

```
In[46]:= SubstTest[implies,
           and[WELLORDER[w], thin[inverse[w]], subclass[x, fix[w]], not[empty[x]]],
           not[empty[least[w, x]]], w → restrict[S, ord[y], ord[y]]] // Reverse // MapNotNot

Out[46]= or[equal[0, x], member[A[x], x], not[subclass[x, ord[y]]]] == True

In[47]:= or[equal[0, x_], member[A[x_], x_], not[subclass[x_, ord[y_]]]] := True
```

A similar result:

```
In[48]:= SubstTest[implies,
           and[WELLORDER[w], thin[inverse[w]], subclass[x, fix[w]], not[empty[x]]],
           not[empty[least[w, x]]], w → restrict[S, nat[y], nat[y]]] // Reverse // MapNotNot

Out[48]= or[equal[0, x], member[A[x], x], not[subclass[x, nat[y]]]] == True

In[49]:= or[equal[0, x_], member[A[x_], x_], not[subclass[x_, nat[y_]]]] := True
```