the least element is the only minimal element

Johan G. F. Belinfante
2011 September 12

```
In[1]:= SetDirectory["l:"]; << goedel.11sep09a

        :Package Title: goedel.11sep09a            2011 September 9 at 9:00 a.m.

        Loading takes about twelve minutes, half that time due to builtin pauses.

        It is now:  2011 Sep 12 at 10:49

        Loading Simplification Rules

        TOOLS.M is now incorporated in the GOEDEL program as of 2010 September 3

        weightlimit = 40

        Loading completed.

        It is now:  2011 Sep 12 at 11:1
```

summary

If **A[x]** belongs to **x**, then **A[x]** is the only minimal element of **x**.

derivation

The derivation in this section is patterned after a dual result obtained in the notebook **max-u.nb** dated 2007 November 3.

Observation.

```
In[2]:= or[equal[y, A[x]], member[y, image[PS, x]],
          not[member[y, V]], not[member[A[x], x]], not[subclass[A[x], y]]]

Out[2]= True
```

The following result is obtained by eliminating the variable **y** using a combination of **reify** and **case**.

Lemma.

```
In[3]:= Map[equal[V, domain[#]] &,
         SubstTest[reify, y, case[or[equal[y, A[x]], member[y, image[t, x]],
            not[member[y, V]], not[member[A[x], x]], not[subclass[A[x], y]]]], t -> PS]]

Out[3]= or[not[member[A[x], x]],
          subclass[image[S, set[A[x]]], union[image[PS, x], set[A[x]]]]] == True

In[4]:= (% /. x_ → x_) /. Equal → SetDelayed
```

Observation. The following fact is the dual counterpart of the inclusion **x ⊂ P[U[x]]** that was used in November 2007.

```
In[5]:= subclass[x, image[S, set[A[x]]]]

Out[5]= True
```

Theorem.

```
In[6]:= Map[not, SubstTest[and, implies[p1, p2],
           implies[p2, p3], not[implies[p1, p3]], {p1 → member[A[x], x],
            p2 -> subclass[image[S, set[A[x]]], union[image[PS, x], set[A[x]]]],
            p3 -> subclass[x, union[image[PS, x], set[A[x]]]]}]] // Reverse

Out[6]= or[not[member[A[x], x]], subclass[x, union[image[PS, x], set[A[x]]]]] == True

In[7]:= or[not[member[A[x_], x_]], subclass[x_, union[image[PS, x_], set[A[x_]]]]] := True
```

One can eliminate the variable **x** using a combination of **reify** and **case**.

Lemma. A temporary simplification rule.

```
In[8]:= Map[equal[V, domain[#]] &, SubstTest[reify, x,
          case[or[not[member[A[x], x]], subclass[x, union[image[t, x], set[A[x]]]]], t → PS]]

Out[8]= equal[0, intersection[fix[composite[E, BIGCAP]],
           fix[composite[inverse[BIGCAP], Di, MINIMAL[S]]]]] == True

In[9]:= intersection[fix[composite[E, BIGCAP]],
          fix[composite[inverse[BIGCAP], Di, MINIMAL[S]]]] := 0
```

Theorem. Simplification rule.

```
In[10]:= fix[composite[inverse[BIGCAP], MINIMAL[S]]] // Normality

Out[10]= fix[composite[inverse[BIGCAP], MINIMAL[S]]] == fix[composite[E, BIGCAP]]

In[11]:= fix[composite[inverse[BIGCAP], MINIMAL[S]]] := fix[composite[E, BIGCAP]]
```

Lemma. A temporary rewrite rule.

```
In[12]:= Map[complement[complement[#]] &,
          SubstTest[fix, union[u, v], {u -> composite[inverse[BIGCAP], Di, MINIMAL[S]],
            v -> composite[inverse[BIGCAP], MINIMAL[S]]}]]

Out[12]= union[fix[composite[E, BIGCAP]], fix[composite[inverse[BIGCAP], Di, MINIMAL[S]]]] ==
          complement[subvar[PS]]

In[13]:= % /. Equal→ SetDelayed
```

Since this is a disjoint union, one can solve for the fixed point class that involves **Di**.

Theorem.

```
In[14]:= SubstTest[implies, and[disjoint[u, v], equal[union[u, v], w]], equal[v, dif[w, u]],
           {u -> fix[composite[E, BIGCAP]], v -> fix[composite[inverse[BIGCAP], Di, MINIMAL[S]]],
            w -> complement[subvar[PS]]}] // Reverse

Out[14]= equal[fix[composite[inverse[BIGCAP], Di, MINIMAL[S]]],
           intersection[complement[fix[composite[E, BIGCAP]]], complement[subvar[PS]]]] == True

In[15]:= fix[composite[inverse[BIGCAP], Di, MINIMAL[S]]] :=
          intersection[complement[fix[composite[E, BIGCAP]]], complement[subvar[PS]]]
```

---

minimal ⇒ least

If a total order has a minimal element, then that element is the least element. In this section, this result is applied to the case of the subset relation restricted to a chain of sets.

Theorem.

```
In[16]:= SubstTest[implies, and[TOTALORDER[t], subclass[x, fix[t]]],
           subclass[maximal[t, x], greatest[t, x]], t → restrict[inverse[S], x, x]] // Reverse

Out[16]= or[not[subclass[cart[x, x], union[S, inverse[S]]]],
           subclass[x, union[image[PS, x], set[A[x]]]]] == True

In[17]:= or[not[subclass[cart[x_, x_], union[S, inverse[S]]]],
           subclass[x_, union[image[PS, x_], set[A[x_]]]]] := True
```

Comment. A variable-free restatement of this is already available.

```
In[18]:= subclass[chains[S], union[fix[composite[E, BIGCAP]], subvar[PS]]]

Out[18]= True
```