greatest[to[x], fix[to[x]]]

Johan G. F. Belinfante
2009 May 2

```
In[1]:= SetDirectory["l:"]; << goedel.09may01b;<< tools.m

:Package Title: goedel.09may01b            2009 May 1 at 9:55 p.m.

It is now:  2009 May 2 at 19:6

Loading Simplification Rules

TOOLS.M                                    Revised 2009 April 6

weightlimit = 40
```

summary

The set of maximal elements and the set of greatest elements of a total order are the same, and each of these is equal to the fixed point class of the funpart of the total order. This set is either empty or a singleton. The funpart of a nonempty finite total order is a singleton.

maximal means greatest

Lemma.

```
In[2]:= Map[not,
         SubstTest[and, implies[p1, p3], implies[and[p2, p3], p4], implies[and[p1, p4], p5],
          not[implies[and[p1, p2], p5]], {p1 -> member[pair[u, v], to[x]],
           p2 -> subclass[fix[to[x]], image[inverse[to[x]], set[u]]], p3 →
            member[v, fix[to[x]]], p4 → member[pair[v, u], to[x]], p5 → equal[u, v]}]] // Reverse

Out[2]= or[equal[u, v], not[member[pair[u, v], to[x]]],
         not[subclass[fix[to[x]], image[inverse[to[x]], set[u]]]]] == True

In[3]:= (% /. {u → u_, v → v_, x → x_}) /. Equal → SetDelayed
```

Removing the variables from the lemma yields:

Theorem.

```
In[4]:= Map[empty[composite[Id, complement[#]]] &,
         SubstTest[class, pair[u, v], or[equal[u, v], not[member[pair[u, v], t]],
           not[subclass[fix[t], image[inverse[t], set[u]]]]], t → to[x]]]

Out[4]= subclass[composite[to[x], id[ub[to[x], fix[to[x]]]]], Id] == True

In[5]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[6]:= SubstTest[subclass, greatest[t, domain[t]], maximal[t, range[t]], t → to[x]] // Reverse

Out[6]= subclass[intersection[fix[to[x]], ub[to[x], fix[to[x]]]], fix[funpart[to[x]]]] == True

In[7]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma.

```
In[8]:= SubstTest[empty, dif[u, v], {u -> cart[fix[to[x]], fix[funpart[to[x]]]], v -> to[x]}]

Out[8]= subclass[cart[fix[to[x]], fix[funpart[to[x]]]], to[x]] == True

In[9]:= subclass[cart[fix[to[x_]], fix[funpart[to[x_]]]], to[x_]] := True
```

Theorem. The class of greatest elements of a total order is equal to the class of maximal elements.

```
In[10]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> greatest[to[x], fix[to[x]]], v -> maximal[to[x], fix[to[x]]]}]

Out[10]= equal[fix[funpart[to[x]]], intersection[fix[to[x]], ub[to[x], fix[to[x]]]]] == True

In[11]:= intersection[fix[to[x_]], ub[to[x_], fix[to[x_]]]] := fix[funpart[to[x]]]
```

Restatement.

```
In[12]:= greatest[to[x], fix[to[x]]] == maximal[to[x], fix[to[x]]]

Out[12]= True
```

uniqueness of greatest elements

Lemma.

```
In[16]:= SubstTest[empty, fix[funpart[po[t]]], t → to[x]] // Reverse

Out[16]= equal[0, fix[funpart[to[x]]]] == equal[0, funpart[to[x]]]

In[17]:= equal[0, fix[funpart[to[x_]]]] := equal[0, funpart[to[x]]]
```

Theorem. A total order has at most one maximal element.

```
In[20]:= Map[or[equal[0, funpart[to[x]]], #] &, SubstTest[member, greatest[po[t], y],
            union[set[0], range[SINGLETON]], {t → to[x], y → fix[to[x]]}]] // Reverse

Out[20]= or[equal[0, funpart[to[x]]], member[fix[funpart[to[x]]], range[SINGLETON]]] == True

In[21]:= (% /. x → x_) /. Equal → SetDelayed
```

Corollary. A better rewrite rule.

```
In[22]:= equiv[member[fix[funpart[to[x]]], range[SINGLETON]], not[equal[0, funpart[to[x]]]]]

Out[22]= True

In[24]:= member[fix[funpart[to[x_]]], range[SINGLETON]] := not[equal[0, funpart[to[x]]]]
```

Corollary.

```
In[27]:= SubstTest[member, id[t], range[SINGLETON], t → fix[funpart[to[x]]]] // Reverse

Out[27]= member[funpart[to[x]], range[SINGLETON]] == not[equal[0, funpart[to[x]]]]

In[28]:= member[funpart[to[x_]], range[SINGLETON]] := not[equal[0, funpart[to[x]]]]
```

Corollary. The funpart of a total order is a set.

```
In[29]:= SubstTest[implies, member[u, v], member[u, V],
           {u → funpart[to[x]], v -> union[set[0], range[SINGLETON]]}] // Reverse

Out[29]= member[funpart[to[x]], V] == True

In[30]:= member[funpart[to[x_]], V] := True
```

If **t** is a total order then its funpart is either empty or is a singleton and an identity function. In the latter case **funpart[t] = set[PAIR[u, u]]**, where **u** is the greatest element of **fix[t]**. In either case the funpart of a total order is a cartesian square.

Corollary. The funpart of a total order is a cartesian square.

```
In[31]:= SubstTest[member, t, intersection[u, v],
          {t → funpart[to[x]], u → image[CART, Id], v → P[Id]}]

Out[31]= equal[cart[fix[funpart[to[x]]], fix[funpart[to[x]]]], funpart[to[x]]] == True

In[32]:= cart[fix[funpart[to[x_]]], fix[funpart[to[x_]]]] := funpart[to[x]]
```

Theorem. Every non-empty finite total order has a greatest element.

```
In[33]:= Map[not, SubstTest[member, fix[to[t]], domain[GREATEST[to[t]]], t → fin[x]]]

Out[33]= equal[0, funpart[to[fin[x]]]] == equal[0, to[fin[x]]]

In[34]:= equal[0, funpart[to[fin[x_]]]] := equal[0, to[fin[x]]]
```

miscellaneous rewrite rules

Theorem. Simplification rule.

```
In[37]:= Assoc[to[x], id[fix[to[x]]], id[ub[to[x], fix[to[x]]]]] // Reverse

Out[37]= composite[to[x], id[ub[to[x], fix[to[x]]]]] == funpart[to[x]]

In[38]:= composite[to[x_], id[ub[to[x_], fix[to[x_]]]]] := funpart[to[x]]
```

Corollary.

```
In[39]:= ImageComp[to[x], id[ub[to[x], fix[to[x]]]], V] // Reverse

Out[39]= image[to[x], ub[to[x], fix[to[x]]]] == fix[funpart[to[x]]]

In[40]:= image[to[x_], ub[to[x_], fix[to[x_]]]] := fix[funpart[to[x]]]
```