lower and upper bounds for partial orders

Johan G. F. Belinfante
2010 August 29

```
In[1]:= SetDirectory["l:"]; << goedel.10aug28a; << tools.m

        :Package Title: goedel.10aug28a            2010 August 28 at 10:25 a.m.

        It is now:  2010 Aug 29 at 19:22

        Loading Simplification Rules

        TOOLS.M                                    Revised 2010 August 11

        weightlimit = 40
```

summary

Rewrite rules about lower and upper bounds for partial orders are derived in this notebook. Several of these were inspired by theorems in the following reference, including Theorem 8.12 on page 32 and Theorem 10.1 on page 40.

```
In[2]:= "Egbert Harzheim, Ordered Sets, Advances in Mathematics, volume 7, Springer
          Science+Business Media, Inc., 2005.  ISBN 0387-24219-8.  QA171.48 .H37";
```

Following Harzheim, a class **t** is called a **final segment** of a partial order **po[x]** if **image[po[x], t] = t**. A class **t** is called an **initial segment** of **po[x]** if it is a final segment of **inverse[po[x]]**.

invariant classes

A class **t** is **invariant** under **p** if the inclusion **image[p, t] ⊂ t** holds. In this section it is shown that a class is invariant under a partial order **po[x]** if and only if its intersection with **fix[po[t]]** is a final segment.

Theorem. An invariant subclass of **fix[po[x]]** is a final segment.

```
In[3]:= SubstTest[and, subclass[y, z], subclass[z, y], z -> image[po[x], y]] // Reverse

Out[3]= and[subclass[y, fix[po[x]]], subclass[image[po[x], y], y]] == equal[y, image[po[x], y]]

In[4]:= and[subclass[y_, fix[po[x_]]], subclass[image[po[x_], y_], y_]] :=
          equal[y, image[po[x], y]]
```

Corollary. A subclass of **fix[po[x]]** that is invariant under **inverse[po[x]]** is an initial segment.

```
In[5]:= SubstTest[and, subclass[y, fix[po[t]]],
          subclass[image[po[t], y], y], t → inverse[po[x]]] // Reverse

Out[5]= and[subclass[y, fix[po[x]]], subclass[image[inverse[po[x]], y], y]] ==
         equal[y, image[inverse[po[x]], y]]

In[6]:= and[subclass[y_, fix[po[x_]]], subclass[image[inverse[po[x_]], y_], y_]] :=
         equal[y, image[inverse[po[x]], y]]
```

Theorem. If **y** is invariant under **po[x]**, then **fix[po[x]]** ⋂ **y** is a final segment.

```
In[7]:= Map[implies[subclass[image[po[x], y], y], #] &, SubstTest[and, subclass[t, fix[po[x]]],
          subclass[image[po[x], t], t], t → intersection[y, fix[po[x]]]]]

Out[7]= or[equal[image[po[x], y], intersection[y, fix[po[x]]]],
          not[subclass[image[po[x], y], y]]] == True

In[8]:= or[equal[image[po[x_], y_], intersection[y_, fix[po[x_]]]],
          not[subclass[image[po[x_], y_], y_]]] := True
```

Corollary. If **y** is invariant under **inverse[po[x]]**, then **fix[po[x]]** ⋂ **y** is an initial segment.

```
In[9]:= SubstTest[implies, subclass[image[po[t], y], y],
          equal[image[po[t], y], intersection[y, fix[po[t]]]], t → inverse[po[x]]] // Reverse

Out[9]= or[equal[image[inverse[po[x]], y], intersection[y, fix[po[x]]]],
          not[subclass[image[inverse[po[x]], y], y]]] == True

In[10]:= or[equal[image[inverse[po[x_]], y_], intersection[y_, fix[po[x_]]]],
           not[subclass[image[inverse[po[x_]], y_], y_]]] := True
```

eliminating the variable y

The results of the preceding section are restated in this section after eliminating the variable **y**.

Lemma. An inclusion in one direction.

```
In[11]:= Map[equal[V, #] &,
          complement[dif[image[inverse[IMAGE[id[fix[po[x]]]]], fix[IMAGE[po[x]]]],
             invar[po[x]]]] // Normality]

Out[11]= subclass[image[inverse[IMAGE[id[fix[po[x]]]]], fix[IMAGE[po[x]]]], invar[po[x]]] ==
          True

In[12]:= (% /. x → x_) /. Equal → SetDelayed
```

Lemma. An inclusion in the opposite direction.

```
In[13]:= Map[equal[V, #] &, SubstTest[class, y, member[setpart[y], t], t → complement[
            dif[invar[po[x]], image[inverse[IMAGE[id[fix[po[x]]]]], fix[IMAGE[po[x]]]]]]]]

Out[13]= subclass[image[IMAGE[id[fix[po[x]]]], invar[po[x]]], fix[IMAGE[po[x]]]] == True

In[14]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. An equation obtained by combining the above two inclusions.

```
In[15]:= SubstTest[and, subclass[u, v], subclass[v, u],
          {u -> image[inverse[IMAGE[id[fix[po[x]]]]], fix[IMAGE[po[x]]]], v -> invar[po[x]]}]

Out[15]= equal[image[inverse[IMAGE[id[fix[po[x]]]]], fix[IMAGE[po[x]]]], invar[po[x]]] == True

In[16]:= image[inverse[IMAGE[id[fix[po[x_]]]]], fix[IMAGE[po[x_]]]] := invar[po[x]]
```

Corollary. An equation relating the class of invariant subsets of a partial order and the class of final segments.

```
In[17]:= ImageComp[IMAGE[id[fix[po[x]]]],
           inverse[IMAGE[id[fix[po[x]]]]], fix[IMAGE[po[x]]]] // Reverse

Out[17]= image[IMAGE[id[fix[po[x]]]], invar[po[x]]] == fix[IMAGE[po[x]]]

In[18]:= image[IMAGE[id[fix[po[x_]]]], invar[po[x_]]] := fix[IMAGE[po[x]]]
```

Corollary. An equation relating the class of subsets invariant under the inverse of a partial order and the class of initial segments.

```
In[19]:= SubstTest[image, IMAGE[id[fix[po[t]]]], invar[po[t]], t → inverse[po[x]]] // Reverse

Out[19]= image[IMAGE[id[fix[po[x]]]], invar[inverse[po[x]]]] == fix[IMAGE[inverse[po[x]]]]

In[20]:= image[IMAGE[id[fix[po[x_]]]], invar[inverse[po[x_]]]] := fix[IMAGE[inverse[po[x]]]]
```

---

derivation

Theorem. The relative complement in **fix[po[x]]** of any class of lower bounds is a final segment.

```
In[21]:= ImageComp[po[x], complement[inverse[po[x]]], y] // Reverse

Out[21]= image[po[x], complement[lb[po[x], y]]] ==
          intersection[complement[lb[po[x], y]], fix[po[x]]]

In[22]:= image[po[x_], complement[lb[po[x_], y_]]] :=
          intersection[complement[lb[po[x], y]], fix[po[x]]]
```

Theorem. The relative complement in **fix[po[x]]** of any class of upper bounds is an initial segment.

```
In[23]:= SubstTest[image, po[t], complement[lb[po[t], y]], t → inverse[po[x]]] // Reverse

Out[23]= image[inverse[po[x]], complement[ub[po[x], y]]] ==
          intersection[complement[ub[po[x], y]], fix[po[x]]]
```

```
In[24]:= image[inverse[po[x_]], complement[ub[po[x_], y_]]] :=
          intersection[complement[ub[po[x], y]], fix[po[x]]]
```

The complement of a class is invariant under a relation if and only if the class itself is invariant under the inverse relation. This observation yields the following corollaries.

Corollary.

```
In[25]:= SubstTest[invariant, u, complement[v], {u → inverse[po[x]], v → ub[po[x], y]}]

Out[25]= subclass[cart[y, image[po[x], ub[po[x], y]]], po[x]] == True

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

Corollary.

```
In[27]:= SubstTest[invariant, u, complement[v], {u → po[x], v → lb[po[x], y]}]

Out[27]= subclass[cart[image[inverse[po[x]], lb[po[x], y]], y], po[x]] == True

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

Using the results of the preceding sections, a stronger result is obtained, which can be made into a convenient rewrite rule.

Theorem. The class of upper bounds for any subclass **y ⊂ fix[po[x]]** is a final segment.

```
In[29]:= SubstTest[implies, invariant[po[x], t],
           equal[image[po[x], t], intersection[fix[po[x]], t]], t → ub[po[x], y]] // Reverse

Out[29]= equal[image[po[x], ub[po[x], y]], intersection[fix[po[x]], ub[po[x], y]]] == True

In[30]:= image[po[x_], ub[po[x_], y_]] := intersection[fix[po[x]], ub[po[x], y]]
```

Corollary. The class of lower bounds for **y ⊂ fix[po[x]]** is an initial segment.

```
In[31]:= SubstTest[image, po[t], ub[po[t], y], t → inverse[po[x]]] // Reverse

Out[31]= image[inverse[po[x]], lb[po[x], y]] == intersection[fix[po[x]], lb[po[x], y]]

In[32]:= image[inverse[po[x_]], lb[po[x_], y_]] := intersection[fix[po[x]], lb[po[x], y]]
```

eliminating the variable y

The results of the preceding section are here restated after using **reify** to eliminate the variable **y**.

Theorem. A formula for the composite of a partial order and its corresponding upper bound relation.

```
In[33]:= SubstTest[reify, y, image[t, ub[t, y]], t → po[x]]

Out[33]= composite[po[x], UB[po[x]]] == composite[id[fix[po[x]]], UB[po[x]]]
```

```
In[34]:= composite[po[x_], UB[po[x_]]] := composite[id[fix[po[x]]], UB[po[x]]]
```

Corollary. A similar result involving the lower bound relation.

```
In[35]:= SubstTest[composite, po[t], UB[po[t]], t → inverse[po[x]]] // Reverse

Out[35]= composite[inverse[po[x]], LB[po[x]]] == composite[id[fix[po[x]]], LB[po[x]]]

In[36]:= composite[inverse[po[x_]], LB[po[x_]]] := composite[id[fix[po[x]]], LB[po[x]]]
```

For the subset relation **S**, this yields the following result (which can also be independently derived using **RelnNormality**).

Theorem.

```
In[37]:= SubstTest[composite, po[x], UB[po[x]], x → S] // Reverse

Out[37]= composite[S, inverse[POWER], S] == composite[inverse[POWER], S]

In[38]:= composite[S, inverse[POWER], S] := composite[inverse[POWER], S]
```

For thin partial orders, the relation **LB[po[x]]** is rewritten, so an additional rewrite rule is needed.

Theorem.

```
In[39]:= Map[inverse,
           SubstTest[composite, inverse[po[t]], LB[po[t]], t → thinpart[x]]] // Reverse

Out[39]= composite[inverse[S], VERTSECT[po[thinpart[x]]], po[thinpart[x]]] ==
          composite[inverse[S], VERTSECT[po[thinpart[x]]], id[fix[po[thinpart[x]]]]]

In[40]:= composite[inverse[S], VERTSECT[po[thinpart[x_]]], po[thinpart[x_]]] :=
          composite[inverse[S], VERTSECT[po[thinpart[x]]], id[fix[po[thinpart[x]]]]]
```

This is the case, for example, for the divisibility relation.

Theorem.

```
In[41]:= Map[inverse, SubstTest[composite, po[x], UB[po[x]], x → inverse[DIV]]] // Reverse

Out[41]= composite[inverse[S], VERTSECT[DIV], DIV] ==
          composite[inverse[S], VERTSECT[DIV], id[omega]]

In[42]:= composite[inverse[S], VERTSECT[DIV], DIV] :=
          composite[inverse[S], VERTSECT[DIV], id[omega]]
```