convex subsets

Johan G. F. Belinfante
2010 July 26

```
In[1]:= SetDirectory["l:"]; << goedel.10jul25a; << tools.m

:Package Title: goedel.10jul25a              2010 July 25 at 8:50 p.m.

It is now:  2010 Jul 26 at 16:16

Loading Simplification Rules

TOOLS.M                                     Revised 2010 July 24

weightlimit = 40
```

summary

A set **t** is said to be a **convex subset** or **segment** for a partial order **p** if

t = image[p, t] ⋂ image[inverse[p], t].

It is shown that the intersection of any (nonempty) set of convex subsets is convex, and that the union of any chain of convex subsets is convex. These results are discussed in pages 29-31 of the following reference.

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

Since the hypothesis that **p** be a partial order is not needed for some of these results, the definitions and theorems in this notebook will be given for an arbitrary class, and only specialized to partial orders when necessary.

convex subsets

Because full equality substitution can easily produce combinatorial explosions, the **GOEDEL** program has only a restricted version for equality substitution. The following lemma is needed to overcome these limitations.

Technical Lemma. (A special case of equality substitution.)

```
In[3]:= SubstTest[implies, and[member[w, u], member[w, v], equal[t, intersection[u, v]]],
          member[w, t], {u → image[x, t], v → image[inverse[x], t]}] // Reverse

Out[3]= or[member[w, t], not[equal[t, intersection[image[x, t], image[inverse[x], t]]]],
          not[member[w, image[x, t]]], not[member[w, image[inverse[x], t]]]] == True
```

```
In[4]:= (% /. {t → t_, w → w_, x → x_, y → y_}) /. Equal → SetDelayed
```

In the following **x** can be any class. It will be convenient to write **u ≤ v** if **u, v** are sets satisfying **pair[u, v] ∈ x**. A set **w** is said to be **between** the sets **u** and **v** if **u ≤ w** and **w ≤ v**. The convexity property can be expressed as follows.

Theorem. If **t** is convex, then all sets between any two members of **t** are also members of **t**.

```
In[5]:= Map[not, SubstTest[and, implies[and[p2, p4], p5], implies[and[p3, p4], p6],
          implies[and[p1, p5, p6], p7], not[implies[and[p1, p2, p3, p4], p7]],
          {p1 → equal[t, intersection[image[x, t], image[inverse[x], t]]],
           p2 → and[member[u, t], member[pair[u, w], x]],
           p3 → and[member[v, t], member[pair[w, v], x]],
           p4 → member[w, y], p5 → member[w, image[x, t]],
           p6 → member[w, image[inverse[x], t]], p7 → member[w, t]}]] // Reverse

Out[5]= or[member[w, t], not[equal[t, intersection[image[x, t], image[inverse[x], t]]]],
         not[member[u, t]], not[member[v, t]], not[member[w, y]],
         not[member[pair[u, w], x]], not[member[pair[w, v], x]]] == True

In[6]:= or[member[w_, t_], not[equal[intersection[image[inverse[x_], t_], image[x_, t_]], t_]],
         not[member[pair[u_, w_], x_]], not[member[pair[w_, v_], x_]],
         not[member[u_, t_]], not[member[v_, t_]], not[member[w_, y_]]] := True
```

convex[x]

Let **x** be any class. The class **convex[x]** of all convex subsets for **x** is defined by the following membership rule.

```
In[7]:= member[t_, convex[x_]] :=
          and[member[t, V], equal[t, intersection[image[x, t], image[inverse[x], t]]]]
```

Comment. The empty set is always convex.

```
In[8]:= member[0, convex[x]]

Out[8]= True
```

Lemma.

```
In[9]:= symdif[convex[x], fix[
          composite[IMAGE[intersection[composite[x, FIRST], composite[inverse[x], SECOND]]],
           CART, DUP]]] // Normality

Out[9]= union[intersection[complement[convex[x]], fix[composite[IMAGE[
              intersection[composite[x, FIRST], composite[inverse[x], SECOND]]], CART, DUP]]],
         intersection[complement[fix[composite[IMAGE[intersection[composite[x, FIRST],
                composite[inverse[x], SECOND]]], CART, DUP]]], convex[x]]] == 0

In[10]:= (% /. x → x_) /. Equal → SetDelayed
```

Theorem. A formula for the class **convex[x]**.

```
In[11]:= SubstTest[empty, symdif[u, v],
          {u → convex[x], v -> fix[composite[IMAGE[intersection[composite[x, FIRST],
                composite[inverse[x], SECOND]]], CART, DUP]]}]

Out[11]= equal[convex[x], fix[
            composite[IMAGE[intersection[composite[x, FIRST], composite[inverse[x], SECOND]]],
             CART, DUP]]] == True

In[12]:= fix[
           composite[IMAGE[intersection[composite[x_, FIRST], composite[inverse[x_], SECOND]]],
            CART, DUP]] := convex[x]
```

It is clear from the definition that the convex subsets of **x** and **inverse[x]** are the same.

```
In[13]:= Map[empty, symdif[convex[inverse[x]], convex[x]] // Normality]

Out[13]= equal[convex[x], convex[inverse[x]]] == True

In[14]:= convex[inverse[x_]] := convex[x]
```

Corollary.

```
In[15]:= SubstTest[convex, inverse[t], t → inverse[x]] // Reverse

Out[15]= convex[composite[Id, x]] == convex[x]

In[16]:= convex[composite[Id, x_]] := convex[x]
```

Theorem. The union of a chain of convex subsets is convex.

```
In[17]:= SubstTest[Uchains, fix[composite[IMAGE[t], CART, DUP]],
           t -> intersection[composite[x, FIRST], composite[inverse[x], SECOND]]] // Reverse

Out[17]= Uchains[convex[x]] == convex[x]

In[18]:= Uchains[convex[x_]] := convex[x]
```

inclusions

Theorem.

```
In[19]:= SubstTest[subclass,
           fix[composite[IMAGE[intersection[composite[x, FIRST], composite[y, SECOND]]],
             CART, DUP]], subvar[x], y → inverse[x]] // Reverse

Out[19]= subclass[convex[x], subvar[x]] == True

In[20]:= subclass[convex[x_], subvar[x_]] := True
```

Corollary.

```
In[21]:= SubstTest[subclass, convex[t], subvar[t], t → inverse[x]] // Reverse

Out[21]= subclass[convex[x], subvar[inverse[x]]] == True

In[22]:= subclass[convex[x_], subvar[inverse[x_]]] := True
```

Theorem.

```
In[24]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]],
           subclass[u, w], {u → convex[x], v → subvar[x], w → P[range[x]]}] // Reverse

Out[24]= subclass[U[convex[x]], range[x]] == True

In[25]:= subclass[U[convex[x_]], range[x_]] := True
```

Corollary.

```
In[26]:= SubstTest[subclass, U[convex[t]], range[t], t → inverse[x]] // Reverse

Out[26]= subclass[U[convex[x]], domain[x]] == True

In[27]:= subclass[U[convex[x_]], domain[x_]] := True
```

convex subsets of a partial order

Lemma.

```
In[28]:= SubstTest[implies, and[subclass[u, v], subclass[v, w]],
           subclass[u, w], {u → fix[IMAGE[x]], v → invar[x], w -> binclosed[
              intersection[composite[x, FIRST], composite[inverse[x], SECOND]]]}] // Reverse

Out[28]= subclass[fix[IMAGE[x]],
           binclosed[intersection[composite[x, FIRST], composite[inverse[x], SECOND]]]] == True

In[29]:= subclass[fix[IMAGE[x_]], binclosed[
            intersection[composite[x_, FIRST], composite[inverse[x_], SECOND]]]] := True
```

Theorem. Any final segment of a partial order is convex.

```
In[30]:= Map[subclass[fix[IMAGE[po[x]]], #] &, SubstTest[intersection,
           binclosed[intersection[composite[t, FIRST], composite[inverse[t], SECOND]]],
           subvar[t], subvar[inverse[t]], t → po[x]]]

Out[30]= subclass[fix[IMAGE[po[x]]], convex[po[x]]] == True

In[31]:= subclass[fix[IMAGE[po[x_]]], convex[po[x_]]] := True
```

Corollary. Initial segments of partial orders are convex.

```
In[32]:= SubstTest[subclass, fix[IMAGE[po[t]]], convex[po[t]], t → inverse[x]] // Reverse

Out[32]= subclass[fix[IMAGE[inverse[po[x]]]], convex[po[x]]] == True
```

```
In[33]:= subclass[fix[IMAGE[inverse[po[x_]]]], convex[po[x_]]] := True
```

Theorem. The class of convex subsets of a partial order is closed under arbitrary intersections.

```
In[34]:= Map[equal[fix[HULL[#]], #] &, SubstTest[intersection,
           binclosed[intersection[composite[t, FIRST], composite[inverse[t], SECOND]]],
           subvar[t], subvar[inverse[t]], t → po[x]]]

Out[34]= equal[convex[po[x]], fix[HULL[convex[po[x]]]]] == True

In[35]:= fix[HULL[convex[po[x_]]]] := convex[po[x]]
```

antichains for a partial order are convex

The proof step **implies[and[p1, p2, p3, p6], p7]** was deliberately omitted in the following lemma to expedite execution.

Lemma.

```
In[36]:= Map[not, SubstTest[and, implies[and[p4, p5], p6], implies[and[p4, p5, p7], p8],
            implies[and[p2, p8], p9], not[implies[and[p1, p2, p3, p4, p5], p9]],
            {p1 → equal[composite[id[t], po[x], id[t]], id[t]],
             p2 → member[u, t], p3 → member[v, t], p4 → member[pair[u, w], po[x]],
             p5 → member[pair[w, v], po[x]], p6 → member[pair[u, v], po[x]],
             p7 → equal[u, v], p8 → equal[w, u], p9 → member[w, t]}]] // Reverse

Out[36]= or[member[w, t], not[equal[composite[id[t], po[x], id[t]], id[t]]],
           not[member[u, t]], not[member[v, t]], not[member[pair[u, w], po[x]]],
           not[member[pair[w, v], po[x]]]] == True

In[37]:= (% /. {t → t_, u → u_, v → v_, w → w_, x → x_}) /. Equal → SetDelayed
```

Lemma. Eliminating the three variables **u**, **v** and **w**.

```
In[38]:= Map[empty[composite[complement[#], id[cart[V, V]]]] &,
           SubstTest[class, pair[pair[u, v], w], implies[and[equal[r, s], member[u, t],
              member[v, t], member[pair[u, w], p], member[pair[w, v], p]], member[w, t]],
            {p → po[x], r -> composite[id[t], po[x], id[t]], s -> id[t]}]] // MapNotNot

Out[38]= or[not[equal[composite[id[t], po[x], id[t]], id[t]]],
           subclass[intersection[image[inverse[po[x]], t], image[po[x], t]], t]] == True

In[39]:= (% /. {t → t_, x → x_}) /. Equal → SetDelayed
```

Lemma.

```
In[40]:= SubstTest[and, subclass[s, t], subclass[t, s],
           s -> intersection[image[inverse[po[x]], t], image[po[x], t]]] // Reverse

Out[40]= and[subclass[t, fix[po[x]]],
           subclass[intersection[image[inverse[po[x]], t], image[po[x], t]], t]] ==
          equal[t, intersection[image[inverse[po[x]], t], image[po[x], t]]]
```

```
In[41]:= and[subclass[t_, fix[po[x_]]],
           subclass[intersection[image[inverse[po[x_]], t_], image[po[x_], t_]], t_]] :=
          equal[t, intersection[image[inverse[po[x]], t], image[po[x], t]]]
```

Corollary.

```
In[42]:= or[equal[t, intersection[image[inverse[po[x]], t], image[po[x], t]]],
           not[subclass[t, fix[po[x]]]], not[
            subclass[intersection[image[inverse[po[x]], t], image[po[x], t]], t]]] // NotNotTest

Out[42]= or[equal[t, intersection[image[inverse[po[x]], t], image[po[x], t]]],
           not[subclass[t, fix[po[x]]]],
           not[subclass[intersection[image[inverse[po[x]], t], image[po[x], t]], t]]] == True

In[43]:= (% /. {t → t_, x → x_}) /. Equal → SetDelayed
```

Theorem. Antichains of a partial order are convex.

```
In[44]:= Map[not, SubstTest[and, implies[p1, p2],
            implies[p1, p3], implies[and[p2, p3], p4], not[implies[p1, p4]],
            {p1 → equal[composite[id[t], po[x], id[t]], id[t]], p2 → subclass[t, fix[po[x]]],
             p3 → subclass[intersection[image[inverse[po[x]], t], image[po[x], t]], t],
             p4 → equal[intersection[image[inverse[po[x]], t], image[po[x], t]], t]}]] // Reverse

Out[44]= or[equal[t, intersection[image[inverse[po[x]], t], image[po[x], t]]],
           not[equal[composite[id[t], po[x], id[t]], id[t]]]] == True

In[45]:= or[equal[intersection[image[inverse[po[x_]], t_], image[po[x_], t_]], t_],
           not[equal[composite[id[t_], po[x_], id[t_]], id[t_]]]] := True
```

Eliminating the variable **t** yields this corollary:

Corollary. Antichains of a partial order are convex.

```
In[46]:= Map[equal[V, #] &, SubstTest[class, t,
           not[member[setpart[t], w]], w -> dif[antichains[po[x]], convex[po[x]]]]]]

Out[46]= subclass[antichains[po[x]], convex[po[x]]] == True

In[47]:= subclass[antichains[po[x_]], convex[po[x_]]] := True
```

a counterexample

The following counterexample shows that the inclusion **fix[IMAGE[x]] ⊂ convex[x]** does not hold for an arbitrary class **x**.

```
In[48]:= (subclass[fix[IMAGE[x]], convex[x]] /. x → cart[set[0], succ[set[0]]]) // assert

Out[48]= False
```

symmetric x

Theorem. If **x** is symmetric, then **convex[x] = fix[IMAGE[x]]**.

```
In[49]:= SubstTest[implies, equal[y, inverse[x]],
           equal[fix[IMAGE[x]], fix[composite[IMAGE[intersection[composite[x, FIRST],
                composite[inverse[y], SECOND]]], CART, DUP]]], y → x] // Reverse

Out[49]= or[equal[convex[x], fix[IMAGE[x]]], not[equal[x, inverse[x]]]] == True

In[50]:= or[equal[convex[x_], fix[IMAGE[x_]]], not[equal[x_, inverse[x_]]]] := True
```

Theorem. An example.

```
In[51]:= convex[id[x]] // Normality

Out[51]= convex[id[x]] == P[x]

In[52]:= convex[id[x_]] := P[x]
```

Theorem. Another example.

```
In[53]:= equal[set[0], convex[DISJOINT]] // assert

Out[53]= True

In[54]:= convex[DISJOINT] := set[0]
```