Lagrange's theorem for finite groups

Johan G. F. Belinfante
2011 July 27

```
In[1]:= SetDirectory["l:"]; << goedel.11jul27a

:Package Title: goedel.11jul27a                2011 July 27 at 11:40 a.m.

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

It is now:  2011 Jul 27 at 13:24

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 Jul 27 at 13:35
```

summary

If **x** is a subgroup of a finite group **y**, then the cardinality of **range[x]** divides the cardinality of **range[y]**.

the left coset of the identity element

If **x** is a subgroup of a group **y**, then **range[x]** is the left coset of the identity element **e[y]**.

Theorem. If **x** is a subgroup of a group **y**, then **range[x]** belongs to the class of all left cosets.

```
In[3]:= Map[not, SubstTest[and, implies[p1, p2], implies[p1, p3],
      implies[and[p1, p2], p4], implies[and[p3, p4], p5], not[implies[p1, p5]],
      {p1 → and[member[x, GROUPS], subclass[x, y], member[y, GROUPS]], p2 →
        member[e[y], range[y]], p3 → equal[range[x], image[y, cart[set[e[y]], range[x]]]],
       p4 → member[image[y, cart[set[e[y]], range[x]]],
         image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]], range[y]]],
       p5 → member[range[x], image[VERTSECT[composite[y, id[cart[V, range[x]]],
            inverse[FIRST]]], range[y]]]}]] // Reverse

Out[3]= or[member[range[x],
      image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]], range[y]]],
     not[member[x, GROUPS]], not[member[y, GROUPS]], not[subclass[x, y]]] == True
```

```
In[4]:= or[member[range[x_],
           image[VERTSECT[composite[y_, id[cart[V, range[x_]]], inverse[FIRST]]], range[y_]]],
          not[member[x_, GROUPS]], not[member[y_, GROUPS]], not[subclass[x_, y_]]] := True
```

The left coset **image[y,{t}× range[x]]** for every element **t ∈ range[y]** is equipollent to the particular coset **range[x]**. The next lemma just restates this fact, eliminating the variable **t**.

Lemma. (Eliminating the variable **t**.)

```
In[5]:= Map[equal[V, #] &,
         SubstTest[class, t, implies[and[member[x, g], subclass[x, y], member[y, g],
            member[t, range[y]]], member[t, q]], {g → GROUPS,
           q → image[inverse[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]]],
             image[Q, set[range[x]]]], r → range[x]}]]

Out[5]= or[not[member[x, GROUPS]],
          not[member[y, GROUPS]], not[subclass[x, y]], subclass[range[y],
           image[inverse[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]]],
            image[Q, set[range[x]]]]]] == True

In[6]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

The inverse image in the above lemma can be replaced with a direct image, yielding the following somewhat nicer statement.

Theorem. Every left coset of a subgroup **x** of a group **y** is equipollent to **range[x]**.

```
In[7]:= Map[not, SubstTest[and, implies[p1, p2], implies[p2, p3], not[implies[p1, p3]],
           {p1 -> and[member[x, GROUPS], subclass[x, y], member[y, GROUPS]],
            p2 -> subclass[range[y], image[inverse[VERTSECT[composite[y,
                  id[cart[V, range[x]]], inverse[FIRST]]]], image[Q, set[range[x]]]]],
            p3 -> subclass[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
               range[y]], image[Q, set[range[x]]]]}]] // Reverse

Out[7]= or[not[member[x, GROUPS]], not[member[y, GROUPS]], not[subclass[x, y]],
          subclass[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
            range[y]], image[Q, set[range[x]]]]] == True

In[8]:= or[not[member[x_, GROUPS]], not[member[y_, GROUPS]], not[subclass[x_, y_]],
          subclass[image[VERTSECT[composite[y_, id[cart[V, range[x_]]], inverse[FIRST]]],
            range[y_]], image[Q, set[range[x_]]]]] := True
```

Since the left cosets are pairwise disjoint, it follows that the class of left cosets is a uniform partition of **range[y]**.

uniform partitions of finite sets

A corollary of a recently derived theorem about uniform partitions of finite sets is derived in this section.

Lemma.

```
In[9]:= SubstTest[or, equal[card[U[x]], natmul[t, card[x]]], not[member[x, FINITE]],
          not[member[t, omega]], not[subclass[x, image[Q, set[t]]]],
          not[subclass[cart[x, x], union[DISJOINT, Id]]], {t → card[y]}] // Reverse

Out[9]= or[equal[card[U[x]], natmul[card[x], card[y]]], not[member[x, FINITE]],
          not[member[y, FINITE]], not[subclass[x, image[Q, set[card[y]]]]],
          not[subclass[cart[x, x], union[DISJOINT, Id]]]] == True

In[10]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Theorem. If each member of a finite partition **x** is equipollent to the same finite set **y**, then **card[U[x]]** is the product of **card[x]** and **card[y]**.

```
In[13]:= Map[not, SubstTest[and, implies[and[p1, p2, p4, p6], p7], implies[p2, p5],
            implies[and[p3, p5], p6], not[implies[and[p1, p2, p3, p4], p7]],
            {p1 -> member[x, FINITE], p2 -> member[y, FINITE],
             p3 -> subclass[x, image[Q, set[y]]], p4 -> subclass[cart[x, x], union[DISJOINT, Id]],
             p5 → equal[image[Q, set[y]], image[Q, set[card[y]]]],
             p6 → subclass[x, image[Q, set[card[y]]]],
             p7 -> equal[card[U[x]], natmul[card[x], card[y]]]}]] // Reverse

Out[13]= or[equal[card[U[x]], natmul[card[x], card[y]]], not[member[x, FINITE]],
           not[member[y, FINITE]], not[subclass[x, image[Q, set[y]]]],
           not[subclass[cart[x, x], union[DISJOINT, Id]]]] == True

In[14]:= or[equal[card[U[x_]], natmul[card[x_], card[y_]]], not[member[x_, FINITE]],
           not[member[y_, FINITE]], not[subclass[x_, image[Q, set[y_]]]],
           not[subclass[cart[x_, x_], union[DISJOINT, Id]]]] := True
```

This theorem about uniform partitions will now be applied to the equivalence classes of a finite equivalence relation **eqv[fin[x]]**.

Lemma.

```
In[18]:= SubstTest[member, image[funpart[u], fin[v]],
           FINITE, {u → VERTSECT[x], v → fix[eqv[fin[y]]]}] // Reverse

Out[18]= member[image[VERTSECT[x], fix[eqv[fin[y]]]], FINITE] == True

In[19]:= member[image[VERTSECT[x_], fix[eqv[fin[y_]]]], FINITE] := True
```

Lemma. (The uniform partition theorem is applied to a generic finite equivalence relation.)

```
In[20]:= SubstTest[or, equal[card[U[t]], natmul[card[t], card[y]]],
           not[member[t, FINITE]], not[member[y, FINITE]],
           not[subclass[t, image[Q, set[y]]]], not[subclass[cart[t, t], union[DISJOINT, Id]]],
           t → image[VERTSECT[eqv[fin[x]]], fix[eqv[fin[x]]]]] // Reverse

Out[20]= or[equal[card[fix[eqv[fin[x]]]],
            natmul[card[y], card[image[VERTSECT[eqv[fin[x]]], fix[eqv[fin[x]]]]]]],
           not[member[y, FINITE]], not[
            subclass[image[VERTSECT[eqv[fin[x]]], fix[eqv[fin[x]]]], image[Q, set[y]]]]] == True
```

```
In[21]:=  (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

The double wrapper **eqv[fin[x]]** can now be eliminated.

Theorem. If every equivalence class of a finite equivalence relation **x** is equipollent to the same finite set **y**, then the cardinality of **fix[x]** is the product of the number of equivalence classes and the cardinality of the set **y**.

```
In[22]:=  SubstTest[implies, equal[x, eqv[fin[t]]], or[equal[card[fix[x]],
              natmul[card[y], card[image[VERTSECT[x], fix[x]]]]], not[member[y, FINITE]],
            not[subclass[image[VERTSECT[x], fix[x]], image[Q, set[y]]]]], t → x] // Reverse

Out[22]=  or[equal[card[fix[x]], natmul[card[y], card[image[VERTSECT[x], fix[x]]]]],
            not[EQUIVALENCE[x]], not[member[x, FINITE]], not[member[y, FINITE]],
            not[subclass[image[VERTSECT[x], fix[x]], image[Q, set[y]]]]] == True

In[23]:=  or[equal[card[fix[x_]], natmul[card[y_], card[image[VERTSECT[x_], fix[x_]]]]],
            not[EQUIVALENCE[x_]], not[member[x_, FINITE]], not[member[y_, FINITE]],
            not[subclass[image[VERTSECT[x_], fix[x_]], image[Q, set[y_]]]]] := True
```

A slightly more general result can be derived.

Lemma.

```
In[26]:=  SubstTest[member, image[VERTSECT[fin[t]], y], FINITE, t → eqv[fin[x]]] // Reverse

Out[26]=  member[image[VERTSECT[eqv[fin[x]]], y], FINITE] == True

In[27]:=  member[image[VERTSECT[eqv[fin[x_]]], y_], FINITE] := True
```

Lemma.

```
In[28]:=  SubstTest[or, equal[card[U[t]], natmul[card[t], card[y]]],
            not[member[t, FINITE]], not[member[y, FINITE]],
            not[subclass[t, image[Q, set[y]]]], not[subclass[cart[t, t], union[DISJOINT, Id]]],
            t → image[VERTSECT[eqv[fin[x]]], z]] // Reverse

Out[28]=  or[equal[card[image[eqv[fin[x]], z]],
              natmul[card[y], card[image[VERTSECT[eqv[fin[x]]], z]]]], not[member[y, FINITE]],
            not[subclass[image[VERTSECT[eqv[fin[x]]], z], image[Q, set[y]]]]] == True

In[29]:=  (% /. {x → x_, y → y_, z → z_}) /. Equal → SetDelayed
```

Theorem. If every equivalence class of a finite equivalence relation **x** is equipollent to the same finite set **y**, then for any class **z**, the cardinality of **image[x, z]** is the product of the cardinality of **image[VERTSECT[x], z]** and the cardinality of the set **y**.

```
In[30]:= SubstTest[implies, equal[x, eqv[fin[t]]], or[equal[card[image[x, z]],
             natmul[card[y], card[image[VERTSECT[x], z]]]], not[member[y, FINITE]],
           not[subclass[image[VERTSECT[x], z], image[Q, set[y]]]]], t → x] // Reverse

Out[30]= or[equal[card[image[x, z]], natmul[card[y], card[image[VERTSECT[x], z]]]],
           not[EQUIVALENCE[x]], not[member[x, FINITE]], not[member[y, FINITE]],
           not[subclass[image[VERTSECT[x], z], image[Q, set[y]]]]] == True

In[31]:= or[equal[card[image[x_, z_]], natmul[card[image[VERTSECT[x_], z_]], card[y_]]],
           not[EQUIVALENCE[x_]], not[member[x_, FINITE]], not[member[y_, FINITE]],
           not[subclass[image[VERTSECT[x_], z_], image[Q, set[y_]]]]] := True
```

Lagrange’s theorem

In this section, the general result derived in the last section is used to derive Lagrange’s theorem for finite groups. If **x** is a subgroup of a finite group **y**, then **y ∘ id[V × range[x]] ∘ inverse[FIRST]** is an equivalence relation on **range[y]**. The equivalence classes are just the left cosets of **x** in **y**. For the first lemma, the group trappings are ignored, and only the form of the equivalence relation is used, producing a laundry list of hypotheses that need to be checked.

Lemma. (Result of applying the result of last section to an equivalence relation of the form needed.)

```
In[32]:= SubstTest[or, equal[card[image[u, w]], natmul[card[v], card[image[VERTSECT[u], w]]]],
           not[EQUIVALENCE[u]], not[member[u, FINITE]], not[member[v, FINITE]],
           not[subclass[image[VERTSECT[u], w], image[Q, set[v]]]],
           {u -> composite[y, id[cart[V, range[x]]], inverse[FIRST]],
            v → range[x], w → range[y]}] // Reverse

Out[32]= or[equal[card[image[y, cart[range[y], range[x]]]],
            natmul[card[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
               range[y]]], card[range[x]]]],
           not[EQUIVALENCE[composite[y, id[cart[V, range[x]]], inverse[FIRST]]]],
           not[member[composite[y, id[cart[V, range[x]]], inverse[FIRST]], FINITE]],
           not[member[range[x], FINITE]],
           not[subclass[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
              range[y]], image[Q, set[range[x]]]]]] == True

In[33]:=  (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma. (The first finiteness condition encountered in the lemma.)

```
In[38]:= SubstTest[implies,
           and[member[u, FINITE], FUNCTION[inverse[v]]], member[composite[u, v], FINITE],
           {u → y,  v → composite[id[cart[V, range[x]]], inverse[FIRST]]}] // Reverse

Out[38]= or[member[composite[y, id[cart[V, range[x]]], inverse[FIRST]], FINITE],
           not[member[y, FINITE]]] == True

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

Theorem. A preliminary result, using the above lemmas.

```
In[41]:= Map[not,
          SubstTest[and, implies[p1, p2], implies[p1, p3], implies[p1, p4], implies[p4, p5],
           implies[p1, p6], (*implies[and[p2,p3,p5,p6],p7],*) not[implies[p1, p7]],
           {p1 -> and[member[x, GROUPS], subclass[x, y], member[y, GROUPS], member[y, FINITE]],
            p2 → EQUIVALENCE[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
            p3 -> member[composite[y, id[cart[V, range[x]]], inverse[FIRST]], FINITE],
            p4 → member[x, FINITE],
            p5 -> member[range[x], FINITE], p6 -> subclass[
              image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]], range[y]],
              image[Q, set[range[x]]]], p7 -> equal[card[image[y, cart[range[y], range[x]]]],
              natmul[card[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
                 range[y]]], card[range[x]]]]}]] // Reverse

Out[41]= or[equal[card[image[y, cart[range[y], range[x]]]],
           natmul[card[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
              range[y]]], card[range[x]]]], not[member[x, GROUPS]],
          not[member[y, FINITE]], not[member[y, GROUPS]], not[subclass[x, y]]] == True

In[42]:= (% /. {x → x_, y → y_}) /. Equal → SetDelayed
```

Lemma. If the identity element **e[y]** of a group **y** belongs to a class **z**, then the union of the cosets for element of **z** is **range[y]**.

```
In[44]:= SubstTest[implies, equal[y, gp[t]], or[equal[image[y, cart[range[y], z]], range[y]],
            not[member[e[y], z]]], t → y] // Reverse // MapNotNot

Out[44]= or[equal[image[y, cart[range[y], z]], range[y]],
          not[member[y, GROUPS]], not[member[e[y], z]]] == True

In[45]:= (% /. {y → y_, z → z_}) /. Equal → SetDelayed
```

Lagrange's Theorem. If **x** is a subgroup of a finite group **y**, then the cardinality of **range[y]** is equal to the product of the cardinality of **range[x]** and the cardinality of the class of left cosets of **x** in **y**.

```
In[46]:= Map[not, SubstTest[and, implies[p1, p2], implies[and[p1, p2], p3],
           implies[p1, p4], implies[and[p3, p4], p5], not[implies[p1, p5]],
           {p1 -> and[member[x, GROUPS], subclass[x, y], member[y, GROUPS], member[y, FINITE]],
            p2 → member[e[y], range[x]],
            p3 → equal[image[y, cart[range[y], range[x]]], range[y]],
            p4 -> equal[card[image[y, cart[range[y], range[x]]]],
              natmul[card[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
                 range[y]]], card[range[x]]]], p5 -> equal[card[range[y]],
              natmul[card[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
                 range[y]]], card[range[x]]]]}]] // Reverse

Out[46]= or[equal[card[range[y]],
           natmul[card[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
              range[y]]], card[range[x]]]], not[member[x, GROUPS]],
          not[member[y, FINITE]], not[member[y, GROUPS]], not[subclass[x, y]]] == True
```

```
In[48]:= or[equal[card[range[y_]],
            natmul[card[image[VERTSECT[composite[y_, id[cart[V, range[x_]]], inverse[FIRST]]],
               range[y_]]], card[range[x_]]]], not[member[x_, GROUPS]],
          not[member[y_, FINITE]], not[member[y_, GROUPS]], not[subclass[x_, y_]]] := True
```

Lemma.

```
In[50]:= SubstTest[implies, and[member[w, omega], equal[w, natmul[x, z]]],
          member[pair[x, w], DIV], w → card[y]] // Reverse

Out[50]= or[member[pair[x, card[y]], DIV],
          not[equal[card[y], natmul[x, z]]], not[member[y, FINITE]]] == True

In[51]:= or[member[pair[x_, card[y_]], DIV],
          not[equal[card[y_], natmul[x_, z_]]], not[member[y_, FINITE]]] := True
```

The cardinality of the range of a group is called the **order** of the group.

Corollary. The order of any subgroup **x** of a finite group **y** divides the order of the group.

```
In[52]:= Map[not, SubstTest[and, implies[p1, p2],
            implies[p1, p3], implies[and[p2, p3], p4], not[implies[p1, p4]],
           {p1 -> and[member[x, GROUPS], subclass[x, y], member[y, GROUPS], member[y, FINITE]],
            p2 -> equal[card[range[y]],
               natmul[card[image[VERTSECT[composite[y, id[cart[V, range[x]]], inverse[FIRST]]],
                  range[y]]], card[range[x]]]], p3 → member[range[y], FINITE],
            p4 → member[pair[card[range[x]], card[range[y]]], DIV]}]] // Reverse

Out[52]= or[member[pair[card[range[x]], card[range[y]]], DIV], not[member[x, GROUPS]],
          not[member[y, FINITE]], not[member[y, GROUPS]], not[subclass[x, y]]] == True

In[53]:= or[member[pair[card[range[x_]], card[range[y_]]], DIV], not[member[x_, GROUPS]],
          not[member[y_, FINITE]], not[member[y_, GROUPS]], not[subclass[x_, y_]]] := True
```