CHAPTER 12

Lagrange's theorem and the logic of intensions

1 Lagrange's theorem

There is an interesting problem concerning Lagrange's theorem that I shall show is connected to the halting problem. Let us first examine a proof of Lagrange's theorem.

Definition, left coset

Let G be a group and H be a subgroup of G: $H \le G$. For each element $g \in G$ and $h \in H$ form the element gh, which, by the closure of G as a group, is an element of G. Let $gH = \{gh : h \in H\}$ represent the set of every element of G formed by taking a fixed element $g \in G$ and combining it systematically with every distinct element $h \in H$.

Equivalence of cosets

The criterion for when $xH = yH$ is given by: If $x^{-1}y \in H$ then $xH = yH$. An equivalent to this is given by: If $y^{-1}x \in H$ then $xH = yH$

Proof

First we show: If $x^{-1}y \in H$ then $y^{-1}x \in H$

Let	$x^{-1}y \in H.$	
Then	$\left(x^{-1}y\right)^{-1} \in H$	[By the existence of inverses, since H is a group]
Then	$y^{-1}\left(x^{-1}\right)^{-1} \in H$	
Then	$y^{-1}x \in H$	

We will now prove in general that: If $x^{-1}y \in H$ then $xH = yH$

Suppose	$f \in xH$
Then	$f = xh$ for some $h \in H$
Then	$f = yy^{-1}xh$
But	$y^{-1}x \in H$
That is	$h' = y^{-1}x$
Therefore,	$f = yh'h,$ where $h, h' \in H$
Therefore,	$f \in yH$

Likewise	If $f \in yH$
Then	$f = yh,\ h \in H$
	$f = xx^{-1}yh = xh''h,$ where $h, h'' \in H$
Therefore	$f \in xH$
This shows	If $x^{-1}y \in H$ then $xH \subseteq yH$ and $yH \subseteq xH$
Therefore	$xH = yH$.

Partitions

The cosets of H in G form a partition of G. What this means is that if two cosets of H in G are not identical then they do not share any element in common. The proof of this is by contradiction.

Proof

Suppose

$xH \neq yH$

are two cosets of H in G, but that they share at least one element in common. Let this common element be t. That is

$t \in xH$ and $t \in yH$

Therefore,

$t = xh$ and $t = yh'$, where $h, h' \in H$.

Therefore,

$xh = yh'$

$xh(h')^{-1} = y$

$h(h')^{-1} = x^{-1}y$

That is,

$x^{-1}y = h(h')^{-1}$

Therefore,

$x^{-1}y \in H$ since $h(h')^{-1} \in H$

But we just showed that if $x^{-1}y \in H$ then $xH = yH$.

Hence, $xH = yH$

which contradicts $xH \neq yH$.

Thus, if two cosets of H in G share an element in common, then they must be completely identical. Hence, the cosets of H in G partition G. This means that every element of G is in one, and only one, coset of H in G. Hence, the number of elements of each coset of H in G is the same. That is, if xH is a coset of H in G then $|xH| = |H|$. This means that their orders are the same.

Lagrange's theorem

If G is a finite group and H is a subgroup of G, then the order of H divides the order of G.

Formally: If H is a subgoup of the finite group G, then $|H| \mid |G|$, where ... | ... stands for "divides into".

Outline of the proof

The outline of the proof is as follows: Let H be a subgroup of G. That is $H \leq G$. Then the cosets of H in G partition (divide up) G in such a way that

(1) Each coset has exactly the same number of distinct elements as H.

(2) Every element of G is in one and only one coset of H.

Hence, $|G| = (\text{the number of cosets of } H \text{ in } G) \times |H|$

(The order of G is equal to the product of the number of cosets of H in G, and the order of H.) which means that the order of any subgroup of G must divide the order of G.

Proof of Lagrange's theorem

Each coset is formed by taking an element g of G and combining it with each distinct element h of H. For each distinct h in H we get a different element gh in G. Indeed, if $g^{-1}(gh) = g^{-1}(gh')$ then $gh = gh'$ and thus $h = h'$ follows. Hence, there is a one-one correspondence between elements of H and elements of any coset xH of H in G. Further, G is divided into a finite number of cosets xH of H in G. Thus

$|G| = (\text{the number of cosets of } H \text{ in } G) \times |H|$

(The order of G is equal to the product of the number of cosets of H in G, and the order of H.) That is,

$|H| \mid |G|$

The order of H divides the order of G, which proves the theorem.

2 The problem

Beeson writes about Lagrange's theorem: -

> A test problem here is the theorem of LaGrange in group theory, according to which the coset of a subgroup of a finite group all have the same number of elements, i.e., are in one-to-one correspondence. The proof of this theorem is very simple by ordinary mathematical standards, yet it seems to be too difficult for automated deduction; and the bottleneck seems to be that several different data types are involved." (Beeson [1988] p.212)

He lists these data types: -

1. *G*, The type of group elements.
2. $P(G)$ The type of subsets of *G*
3. *C* The type of functions from one subset of *G* to another
   The one-one correspondence of cosets is of this type.
4. "one needs either an operation leading from a subgroup *H* and element *a* of *G* to the coset *aG*, or one needs an operation leading from *H* to the type of cosets of *H*.".

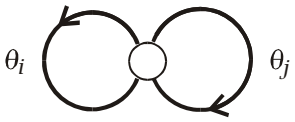
This was written in 1988 and I have no idea whether the programmers since believe they have made progress with it. I will show that all they can produce is an apparent simulation of the theorem on a *glorified typewriter* [1/2.3]. To do so, let me begin by revisiting the proof of the solution to the halting problem by the method of exits. [Theorem 2.2 proven in section 6 above.]

> Proof of the solution to the halting problem by the method of exits
> The method of exits works by tracing back information from every terminus through the machine $T_{n+1} = T_n + Q_{n+1}$. In doing so we work around each loop in T_{n+1}, recording any 1-loop in it by an asterisk and each longer loop by a bar symbol. These encode the possibility of a finite repetition of a configuration leading to an exit (halting configuration) as well as identifying the infinitely recurring non-halting configurations. The problem is finite if the period of the maximal cycle in T_{n+1} is finite. But if the period of the maximal cycle in T_n is finite then the addition of Q_{n+1} adds a finite number of loops to the maximal cycle; and the resultant maximal cycle and its period remain finite. Therefore, the problem can be solved by the method of exits for T_{n+1}. [3.4.5 and Chap. 3 Sec. 6]

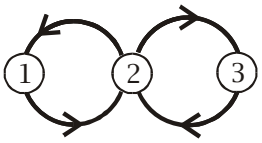
The essence of this proof lies on a property of a decomposition of every maximal cycle in a machine. This was result 4.9 of Chapter 3: -

> Chap.3 / 4.9 (+) Result, decomposition of cycles
> Every cycle is may be decomposed into indecomposable loops. The period a cycle is the sum of the period of its indecomposable loops.

In essence this result arises from the finite decomposition of finite cycles. Ultimately, any indecomposable cycle may be separated into two loops as follows: -



This process may be iterated. Denote the sequence of the first loop by '1', the state through which they are joined by '2' and the remaining sequence in the second loop by '3'. Then the join of two loops is symbolically represented by: -



The composition of the two loops to produce a new maximal cycle has the form: -

$(1\ 2)(2\ 3) = (1\ 2'\ 3\ 2)$

where I am using the notation of permutation groups. The symbol '$2'$' represents the fact that in the maximal cycle we must travel through this linking state twice. Now compare this with the regular composition of permutations using the same symbolism: -

$(1\ 2)(2\ 3) = (1\ 3\ 2)$

There is an immediate correspondence between the two compositions: the machine cycle $(1\ 2'\ 3\ 2)$ corresponds one-one to the permutation $(1\ 3\ 2)$. We have the result: -

> **Result, equivalence of compositions**
> A cycle in a machine may be decomposed iff the corresponding permutation may be decomposed.

The proof of the solution to the halting problem by the method of exits given above adds another result: -

> **Result, equivalence**
> The halting problem is soluble *T* iff any cycle in a machine may be decomposed into indecomposable finite cycles in finitely many steps.

The halting problem is soluble, just as any permutation can be decomposed into 2-cycles: -

> **Result, permutations**
> Every permutation is the product of its cycles. (Equivalently, every permutation can be uniquely expressed as a product of disjoint cycles.) Every permutation is a product of 2-cycles. (For a proof see Herstein [1975] p.78)

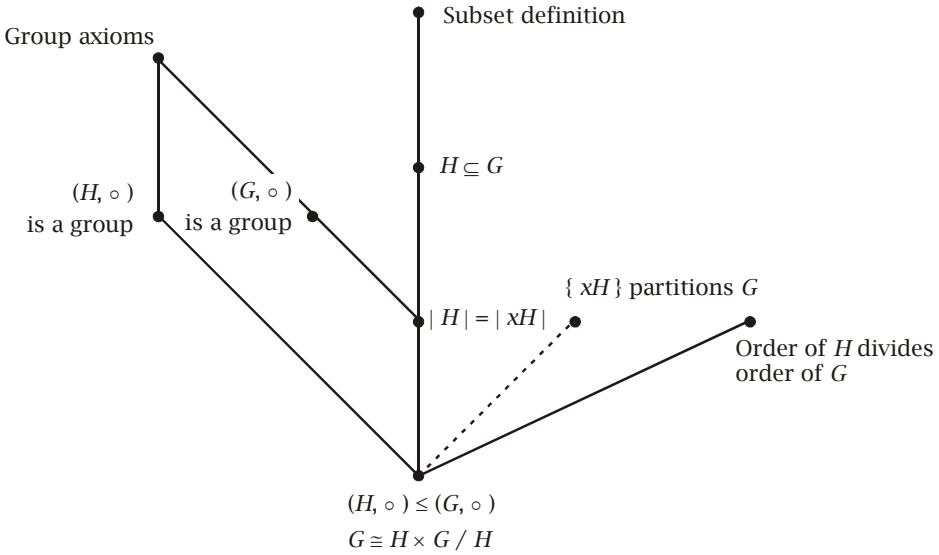
We have also: -

Cayley's theorem

Let S_n be the symmetric group on n elements. Any finite group is isomorphic to a subgroup of S_n for some n. (For a proof see Herstein [1975] p.71)

On examining Lagrange's theorem in depth we may see that the underlying "fact" on which is built is already expressed in the notion of a subgroup: $H \leq G$. Lagrange's theorem exposes a division property of this relation – it shows that the order of the subgroup H divides into the order of the group G. Now this would not be possible unless G was the product of H and some other group – so that the underling "fact" on which the theorem is constructed is the decomposition of groups into cyclic subgroups. The statement, $G \cong H \times \frac{G}{H}$ is in the language of isomorphisms, and also introduces the symbol for a quotient group; Lagrange's theorem is so much more than just the division relation; it is the statement that $G \cong H \times \frac{G}{H}$ where H and $\frac{G}{H}$ are both groups. This is the same fact that underpins Cayley's theorem and finds expression in the notion that every group is isomorphic to some subgroup of a permutation group. This explains the underlying problem that the programmers were having with the proof of Lagrange's theorem, because of a very deep link between it and the solution to the halting problem.

It is interesting to investigate how the proof of Lagrange's theorem does "work". The following is the lattice structure of the proof of Lagrange's theorem: -



In this diagram inference goes up the page. Conjunctions proceed down. My analysis of Lagrange's theorem is as follows. $H \leq G$ stands for H is a subgroup of G. The relation of $H \leq G$ I take to be the conjunction of

α_1 H is a group

α_2 G is a group

α_3 $H \subseteq G$

There is actually some ambiguity as to what the statement of Lagrange's theorem is. Generally, it is taken as: If $H \le G$ then the order of H divides the order of G. As it happens Beeson takes it as the statement on cosets $|H| = |xH|$ and one might also interpret it as the statement $G \cong H \times \frac{G}{H}$. In the proof as I have it, there are two major intermediary steps.

1. A proof that the set of all left cosets xH partitions G.
2. A proof of $|H| = |xH|$

Then Lagrange's theorem follows from the conjunction of these. To say that G is partitioned into a collection of sets is to say that every element of G is in one, and only one of the partitions. This is not sufficient to prove Lagrange's theorem because, on that basis alone, the partition could be uneven, and no result on the order of the partitions would follow. So we need to add to that the statement that the partitions are all of the same size; that is $|H| = |xH|$. The statement, $|H| = |xH|$, does not appear to require that H is a group. It does require that G is a group because the argument depends on combining each element of H with a fixed element of G and we require the group properties of G to prove that each element $h \in H$ corresponds one for one with an element $xh \in xH$ where $x \in G$.

The lattice diagram reveals the *prima facie* problem with any attempt to formalize Lagrange's theorem. According to this diagram, Lagrange's theorem, $G \cong H \times \frac{G}{H}$ is formally equivalent to the statement $H \le G$; so the inference does not take place in the analytic logic built over the lattice. It is an equivalence of names for the same lattice point, and one that translates one language, $H \le G$ (groups, subgroups, order) into another $G \cong H \times \frac{G}{H}$ (isomorphisms, quotient groups), or yet another $|G| \cong |H| \times |xH|$ (arithmetic, number theory, modulus, rings). So what we are looking at is equivalence of languages constructed what is *prima facie* the logic of intentions; it is an inference whose whole purpose is to establish the equivalence of different conceptual ways of looking at an object's (a group G's) internal structure, where $H \le G$ *already embodies the fact that is said to be is consequence* – namely Lagrange's theorem. On a similar line of thought, I observe that the intermediary step: -

1. Set collection $\{xH\}$ partitions G

is a statement that *could* in the theoretical sense be distinct as a lattice point from another assumption from which it follows, but in fact, in the context of the specific assumption, $H \le G$, cannot be so distinguished. In other words, it is a kind of fictional lattice point, whose purpose in the logic of intensions is to be recombined (formation of meet) with the lattice

point $|H| = |xH|$ to return to $H \leq G$ but under a different description, $G \cong H \times \frac{G}{H}$. Once again, the whole argument seems to have more to do with sense and reference than with formal, analytic logic as such.

The point that the programmers were stuck on, according to Beeson, is the proof of $|H| = |xH|$. Here is my version of that: -

> Each coset is formed by taking an element g of G and combining it with each distinct element h of H. For each distinct h in H we get a different element gh in G. Indeed, if $gh = gh'$ then $g^{-1}(gh) = g^{-1}(gh')$ and thus $h = h'$ follows. Hence, there is a one-one correspondence between elements of H and elements of any coset xH of H in G.

(Here H does not need to be a subgroup, but could just be a subset.) The proof works by pairing off elements of H with elements of xH and hence appears to be a fundamental use of the pigeon-hole principle [Chap.15 Sec.3]. All groups are ultimately products of cycles, and for simplicity, suppose that H is a simple cyclic group; then $H = \{h, h^2, \ldots, h^{n-1}, 1\}$ so when we are pairing off the elements of H with the elements of xH we are literally going through the cycle or loop, and hence the application of the pigeon-hole principle. Once we have finished one loop of H we must start over and start filling the "boxes" in xH a second time – loop for loop. This suggests that the presence of loops (modules) within the Boolean lattice (implicit loops) embeds the whole of ring theory into the lattice and makes the lattice incomplete as a system of inference. I note that in programs it is the loops that cause non-halting. I remark also that Cantorian anti-diagonalisation argument is a species of the pigeon-hole principle applied to infinite sets. We cannot fit all of one set into another; or if we take the bigger set and start to fill up the "boxes" of the smaller set, then we have something left over and have to start again.

Every group can be embedded in a permutation group. A permutation is based on a loop, and then the loop can be taken in a clockwise or anticlockwise direction; finally, individual pairs of the loop can be inverted to obtain another loop. I conjecture that it is because of these properties of permutation groups that transfinite ordinals express order invariance; the order in which the lattice points are taken is constantly being juggled. The root cause is the presence of the loops in the system. These are conjectures. I note also that the order or the sizes of the subgroups is not determined by $H \leq G$. That is, we may have in order of size $1 \leq H \leq \frac{G}{H} \leq G$ but also $1 \leq \frac{G}{H} \leq H \leq G$. Also $H = \frac{G}{H}$ is possible. So there is constant juggling of the sub-rings generated by the lattice that I am predicting is the underlying source of order inversions.