Chapter 5. Cayley Graph and Defining Relations

This chapter looks at an interesting graphical representation of a small group. The representation is called the Cayley graph and may be considered to be an abbreviated multiplication table for the group. The representation is used to determine a set of defining relations for the group. These are the laws or axioms obeyed by the group's generators (beyond the axioms obeyed by all groups) that specify the particular group. The defining relations provide a connection between small groups and the combinatorial group algorithms.

Definitions

For a group given by a set of generators, a *Cayley graph* represents the effects of multiplying an element of a group by a generator of the group. For each element there is a node of the graph. From each node and for each generator there is a directed edge, labelled by the generator, that leads to the node corresponding to the product of the element and the generator.

For example, in the Cayley graph of the symmetries of the square, where the generators are $a = elt[2]$ and $b = elt[5]$, there are edges

$$elt[5] \xrightarrow[a]{} elt[8]$$

$$elt[5] \xrightarrow[b]{} elt[1]$$

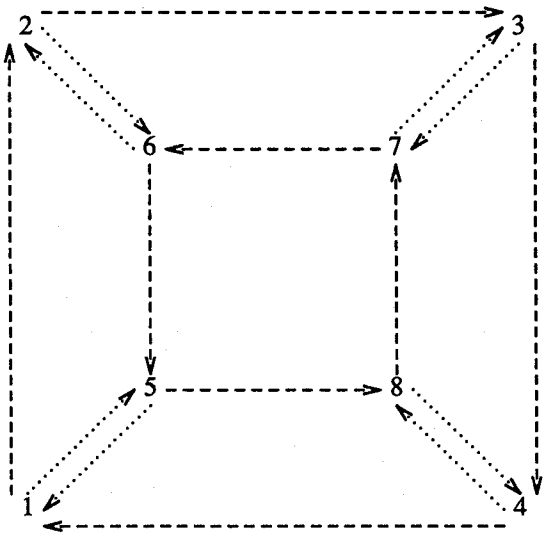
since $elt[5] \times elt[2] = elt[8]$, and $elt[5] \times elt[5] = elt[1]$. The complete Cayley graph for this example is given in Figure 1. The nodes are integers corresponding to the elements $elt[1]$ to $elt[8]$.

A *word* in the generators is a product

$$s_{i_1} \times s_{i_2} \times \cdots \times s_{i_m}$$

where each term is a generator, or the inverse of a generator. Some examples are a, b, b^2, $a \times b$, a^6, and $a \times b \times a \times b \times a^{-1}$. A word is a formal (or symbolic) product. When a word is evaluated we get an element of the group.

Figure 1 : Cayley Graph of Symmetries of the Square



The dashed arrows represent edges labelled by a.
The dotted arrows represent edges labelled by b.

In the group of symmetries of the square,

a	evaluates to	*elt* [2],	
b	evaluates to	*elt* [5],	
b^2	evaluates to	*elt* [1],	
$a \times b$	evaluates to	*elt* [6],	
a^6	evaluates to	*elt* [3],	and
$a \times b \times a \times b \times a^{-1}$	evaluates to	*elt* [4].	

There are an infinite number of words that evaluate to any given element.

The inverse of a word is

$$s_{i_m}^{-1} \times s_{i_{m-1}}^{-1} \times \cdots \times s_{i_1}^{-1}.$$

It evaluates to the inverse of the element to which the word evaluates.

By tracing words through the Cayley graph we can multiply elements. To form *elt* [7] $\times$ *elt* [3] we require a word that evaluates to *elt* [3]. Such a word is a^6. Starting at node 7 we trace a path along edges whose labels correspond to the terms in the word. As shown in Figure 2, this leads to the product of *elt* [7] and *elt* [3] being *elt* [5].

Figure 2 : Tracing as Multiplication

7 ------> 6 ------> 5 ------> 8 ------> 7 ------> 6 ------> 5

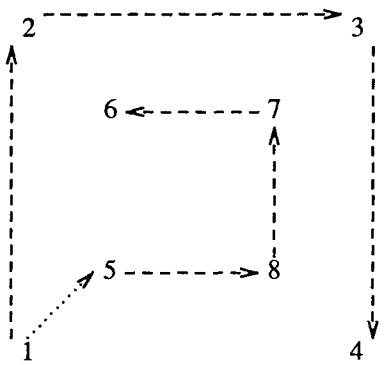
If the word has a term which is the inverse of a generator then follow the edge labelled with that generator, but in the reverse direction. Thus tracing $a^{-1} \times b \times a^{-1}$ from node 7 proceeds to node 3 as shown in Figure 3.

Figure 3 : Tracing Inverses

7 <-------- 8 ·····> 4 <-------- 3

We can determine a word for *elt* [3] by finding a path from the identity (node 1) to node 3. To find a word for each element it is useful to have a spanning tree of the Cayley graph. One spanning tree for the above Cayley graph is given in Figure 4.

Figure 4 : Spanning Tree



This gives a unique word for each element of the group. For this example the correspondence between elements and words is:

element 1	corresponds to	empty word,	
element 2	corresponds to	a,	
element 3	corresponds to	a^2,	
element 4	corresponds to	a^3,	
element 5	corresponds to	b,	
element 6	corresponds to	$b \times a^3$,	
element 7	corresponds to	$b \times a^2$,	and
element 8	corresponds to	$b \times a$.	

A *relation* is a word corresponding to the identity element. Some examples in this case are a^4, a^8, a^{12}, b^2, $a^4 \times b^2$, b^4, and $b \times a \times b \times a$. The relations corresponding to the group axioms are called *trivial relations*. They hold for every group. A relation corresponds to a loop in the Cayley graph that begins and ends at the identity node.

Note that the relations in the above example are not independent. A set of relations is called a set of *defining relations* for the group if every (nontrivial) relation for the group can be deduced from the set of relations.

We can always deduce the trivial relations from a set of relations. The set of deductions is closed under products, inverses, and conjugates of the form $w \times r \times w^{-1}$, for some word w and deduction r.

Viewing relations as loops about the identity, we see that a product of relations is just a concatenation of loops; an inverse of a relation is the loop in the reverse direction; and a conjugate $w \times r \times w^{-1}$ corresponds to tracing the word w to a node i, looping around i with the relation r, and then returning from i to the identity by tracing w^{-1}.

Start with a Spanning Tree

The number of loops around the identity is infinite, so taking all relations of a group and attempting to eliminate deductions is not a feasible approach for determining a set of defining relations. This section aims to reduce the number of loops/relations that must be considered. The role of the spanning tree is evident from the following theorem.

Loop Basis Theorem

Suppose a group G is defined by a set of generators and a Cayley graph. Given a spanning tree T of the Cayley graph, then a set of defining relations for the group is formed by the set

$$R(T) = \{\, R(e) \mid e \text{ is an edge not in } T \,\}.$$

The relation $R(e)$ for the edge

$$e : i \xrightarrow{s} j$$

is the word $w_i \times s \times w_j^{-1}$, where w_i is the word/path in the spanning tree from node 1 to node i.

Consider the example of the symmetries of the square for which we already have a Cayley graph and a spanning tree. The set of defining relations $R(T)$ we obtain is

$$\begin{aligned}
R(2..b..>6) &= (a) \times b \times (a^{-3} \times b^{-1}) \\
R(3..b..>7) &= (a^2) \times b \times (a^{-2} \times b^{-1}) \\
R(4--a-->1) &= (a^3) \times a = a^4 \\
R(4..b..>8) &= (a^3) \times b \times (a^{-1} \times b^{-1}) \\
R(5..b..>1) &= (b) \times b = b^2 \\
R(6..b..>2) &= (b \times a^3) \times b \times (a^{-1}) \\
R(6--a-->5) &= (b \times a^3) \times a \times (b^{-1}) \\
R(7..b..>3) &= (b \times a^2) \times b \times (a^{-2}) \\
R(8..b..>4) &= (b \times a) \times b \times (a^{-3})
\end{aligned}$$

Note that the relation $R(e)$ is defined even for edges in the spanning tree. In this case, the loop lies wholly within the spanning tree and the relation is a trivial relation.

To prove the theorem we must show that any relation/loop is deducible from the set $R(T)$. Take any loop

$$id = j_0 - s_{i_1} \rightarrow j_1 - s_{i_2} \rightarrow j_2 \cdots j_{m-1} - s_{i_m} \rightarrow j_m = id$$

about the identity, corresponding to the relation

$$s_{i_1} \times s_{i_2} \times \cdots \times s_{i_m} = identity.$$

At each node visited in the loop we can insert a path through the spanning tree to the identity node, and back again. Each of these is the insertion of a trivial relation. The resulting loop is the concatenation of the loops $R(j_{k-1} - s_{i_k} \rightarrow j_k)$. These are either in $R(T)$, or are themselves trivial relations. Hence, all relations are deducible from $R(T)$.

The size of $R(T)$ is the number of edges not in the spanning tree. If S is the set of generators, then the size of $R(T)$ is

$$1 + |G| \times \Big[|S| - 1\Big].$$

This is still often larger than is necessary so we will look at ways of choosing a subset of $R(T)$ from which we can deduce the remaining relations in $R(T)$.

The Colouring Algorithm

The colouring algorithm determines a subset of $R(T)$ that is a set of defining relations. The idea behind the algorithm is to colour those edges e for which $R(e)$ can be deduced from the current set of relations. Initially the edges of the spanning tree are coloured because they correspond to the trivial relations. When a new relation of $R(T)$ is added to the subset then we determine all deductions from the subset and colour the appropriate edges. A relation of $R(T)$ is added only if it currently corresponds to an uncoloured edge.

The heart of the algorithm is determining the deductions. For a single relation r in the subset we can deduce $w_i \times r \times w_i^{-1}$ for all nodes i in the graph. This is the loop r around node i. If this loop around i contains precisely one uncoloured edge e then we can colour the edge. The argument for doing this is the same as in the proof of the Loop Basis Theorem. The loop about node i can be considered as a concatenation of loops $R(f)$. All the edges f (except e) are coloured, so the relations can be deduced from the subset. But the whole loop can also be deduced from the subset. Hence $R(e)$ can be expressed as a product involving $w_i \times r \times w_i^{-1}$ and the other $R(f)$ involved in the loop, all of which are deductions from the subset. Hence $R(e)$ is deducible from the subset.

To determine all the deductions from the subset we repeatedly trace single relations in the subset about the nodes of the graph colouring edges where possible until we can colour no more edges. The simplest strategy for determining the deductions is illustrated in Algorithm 1.

Algorithm 1 : Defining Relations by Colouring Cayley Graph

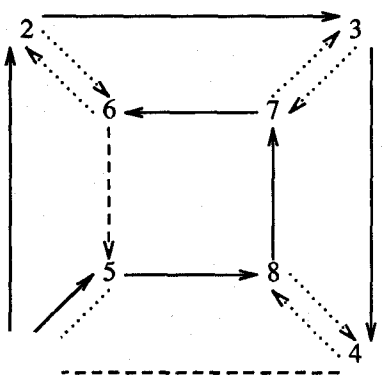
Input : a group G given by a set S of generators and a Cayley graph;

Output : a set of defining relations;
begin
construct a spanning tree of the Cayley graph;
colour the edges of the spanning tree;
defining_relations := empty;
while there are uncoloured edges **do**
choose an uncoloured edge e; add $R(e)$ to *defining_relations*; colour e;
repeat (*determine all deductions*)
for each relation r in *defining_relations* **do**
for each node i **do**
trace r around i;
if loop contains precisely one uncoloured edge f **then**
colour f;
end if;
end for;
end for;
until no more edges can be coloured;
end while;
end;

Example

We will now work through the algorithm for the symmetries of the square, using our previous spanning tree. The uncoloured edges will be omitted, so we can see the effect of the colouring. We begin with the spanning tree, as shown in Figure 5.

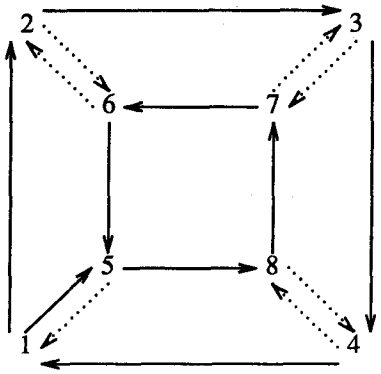
Figure 5 : Initially Coloured Edges



Solid arrow represent coloured edges.

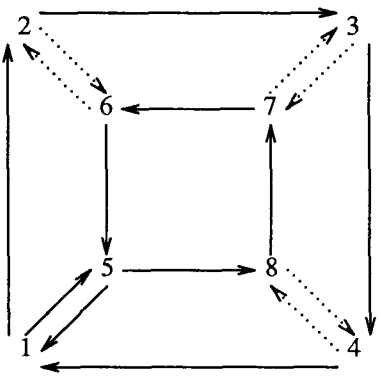
The edge 4 -*a*-> 1 is chosen, and gives the relation a^4. This is the first defining relation. Tracing a^4 around node 5 colours the edge 6 -*a*-> 5. Thus giving Figure 6.

Figure 6 : Coloured Edges after First Defining Relation



The edge 5 -*b*-> 1 is chosen, and gives the relation b^2. This is added to the set of defining relations. Tracing colours no further edges, and we have Figure 7.

Figure 7 : Coloured Edges after Second Defining Relation



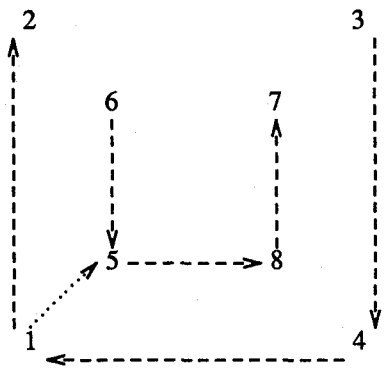
The edge 2 -*b*-> 6 is chosen, and gives the relation $a \times b \times a^{-3} \times b^{-1}$. This is added to the defining relations.
Tracing b^2 around node 2 colours the edge 6 -*b*-> 2.
Tracing $a \times b \times a^{-3} \times b^{-1}$ around node 2 colours the edge 3 -*b*-> 7.
Tracing b^2 around node 3 colours the edge 7 -*b*-> 3.
Tracing $a \times b \times a^{-3} \times b^{-1}$ around node 3 colours the edge 4 -*b*-> 8.
Tracing b^2 around node 4 colours the edge 8 -*b*-> 4.

The graph is completely coloured and the set of defining relations is { a^4, b^2, $a \times b \times a^{-3} \times b^{-1}$ }.

Another Example

In order to obtain short relations we should take a minimal spanning tree such as that shown in Figure 8.

Figure 8 : Minimal Spanning Tree



Working through this example we obtain the defining relations $\{ a^4, b^2, a \times b \times a \times b^{-1} \}$.

Analysis

The total cost of constructing a set of defining relations comprises three components - the cost of constructing the Cayley graph, the cost of constructing the spanning tree, and the cost of the colouring algorithm. Assuming we have a list of elements of G, the cost of constructing a Cayley graph is

$|G| \times |S|$ multiplications, and

$|G| \times |S|$ searches.

Forming the spanning tree is a breadth-first traversal of the edges of the Cayley graph. The cost of this is at most

$|G| \times |S|$ edge traces.

Let L be the sum of the lengths of the defining relations, as constructed by the colouring algorithm. The actual formation of the relations $R(e)$ that make up the set of defining relations requires

L edge traces.

Now let us consider the cost of determining the deductions. Each iteration of the repeat-loop arises from the colouring of at least one edge not in the spanning tree. Hence the number of iterations is at most

$$1 + |G| \times \left\lfloor |S| - 1 \right\rfloor.$$

Each iteration traces the (current) defining relations around all the nodes. This can be no worst than tracing all the defining relations around the nodes, so the total number of edge

traces required to determine the deductions is bounded by

$$L \times |G| \times \left[1 + |G| \times \left[|S| - 1 \right] \right].$$

Thus the total worst case cost is proportional to $L \times |G|^2 \times |S|$.

Summary

The Cayley graph is a useful graphical representation of a small group. The relations of the group correspond to loops in the Cayley graph. The colouring algorithm is an effective means of determining deductions from a set of relations, and hence of determining a set of defining relations of the group.

The strategy we have followed for the colouring is the simplest one. It is also the one used in practical implementations. Its worst case cost is dominated by $L \times |G|^2 \times |S|$, where L is the total length of the defining relations.

Exercises

(1/Easy) Not all Cayley graphs are planar, so for the next example we can not nicely draw the graph. Instead we give the abbreviated multiplication table for the group. The entries indicate the product of the row number by the generator at the column heading.

	a	*b*
1	2	7
2	3	6
3	4	5
4	1	8
5	6	1
6	7	4
7	8	3
8	5	2

Thus there are edges 1 -*a*-> 2 and 1 -*b*-> 7 from node 1.

Work through the algorithm for this example. Choose several different spanning trees, if you so desire, and compare the sets of defining relations.

(2/Difficult) The simple colouring strategy often traces relations when there is no possibility for colouring an edge. For example, the loop may already be totally coloured, or, having just traced the loop in the previous iteration and found two uncoloured edges, the strategy would trace it again even if no edges at all have been coloured in the meantime.

The strategy you should analyse follows. For each pair (node, defining relation) keep track of the number of uncoloured edges in the loop. This information is initialised when a defining relation is added to the set by tracing it around all nodes. At other times only trace an edge if there is a possibility of making a deduction and colouring an edge.

Let N be the number of uncoloured edges in the graph. Initially N is $1 + |G| \times \left(|S| - 1\right)$. If t edges are uncoloured in a loop just traced, then store the integer

0, if $t = 0$, and

$N - t + 1$, if $t \neq 0$.

with the pair (node, defining relation) that corresponds to the loop. The loop is next traced when N is less than or equal to the integer stored with the corresponding pair.

Analyse this strategy.

(3/Moderate) This strategy considers the edges rather than the nodes when tracing relations. A loop will only lead to the colouring of an edge if some other edge in the loop has just been coloured. Keep a list of edges (not in the spanning tree) that have been coloured. Add newly coloured edges to the end of the list.

For each edge in the list, where s is the label of that edge, trace each relation around the edge by matching the edge with each "essentially different" occurrence of s in the relation. For example, the generator a occurs four times in a^4, but each occurrence is the same from the viewpoint of tracing a loop.

If the defining relations are considered to be nearly random, then the average number of essentially different occurrences of a given generator in a set of total length L is $L/|S|$. With this assumption, analyse the strategy to show it is proportional to $L \times |G|$.

Bibliographical Remarks

The use of Cayley graphs to represent groups is widespread in combinatorial group theory. The correspondence between loops and relations is also well known. The classic books H.S.M. Coxeter and W.O.J. Moser, **Generators and Relations for Discrete Groups**, Springer-Verlag, Berlin, 1965, and W. Magnus, A. Karass, and D. Solitar, **Combinatorial Group Theory**, Interscience, New York, 1966 contain much more on Cayley graphs and relations.

The construction of a set of defining relations from a Cayley graph is due to J. J. Cannon, "*Construction of defining relations for finite groups*", Discrete Mathematics **5** (1973) 105-129. This chapter only deals with the single stage algorithm of that paper, and not with the two stage algorithm. While Cannon suggests in his paper that a colouring strategy based on tracing relations around the edges that become coloured may be better than the simplest strategy of tracing relations around nodes, only the simplest strategy has been implemented. Presumably, the simplest strategy is adequate in practice.

It should be noted that the analysis in Cannon's paper is simplified to the point of assuming only a small number of iterations of the **repeat**-loop each time a new defining relation is added to the set.

The paper, John Grover, Lawrence A. Rowe, and Darrell Wilson, "*Applications of coset enumeration*", (Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation, Los Angeles, 1971) (S. R. Petrick, editor), Association of Computing Machinery, New York, 1971, 183-187, describes an interactive approach to constructing defining relations. It constructs a relation $w_1 \times w_2^{-1}$ if the words w_1 and w_2 evaluate to the

same element of the group. A coset enumeration is used to check whether the set of relations is defining. A variation of this approach is described in C. M. Campbell and E. F. Robertson, "*Presentations for the simple groups G,* $10^5 < |G| < 10^6$", Communications in Algebra **12**(21) (1984) 2643-2663.

The Schreier-Todd-Coxeter-Sims method briefly mentioned in chapter 13 and described in J. S. Leon, "*On an algorithm for finding a base and strong generating set for a group given by generating permutations*", Mathematics of Computation **35**, 151 (1980) 941-974 determines defining relations for very large permutation groups.