Let $G$ be a group of permutations of a finite set. For definiteness, we can take $G$ to be the group of permutations of $\{1, 2, 3, 4, 5\}$ generated by the 4-cycles $a := (1 \ 2 \ 4 \ 3)$ and $b := (1 \ 2 \ 5 \ 4)$. We will refer to this as our “running example” in what follows.

What is the order of $G$? Is $G$ solvable? Is $G$ nilpotent? What are the Sylow subgroups of $G$? Can we find normal forms for the elements of $G$ as words in the generators? Can we find a set of defining relations for $G$? Computational group theorists have developed algorithms for answering questions like these. Most such algorithms require first constructing a “base” and “strong generating set” for the given permutation group. The purpose of this handout is to give a very brief introduction to these concepts. For more information see the books by Holt [1], Seress [2], or Sims [3]. I will also be giving more details in class.

As a warm-up, we begin with a simple algorithm.

1. Orbits

Let $G$ be a group with a finite generating set $S$. Suppose $G$ acts on a set $X$ on the right. [Throughout this handout I will use right group actions and will define composition in permutation groups accordingly. This makes it easier for you to draw Schreier graphs as you read, which I encourage you to do. I will also use exponential notation for group actions.] Given $x \in X$, how would you compute the $G$-orbit of $x$? In our running example, you can see at a glance that the orbit of 1 (or any other element) is the entire set $X := \{1, 2, 3, 4, 5\}$. But if $X$ is large, we need a more systematic procedure.

We will build the orbit as a list $Y = \{y_1, \ldots, y_k\}$ without repetition. Start with $y_1 := x$. Now run through the generators $s \in S$ and adjoin $y_{s}^i$ to $Y$ if it is not already present. [By “adjoin”, I mean set $y_{s}^i := y_{s}^j$ for the first such $s$, then define $y_{s}^i$, and so on.] If nothing has been adjoined, stop. Otherwise, repeat the process with $y_1$ replaced by $y_2$, i.e., successively adjoin $y_{s}^i$ to $Y$ for each generator $s$ such that $y_{s}^i$ is not already in $Y$. Continue in this way until you reach an index $k$ such that $Y$ already contains all $y_{s}^k$; this must happen because $X$ is finite. Then $Y$ is the orbit. Indeed, $Y$ is obviously contained in the orbit and is $G$-invariant by construction, so it must be the whole orbit.

**Exercise 1.** Carry out the algorithm for our running example, with $x = 1$.

**Exercise 2.** The assertion that $Y$ is $G$-invariant makes tacit use of the assumption that $X$ is finite. Explain where this is used. How would you modify the procedure to work if $X$ is infinite (still assuming, for simplicity, that $G$ is finitely generated)?

2. Transversals

The orbit algorithm in the previous section not only computes the orbit $Y = x^G$, but it leads to a specific way of representing each $y \in Y$ as $y = x^t$ for some $t = t(y) \in G$. For when $y_i$ is first adjoined to $Y$, it is given as $y_{s}^j$ for some $j < i$ and $s \in S$; so we can set $t(y_i) := t(y_j)s$. To start the induction, take $t(y_1) := id.$
Note that the elements $t(y)$ form a right transversal for $G_x$ in $G$, where $G_x$ is the stabilizer of $x$. (A right transversal of a subgroup is a set of representatives for the right cosets.)

Remark. The construction of $t(y)$ actually gives a specific $S$-word representing $t(y)$. In other words, we have constructed for each $y \in Y$ a path $\gamma(y)$ from $x$ to $y$ in the Schreier graph.

Exercise 3. Carry out the algorithm for our running example, with $x = 1$. Draw the Schreier graph, and indicate the paths $\gamma(y)$. (They should form a tree, with edges directed away from the root 1.)

3. Stabilizers

Further analysis of the orbit algorithm leads to a computation of the stabilizer $G_x$ of $x$ in $G$. Suppose we are carrying out the algorithm and drawing the Schreier graph as we go along. At each step we have some $y \in Y$ and $s \in S$, and we ask whether $z := y^s$ is already in the part of the graph that we have constructed. If not, we adjoin $z$ and an edge $e$ (labeled by $s$) from $y$ to $z$, and we have $\gamma(z) = \gamma(y) * e$, where $*$ denotes path composition. Suppose, however, that $z$ was already in the graph. Then we can still draw an edge $e$ labeled by $s$ from $y$ to $z$, and we get a loop $\gamma(y) * e * \bar{\gamma(z)}$ based at $x$, where the bar denotes path inversion. This loop represents an element

$$h(y, s) := t(y)st(z)^{-1}$$

of the stabilizer $G_x$. The following result is due to Schreier:

**Theorem.** The elements $h(y, s)$ generate $G_x$.

See Exercise 6 and Appendix A for proofs. The elements $h(y, s)$ are called the Schreier generators of $G_x$. If $k := |Y|$ and $n := |S|$, then the number of Schreier generators is

$$kn - (k - 1) = kn - 1 + 1.$$

In practice, many of these generators are often redundant. In general, however, this upper bound on the number of generators of $G_x$ is sharp. See the remarks at the end of this section.

Exercise 4. Explain where the number $kn - (k - 1)$ above comes from.

Exercise 5. Carry out the algorithm for our running example, with $x = 1$. Eliminate redundant generators and give the simplest description you can of $G_x$.

Exercise 6. Prove Schreier’s theorem. [Hint: Let $H$ be the subgroup of $G$ generated by the elements $h(y, s)$. Since $H$ fixes $x$, there is a surjective $G$-set map $H \backslash G \rightarrow Y$. Construct a $G$-set map in the other direction. See Appendix A for a different proof.]

Note that if we are given a finitely-generated group $G$ and a subgroup $H$ of finite index, we can apply the procedure above to the natural action of $G$ on $H \backslash G$ to get a (finite) set of Schreier generators of $H$. If $G$ has $n$ generators and $H$ has index $k$, then we get $k(n - 1) + 1$ Schreier generators. If $G$ is the free group on $n$ generators, one can show that $H$ is free on $k(n - 1) + 1$ generators (Nielsen–Schreier theorem); so the upper bound $k(n - 1) + 1$ on the number of generators required by $H$ is sharp.
We will give an algebraic proof of the Nielsen–Schreier theorem in Appendix A. If you know a little algebraic topology, you might enjoy giving a topological proof. It is based on the following two facts: (1) If $X$ is a connected graph, then its fundamental group is free of rank $n$, where $n$ is defined by the equation
\[ \chi(X) = 1 - n. \]
Here $\chi(-)$ denotes the Euler characteristic. (2) If $X$ is a finite complex and $Y \to X$ is a $k$-fold covering map, then
\[ \chi(Y) = k \cdot \chi(X). \]

4. Order

If you have been doing the exercises involving our running example, you should know that the group in that example has order 20. [$G$ acts transitively on a 5-element set, and the stabilizer of 1 has order 4.] The same idea can be used to compute the order of any group of permutations of a finite set $X$.

Let $G$ be such a group, with a given set of generators. If $G$ is trivial, its order is 1. Otherwise, choose an element $x \in X$ that is not fixed by $G$. [It suffices to choose a nontrivial generator $s$ and then take $x$ that is not fixed by $s$.] Compute the $G$-orbit $Y$ of $x$ and the stabilizer $H$ of $x$. Then $H < G$, and
\[ |G| = |Y| \cdot |H|. \]
If $H$ is trivial, we are done. Otherwise, repeat the process with $G$ replaced by $H$. Continuing in this way, we obtain a sequence of elements $x_1, \ldots, x_k \in X$ and a strictly decreasing chain of subgroups
\[ G = G_1 > G_2 > \cdots > G_k > G_{k+1} = \{1\}, \]
such that $G_{i+1}$ is the stabilizer of $x_i$ in $G_i$ for $i = 1, \ldots, k$. If $Y_i$ is the $G_i$-orbit of $x_i$, then
\[ |G| = |Y_1| \cdot |Y_2| \cdots |Y_k|. \]
In our running example, we can take $x_1 = 1$ and $x_2 = 2$ (so $k = 2$). Then $Y_1 = \{1, 2, 3, 4, 5\}$, $G_2$ is generated by the 4-cycle $(2\ 3\ 5\ 4)$, and $Y_2 = \{2, 3, 4, 5\}$. Hence
\[ |G| = |Y_1| \cdot |Y_2| = 5 \cdot 4 = 20. \]

5. Bases and strong generating sets

The set $\{x_1, \ldots, x_k\}$ constructed in the previous section has the property that nothing in $G$ except the identity fixes all $x_i$. Such a set is called a base for the permutation group $G$. Any base gives rise to a chain of subgroups
\[ G = G_1 \geq G_2 \geq \cdots \geq G_k \geq G_{k+1} = \{1\}, \]
where $G_{i+1}$ is the stabilizer of $x_i$ in $G_i$ for $i = 1, \ldots, k$. And any generating set for $G$ that includes generators for each $G_i$ is called a strong generating set for $G$.

The inductive procedure described above for finding a base yields a generating set for each $G_i$, so we can simply take the union of these to get a strong generating set. That procedure, however, is very inefficient because the sizes of the generating sets (obtained via Schreier’s theorem) increase very rapidly. There are better algorithms
in which, roughly speaking, one tries to construct the chain \((G_i)\) all at once. At each step one has elements \(x_1, \ldots, x_l\) and a chain 

\[ G = H_1 > H_2 > \cdots > H_l > H_{l+1} = \{1\} \]

such that \(H_{i+1}\) stabilizes \(x_i\) but is not necessarily the full stabilizer of \(x_i\) in \(H_i\). One then enlarges one or more of the \(H_{i+1}\) by adding new generators. (If \(i = l\), one also has to enlarge the partial base by choosing a suitable \(x_{l+1}\).) Several such algorithms can be found in the references and will also be given in class.

6. Testing for membership

I mentioned at the beginning of this handout that most algorithms for permutation groups make use of a base and strong generating set (or BSGS). We have already seen how to use a BSGS to compute the order of the group. As another example, we indicate here how to test an arbitrary permutation of \(X\) to see if it is in our given group \(G\). Assume that we have computed the orbit \(Y_i := x_i G_i\) for each \(i\) and, as in Section 2, specific elements of \(G_i\) that map \(x_i\) to the elements of \(Y_i\) (i.e., a right transversal for \(G_{i+1}\) in \(G_i\)). This is usually done in the course of finding the BSGS.

Let \(g\) be a permutation of \(X\). Start by computing the image \(x^g_1\). If it is not in \(Y_1\), then we know \(g \notin G_1 = G\) and we are done. Otherwise, we can find \(t_1 \in G\) such that \(x^g_1 = x^g_1 t_1\). Then \(g' \equiv g t_1^{-1}\) fixes \(x_1\), and \(g \in G \iff g' \in G\), in which case \(g' \in G_2\). Now check whether \(x^g_2\) is in \(Y_2\). If not, then \(g' \notin G_2\), and hence \(g \notin G\). Otherwise, choose \(t_2 \in G_2\) such that \(x^g_2 = x^g_2 t_2\). Continuing in this way, either we discover that \(g \notin G\) or we find an expression \(g = t_k \cdots t_2 t_1\), where \(t_i\) is in our transversal for \(G_{i+1}\) in \(G_i\).

Remark. The procedure used in this algorithm is called sifting or stripping.

Exercise 7. Let \(G\) be the group of our running example. Show that the transposition \((1 \, 2)\) is not in \(G\).

7. Reduction of degree: Sub-\(G\)-sets and quotient \(G\)-sets

A standard technique in the study of permutation groups is the reduction of problems involving a group \(G\) to a problem involving a permutation group of lower degree. Here the degree is simply the size \(|X|\) of the set on which \(G\) acts.

One way of doing this is to restrict the action to a \(G\)-invariant subset \(Y\). This is only useful if the action on \(X\) is intransitive, so that \(Y\) can be taken to be proper and nonempty. We thus get a quotient \(G_Y\) of \(G\) that is a permutation group on \(Y\). If \(Z := X \setminus Y\) is the complement of \(Y\), we have two quotients \(G_Y\) and \(G_Z\), giving an embedding 

\[ G \hookrightarrow G_Y \times G_Z; \]

thus \(G\) is a subdirect product of two permutation groups of lower degree.

A second way of reducing the degree, usually used only if the action of \(G\) on \(X\) is transitive, is to look instead for quotients of \(X\) as a \(G\)-set. There are several equivalent ways of formulating what we are looking for:

- a \(G\)-set \(Y\) and a \(G\)-equivariant surjection \(X \twoheadrightarrow Y\);
- a \(G\)-invariant equivalence relation on \(X\);
- a partition of \(G\) into blocks that are permuted by the \(G\)-action.
In case we have chosen a basepoint in $X$ (so that we can identify $X$ with $H\backslash G$, where $H := G_x$), there is a fourth formulation:

- a subgroup $K$ such that $H \leq K \leq G$.

(Given $K$, set $Y := K\backslash G$. Conversely, given $Y$, let $K$ be the stabilizer of the image in $Y$ of the basepoint of $X$.) In particular, there exists a nontrivial quotient of $X$ if and only if $H$ is not a maximal subgroup of $G$. Given a quotient $G$-set $Y$, we get a quotient of $G$ that is a permutation group acting on $Y$.

**Example.** Let $G \leq S_6$ be the group generated by $r := (1 2 3 4 5 6)$ and $s := (26)(35)$. [This is the dihedral group of order 12, viewed as a group of permutations of the six vertices of a regular hexagon.] Suppose we want to construct the quotient obtained by requiring $1 \sim 5$. Acting by $s$ we see that $1 \sim 5$, so the equivalence class of $1$ contains $\{1, 3, 5\}$. Acting on this set by $r$ we deduce that $\{2, 4, 6\}$ is contained in a single equivalence class. Since these two sets are in fact permuted by $r$ and $s$, it follows that they are the equivalence classes, and we obtain a quotient $G$-set with two elements. From the point of view of stabilizers (using 1 as basepoint), forming the quotient corresponds to passing from the group $H = \langle s, r^2 \rangle$ of order 2 to the group $K = \langle s \rangle$ of order 6.

We close by describing the *minimal block* algorithm that does systematically what we just did in the example. We are given a transitive $G$-set $X$ and $m$ distinct elements $x_1, \ldots, x_m$. We want to construct the smallest $G$-invariant equivalence relation on $X$ that makes $x_1, \ldots, x_m$ equivalent. The method is reminiscent of the construction of quotient graphs to process coincidences in the Todd–Coxeter procedure. As usual, we denote by $S$ a given set of generators of $G$.

At each stage of the construction, every equivalence class will have a chosen representative; we denote by $\text{Rep}(x)$ the representative of the class of $x$. We will maintain a function $p : X \rightarrow X$ with the following properties:

- $p(x) \sim x$ for all $x \in X$;
- $p(x) = x$ if and only if $x = \text{Rep}(x)$;
- $\text{Rep}(x)$ can be computed by iterating $p$.

We will also maintain a list $Q$, which at all times consists of those $x \in X$ such that $\text{Rep}(x) \neq x$, listed without repetition. One should think of $Q$ as a queue of elements to be processed, except that we do not remove elements from $Q$ after processing them.

We initialize by setting $p(x_i) := x_1$ for $i = 2, \ldots, m$, and $p(x) := x$ for all other $x$. Thus $\{x_1, x_2, \ldots, x_m\}$ is an equivalence class, and all other equivalence classes are singletons. The main loop runs through the elements $x \in Q$. To process $x$, we run through the generators $s \in S$; given $s$, set $y := \text{Rep}(x)$ and merge the equivalence classes of $x^s$ and $y^s$ in order to make $x^s \sim y^s$.

It remains to explain the merging process. Let $u, v \in X$ be elements whose classes we want to merge. Compute $u' := \text{Rep}(u)$ and $v' := \text{Rep}(v)$. If $u' = v'$, there is nothing to do. Otherwise, pick one of them, say $u'$, to use as the representative of the merged class. Then set $p(v') := u'$ and append $v'$ to $Q$. Note that this preserves the property that $Q$ is a list without repetition of the elements $x$ such that $\text{Rep}(x) \neq x$. [Even though the merge might change $\text{Rep}(z)$ for one or more elements $z$ that occur earlier in $Q$, it can never cause $\text{Rep}(z)$ to become $z$.]
The description of \( Q \) makes it obvious that the procedure terminates. We must prove that the final equivalence relation is \( G \)-invariant, i.e.,

\[ x_1 \sim x_2 \implies x_1^s \sim x_2^s \]

for \( x_1, x_2 \in X \) and \( s \in S \). We may assume that at least one of \( x_1, x_2 \), say \( x_1 \), is in \( Q \); otherwise \( x_1 \sim x_2 \implies x_1 = x_2 \). At the time we processed \( x_1 \) and considered the generator \( s \), the element \( y := \text{Rep}(x_1) \) was not in \( Q \), and the processing of \( x_1 \) made \( x_1^s \sim y^s \). It therefore suffices to prove that \( y^s \sim x_2^s \). Thus we have “pushed the problem to the right”, i.e., we have replaced \( x_1 \) by an element that either occurs later in \( Q \) (if \( y \) got added to \( Q \) at some point) or is not in \( Q \) at all. A straightforward induction now completes the proof. [Roughly speaking, induct on the distance from \( x_1 \) and \( x_2 \) to the end of \( Q \).]

Let’s see how this procedure works in the example above. Initially we have \( p(3) = 1 \), and \( Q \) consists of 3 alone. To process 3 we consider first \( 3^r \) and then \( 3^s \). Since \( 3^r = 4 \) and \( \text{Rep}(3) = 1 \), we merge the classes of 4 and \( 1^r = 2 \). We’ll use 2 as the representative of the merged class, so we set \( p(4) := 2 \) and append 4 to \( Q \). Now consider \( 3^s = 5 \). We still have \( \text{Rep}(3) = 1 \), so we merge the classes of 5 and \( 1^s = 1 \). Using 1 as the representative, we set \( p(5) := 1 \) and append 5 to \( Q \). At this point the equivalence classes are \( \{1, 3, 5\} \), \( \{2, 4\} \), and \( \{6\} \), and \( Q \) is the list 3, 4, 5.

We have finished processing 3, so we move on to 4. We have \( \text{Rep}(4) = 2 \), so we must merge the classes of \( 4^r = 5 \) and \( 2^r = 3 \). But these classes are already the same. Next we merge the classes of \( 4^s = 4 \) and \( 2^s = 6 \), using 2 = \( \text{Rep}(4) \) as the representative. The equivalence classes are now \( \{1, 3, 5\} \) and \( \{2, 4, 6\} \), and \( Q \) is the list 3, 4, 5, 6. We have finished processing 3 and 4. The reader can easily verify that there is no further change when we process 5 and 6.

**Appendix A. A computational proof of Schreier’s theorem**

The short proof of Schreier’s theorem sketched in Section 3 (see the hint to the Exercise) is unsatisfactory from the computational point of view. One would like to be able to take a word in the generators of \( G \) that represents an element of \( H \) and rewrite it in terms of the Schreier generators. We will give such a proof in this appendix, which is loosely based on Holt’s treatment [1, Section 2.5.1].

Let \( G \) be a group with a generating set \( S \). Let \( H \leq G \) be a subgroup, and let \( T \) be a right transversal for \( H \) in \( G \). We assume that the elements of \( T \) are explicitly given as reduced words in \( S \cup S^{-1} \). We will also assume that \( T \) (as a set of words) is closed under prefixes (and hence, in particular, \( 1 \in T \)). The transversals described in Section 2 have these properties. We will use bars to denote coset representatives. Thus every \( g \in G \) can be expressed as \( g = ht \) with \( h \in H \) and \( t = \overline{g} \in T \).

Note that \( T \) is rarely closed under right translation by \( G \). To measure the failure of \( T \) to be closed, we define elements \( h(t, g) \in H \) (for \( t \in T \) and \( g \in G \)) by requiring

\[ tg = h(t, g)u, \]

with \( u \in T \); necessarily \( u = \overline{tg} \). We will use Equation (1) as a “rewriting rule” that will help us write arbitrary words in the canonical form \( ht \). For lack of a better name, I will call \( h(\cdot, \cdot) \) the **rewriting function** associated to \( G, H, T \). The elements \( h(t, s) \) with \( s \in S \) are precisely the **Schreier generators** of \( H \) introduced in Section 3. Schreier’s theorem, which we will prove below, asserts that they do in fact generate \( H \).
The rewriting function has the following three properties:

\[(2) \quad h(1, t) = 1\]

for all \( t \in T \).

\[(3) \quad h(t, g^{-1}) = h(u, g)^{-1}\]

for \( t \in T \) and \( g \in G \), where \( u := \overline{tg^{-1}} \).

\[(4) \quad h(t, g_1g_2) = h(t, g_1)h(u, g_2),\]

for \( t \in T \) and \( g_1, g_2 \in G \), where \( u := \overline{tg_1} \).

Equation (2) is immediate from the definition. To prove (3), start with

\[ug = h(u, g)t\]

and take inverses to get

\[g^{-1}u^{-1} = t^{-1}h(u, g)^{-1}.\]

Now right-multiply by \( u \) and left-multiply by \( t \) to get

\[tg^{-1} = h(u, g)^{-1}u;\]

Equation (3) follows at once. To prove (4), observe that two rewritings give

\[tg_1g_2 = h(t, g_1)ug_2\]

where \( v := \overline{tg_1g_2} \); the result now follows from the definition of \( h(\cdot, \cdot) \).

Equation (4) generalizes to

\[(5) \quad h(t, g_1g_2 \cdots g_l) = h(t_1, g_1)h(t_2, g_2) \cdots h(t_l, g_l),\]

where \( t_i := \overline{tg_1g_2 \cdots g_{i-1}} \). This can be proved in the same way as (4), or it can be deduced from (4) by induction.

Let’s specialize now to \( t = 1 \) and \( g_i = s_i \in S \cup S^{-1} \). Setting \( g := \overline{s_1s_2 \cdots s_l} \) and noting that \( g = h(1, g)g \) by definition, we obtain an expression of \( g \) as \( h\overline{g} \) with \( h = h(1, g) \) given as a product of the \( h(t, s) \) with \( s \in S \cup S^{-1} \). Using Equation (3), we can reduce this further to an expression involving only the Schreier generators \( h(t, s) \) (\( s \in S \)). Applying this with \( g \in H \) proves Schreier’s theorem, in a very explicit form.

**Remark.** We haven’t yet used the assumption that \( T \) (viewed as a set of reduced words) is prefix closed. One consequence of this assumption is that some of the Schreier generators are trivial. Namely, for each \( t \neq 1 \) in \( T \), write \( t = \overline{s_1s_2 \cdots s_l} \) with \( s_i \in S \cup S^{-1} \); then \( t' := \overline{s_1s_2 \cdots s_{l-1}} \) is in \( T \), and \( h(t', s_l) = 1 \). If \( s_l \in S \), this is a trivial Schreier generator. If \( s_l \in S^{-1} \), then \( h(t'', s_l^{-1}) \) is a trivial Schreier generator, where \( t'' := \overline{t's_l} \). It is straightforward to check that no other Schreier generators are trivial in general. [Consider the case where \( G \) is free.]

We now prove the Nielsen–Schreier theorem mentioned in Section 3.

**Theorem.** Let \( F = F(S) \) be the free group on a set \( S \), let \( E \leq F \) be a subgroup, and let \( T \) be a prefix-closed right transversal for \( E \) in \( F \). Then \( E \) is freely generated by the nontrivial Schreier generators. If \( |S| = n < \infty \) and \( |F : E| = k < \infty \), then there are precisely \( nk - (k - 1) = n(k - 1) + 1 \) such generators.
Proof. The last assertion is immediate from the remark above. To prove the first assertion, let $e(\cdot, \cdot)$ be the rewriting function for $F,E,T$; thus the $e(t,s)$ with $s \in S$ are the Schreier generators of $E$. Let $\bar{E}$ be the free group on symbols $\bar{e}(t,s)$ with $t \in T$, $s \in S$, and $ts \notin T$. (These correspond precisely to the nontrivial Schreier generators.) There is an obvious surjection $\bar{E} \to E$ such that $\bar{e}(t,s) \mapsto e(t,s)$. To prove the theorem, we will define a homomorphism in the opposite direction such that $e(t,s) \mapsto \bar{e}(t,s)$. The crucial step is to lift the rewriting function $e(\cdot, \cdot)$ to a function

$$\bar{e}: T \times F \to \bar{E}.$$ 

We’ve already defined $\bar{e}(t,s)$ for $t \in T$ and $s \in S$ such that $ts \notin T$. Now define

$$\bar{e}(t,s) := \bar{1}$$

for $t \in T$ and $s \in S$ such that $ts \in T$. Next, set

$$\bar{e}(t,s^{-1}) := e(u,s)^{-1}$$

for $t \in T$ and $s \in S$, where $u := ts^{-1}$. Finally, motivated by Equation (5), we would like to define

$$\bar{e}(t,s_1s_2\cdots s_l) = \bar{e}(t_1,s_1)\bar{e}(t_2,s_2)\cdots \bar{e}(t_l,s_l)$$

for $t \in T$ and $s_i \in S \cup S^{-1}$, where $t_i := \bar{e}s_1s_2\cdots s_{i-1}$. To justify this, let $M := (S \cup S^{-1})^*$ be the free monoid on $S \cup S^{-1}$. We can certainly define a function $\rho: T \times M \to \bar{E}$ by

$$\rho(t,s_1s_2\cdots s_l) := \bar{e}(t_1,s_1)\bar{e}(t_2,s_2)\cdots \bar{e}(t_l,s_l).$$

Recall now that $F$ is the quotient of $M$ by the equivalence relation in which two words are equivalent if one can be obtained from the other by finitely many insertions or deletions of subwords $ss^{-1}$ with $s \in S \cup S^{-1}$. So it suffices to check that $\rho(w_1sw^{-1}w_2) = \rho(w_1w_2)$ for all $w_1,w_2 \in M$ and $s \in S \cup S^{-1}$; this follows easily from the definition of $\rho$ together with (7).

Note, for future reference, that the analogue of (2) holds:

$$\bar{e}(1,t) = 1$$

for $t \in T$. This follows from (8) and (9) and the assumption that $T$ is prefix closed.

It follows from (8) that

$$\bar{e}(t,w_1w_2) = \bar{e}(t,w_1)\bar{e}(u,w_2)$$

for $t \in T$ and $w_1,w_2 \in F$, where $u := \bar{w}w_1$. It is also easy to check, either directly from (8) or as a consequence of (10), that

$$\bar{e}(t,w^{-1}) = \bar{e}(u,w)^{-1}$$

for $t \in T$ and $w \in F$, where $u := \bar{w}w^{-1}$.

Now define $\psi: E \to \bar{E}$ by

$$\psi(w) := \bar{e}(1,w)$$

for $w \in E$. Equation (10) implies that $\psi$ is a homomorphism. It remains to show that $\psi(e(t,s)) = \bar{e}(t,s)$ for $t \in T$ and $s \in S$. Recall that, by definition,
\[ e(t, s) = tsu^{-1}, \text{ where } u := t. \] Using (10), (11), and (9), we now compute
\[ \psi(e(t, s)) = \tilde{e}(1, e(t, s)) = \tilde{e}(1, tsu^{-1}) = \tilde{e}(1, t)\tilde{e}(t, s)\tilde{e}(1, u)^{-1} = \tilde{e}(t, s), \]
as required. \(\Box\)

We close this section by proving the Reidemeister–Schreier theorem, which gives a presentation for a subgroup \( H \leq G \) if we are given a presentation for \( G \). Let \( G = \langle S ; R \rangle \). Thus \( G = F/N \), where \( F := F(S) \), \( R \) is a subset of \( F \), and \( N \) is the normal closure of \( R \) in \( F \). Let \( E \) be the subgroup of \( F \) such that \( E \geq N \) and \( H = E/N \). Let \( T \) be a prefix-closed transversal for \( E \) in \( F \), and let \( e(-, -) \) be the corresponding rewriting function. We know that \( E \) is freely generated by the nontrivial Schreier generators. To get a presentation for \( H \), then, we need a subset of \( N \) whose normal closure in \( E \) is \( N \).

**Theorem.** \( N \) is the normal closure of \( \{ e(t, w) \mid t \in T, w \in R \} \). Hence these elements form a set of defining relators for \( H \) with respect to the nontrivial Schreier generators.

**Proof.** We know that \( N \) is generated by the elements \( fwf^{-1} \) \((f \in F, w \in R)\). Writing \( f = et \) with \( e \in E \) and \( t \in T \), we conclude that \( N \) is the normal closure in \( E \) of the elements \( twt^{-1} \) \((t \in T, w \in R)\). Since \( tw = (twt^{-1})t \), the definition of the rewriting function implies that \( twt^{-1} = e(t, w) \), whence the theorem. \(\Box\)

**References**

