Algebra and Computation Course Instructor: V. Arvind

Lecture 4: Membership Testing in Permutation Groups

Lecturer: V. Arvind *Scribe: Shreevatsa R and Ramprasad*

1 Overview

In the previous lecture, we began studying the problem of testing membership in permutation groups. In this lecture, we describe its solution and explore related problems.

2 The problem

Given a subgroup G (given by a small generating set, say $G = \langle A \rangle$) of the permutation group S_n, and a permutation $g \in S_n$, the **membership testing** problem is to decide whether $g \in G$. If the answer is yes, we might also want a representation of g in terms of the generators.

If the output has to be a product $g = a_{i_1} a_{i_2} \dots a_{i_N}$, how large might N have to be, in the worst case? It is easy to see that we can ensure N is at most by $|G| - 1$: consider the prefix products $s_1 = a_{i_1}, s_2 = a_{i_1} a_{i_2}, \dots, s_N = a_{i_1} a_{i_2} \dots a_{i_N}$. If $N > |G| - 1$, either some s_j is the identity, in which case we can write $g = a_{i_{j+1}} \dots a_{i_N}$, or $s_j = s_k$ for some j and k, in which case we can write $g = a_{i_1} \dots a_{i_j} a_{i_{k+1}} \dots a_{i_N}$, so in either case N can be made smaller.

However, N might have to be very large. As an example, write n as $n = p_1 + p_2 + \dots + p_m$ where the p_is are distinct primes, and look at the cyclic group generated by

$$a = (1\ 2\ \dots\ p_1)(p_1 + 1\ \dots\ p_1 + p_2)(\dots) \quad \dots \quad (\dots) = C_1 C_2 \dots C_m.$$

The C_is are disjoint cycles with lengths p_i, so the order of a is $M = p_1 p_2 \dots p_m$. The group is $G = \langle \{a\} \rangle = 1, a, a^2, \dots, a^{M-1}$, in which the element a^{M-1} can be written only as the product of $M - 1$ as. So here, $N = p_1 p_2 \dots p_m - 1 \geq 2^m - 1$, and we know by the prime number theorem that $m = \Omega(\frac{p_m}{\ln p_m})$. Further, as $n = p_1 + p_2 + \dots + p_m \leq 1 + 2 + 3 + \dots + p_m \leq {p_m}^2$, we have that $p_m \geq \sqrt{n}$, so $N \geq 2^{\frac{c\sqrt{n}}{\log n}} - 1$ for some c. This is not polynomial in n.

Thus, we cannot hope to solve the problem in polynomial time if the required output is an explicit product; the "representation" has to be a circuit on A. In the example above, a^{M-1} can easily be expressed (or computed) in terms of the a through repeated squaring. Our algorithm will give a similar good representation.

3 Idea

Membership testing reduces to the problem of **order computation**, as $g \in G \iff |G| = |G \cup g|$. The idea for solving the latter, as we saw in the previous lecture, is to find a tower of subgroups

$$G = G^{(0)} \geq G^{(1)} \geq \cdots \geq G^{(n-1)} = \{1\}$$

such that the index $\left[G^{(i-1)} : G^{(i)}\right]$ is easy to calculate, for each i. Then, as $|G| = \prod_{i=1}^{n-1} \left[G^{(i-1)} : G^{(i)}\right]$, we can easily compute it.

Consider the pointwise stabilisers, and look at the tower of subgroups

$$G^{(i)} = \{g \in G \ : \ j^g = j \text{ for all } j, \ 1 \leq j \leq i\}$$

Here, $\left[G^{(i-1)} : G^{(i)}\right] \leq n - i$ for every i(why? since fixing $i-1$ leaves $n-i$ choices for the ith index), and our algorithm will compute it by computing a system of coset representatives for this. The algorithm will also give us a *new* generating set for G; thus it can be used to decide many membership queries without recomputing it.

4 Algorithm

4.1 Finding the cosets at each level of the tower

How do we find the coset representatives of $G^{(1)}$ in G? Let X_1 be the orbit of 1 under the action of G, and for any $k \in X_1$, let g_{1k} be the element that maps 1 to it (i.e., $1^{g_{1k}} = k$). Then

$$G = \bigcup_{k \in X_1} G^{(1)} g_{1k}$$

We shall see shortly (in subsection 4.3) how to find a small generating set for $G^{(1)}$. Assuming that we can do it, we can similarly find X_2, the orbit of 2 under the action of $G^{(1)}$, find the g_{2k}s such that $k = 2^{g_{2k}}$, and similarly find representatives for the cosets of $G^{(2)}$ in $G^{(1)}$.

In general, for each i from 1 to n, we find the orbit X_i of i under the action of $G^{(i-1)}$, and also, for each element $k \in X_i$, the element g_{ik} such that $i^{g_{ik}} = k$. Then $G^{(i-1)} = \bigcup_{k \in X} G^{(i)} g_{ik}$

Taking the union of the sets of coset representatives we get for each $\left[G^{(i-1)} : G^{(i)}\right]$ gives us a generating set for G. What we have just found has a name:

Definition 1. *Let Ω be an ordered set $\{\omega_1, \omega_2, \ldots, \omega_n\}$ and let $G \leq \mathrm{Sym}(\Omega)$. A* ***strong generating set*** *with respect to this ordering is the union of the sets of right-coset representatives (or the right-transversals) for $G^{(i)}$ in $G^{(i-1)}$, for $0 \leq i \leq n-1$.*

4.2 Testing membership of a given g

Given a $g \in G$, let $k_1 = 1^g$. If $k \notin X_1$, we can immediately conclude that $g \notin G$. Else, let $g^{(1)} = g g_{1k}^{-1}$. By construction, $g^{(1)} \in G^{(1)}$. Next, letting $k_2 = 2^{g^{(1)}}$, if $k_2 \notin X_2$, we can abort and report that $g \notin G$; else we let $g^{(2)} = g g_{1k_1}^{-1} g_{2k_2}^{-1}$ and continue similarly. In other words, for each i from 1 to n, we let $k_i = i^{g^{(i-1)}}$, abort if $k_i \notin X_i$, and set $g^{(i)} = g^{(i-1)} g_{ik_i}^{-1}$ otherwise, in which case it is guaranteed to be in $G^{(i)}$. If at any time $g^{(r)} = g g_{1k_1}^{-1} g_{2k_2}^{-1} \cdots g_{rk_r}^{-1} = 1$ then we have g as a product of elements of G. If $g \in G$, it is guaranteed that this will eventually happen, for if we have not aborted by the time i takes the value of n, then it is guaranteed that $g^{(n-1)} \in G^{(n-1)} = 1$.

4.3 Finding generating sets for the subgroups

Our algorithm above for finding the strong generating set depends on being able to find a generating set for $G^{(i)}$ when we know one for $G^{(i-1)}$. This is made possible by Schreier's lemma, as we saw in the previous lecture.

Theorem 1 (Schreier's lemma)**.** *If $G = \langle A \rangle$ and R is a set of distinct right coset representatives for the subgroup H in G, then*

$$B = \left\{ r_1 a r_2^{-1} \ : \ a \in A, r_1, r_2 \in R \right\} \cap H$$

generates H.

For every r_1 and a, there is a unique r_2 such that $r_1 a r_2^{-1} = h$, for any h in H, so we know that $|B| \leq |R|\,|A|$. However, this alone is not sufficient for us to complete our algorithm, as it does not ensure polynomial time — the size of the generating set might increase by $\Theta(n)$ each time. To avoid this, we need to make sure we can keep the generating set small. We now show that it can be done.

4.4 The REDUCE Algorithm

Theorem 2. *Given a group $G \leq S_n$, we can find a generating set for it of size at most n^2.*

The idea is to remove collisions as you see them. Suppose π and ψ are two elements of your generating set such that they map 1 to the same element, (i.e) $1\pi = 1\psi$. What we do then is replace $\{\pi, \psi\}$ by $\{\pi, \psi\pi^{-1}\}$. This then ensures that one of the elements fix 1, and is taken care in the following levels. A better way to look at it would be to think of a table where row i represents $G^{(i)}$.

Start with the bottom row, representing $G^{(0)} = G$. For each element $\pi \in B$, if $1\pi = 1$, move it to the row $G^{(1)}$, else place it in column 1^π. Whenever you have collisions, do the pruning process and send the element that fixes 1 and send it to the next row. Once row 1 is done (no more collisions), move to the next and repeat this process.

The complete algorithms for REDUCE and MEMBERSHIP can be found at the end of this file.

5 Other Problems

Using the technique of using group towers, there is a whole pool of problems we could inspect. We shall see a few of them now, which will be very useful for the lectures to follow.

SUBGROUP: Given $H = \langle B \rangle$ and $G = \langle A \rangle$, subgroups of Sym_n. Check if $H \leq G$.

The solution is extremely simple, for every element $b \in B$, check if $b \in G$ using the MEMBERSHIP algorithm.

NORMAL: Given $H = \langle B \rangle$ and $G = \langle A \rangle$, subgroups of Sym_n. Check if H is a normal subgroup of G, denoted by $H \lhd G$.

This is also easy, for each $a \in A$ and each $b \in B$, check if $aba^{-1} \in H$ using MEMBERSHIP.

Definition 2. *For any subgroup H of G, normal closure of H is defined as the smallest normal subgroup of G that contains H. It is denoted by $\langle H^G \rangle$*

$$H \leq \langle H^G \rangle \lhd G$$

The normalizer of H, denoted by $N_G(H)$, is the largest subgroup of G in which H is normal.

$$H \lhd N_G(H) \leq G$$

NORMAL CLOSURE: Given $H = \langle B \rangle$ and $G = \langle A \rangle$, subgroups of Sym_n, find $\langle H^G \rangle$

This is easy too, look at $\left\{aba^{-1} \ : \ a \in A, b \in B\right\}$. If something of this set is not in H, throw it into B and repeat. Everytime, the size of the group doubles and hence you will definitely hit the normal closure quickly.

5.1 The Group-Intersection Problem

GROUP-INTER: Given $G = \langle A \rangle$ and $H = \langle B \rangle$, subgroups of Sym_n. Find $G \cap H$.

There is no known polynomial time algorithm for this problem in general, and we don't expect one to exists even because of the following theorem.

Theorem 3. SET-STAB *is polynomial time equivalent to* GROUP-INTER

Proof. SET-STAB $\leq_P$ GROUP-INTER:

Suppose Δ is the set we want to stabilize, all we need to do is to compute $G \cap \left\{\mathrm{Sym}_\Delta \times \mathrm{Sym}_{\Omega \setminus \Delta}\right\}$. One could just choose transpositions as a generating set for the product.

The intersection of the two sets precisely yields $stab_\Delta(G)$.

GROUP-INTER $\leq_P$ SET-STAB:

Look at the product $G \times H \leq \mathrm{Sym}_\Omega \times \mathrm{Sym}_\Omega \leq Sym_{\Omega \times \Omega}$ and let the action be just the coordinate wise action, $(g,h)(i,j) = (i^g, j^h)$. All we need to do now is stabilize the diagonal, $\Omega \times \Omega = \{(i,i) \ : \ i \in \Omega\}$ and this would yield $G \cap H$. $\square$

And since we say that graph isomorphism reduced to SET-STAB, it is unlikely that we have a polynomial time algorithm for the intersection problem.

However, for a special case when G normalizes H (i.e $G \leq N_{\mathrm{Sym}_\Omega}(H)$), we can solve the intersection problem in polynomials time.

Claim 4. *If G normalizes H, then we can compute $G \cap H$ in polynomial time.*

Proof. Again, we are looking for a tower of subgroups with "nice" properties. And since G normalizes H, the central idea is that $GH = \{gh \ : \ g \in G, h \in H\}$ is a subgroup of Sym_n. And hence, for all i, $G^{(i)}H$ is also a subgroup. Further, G normalizes $H \implies H \lhd GH$.

The tower we are looking for is

$$G \cap H = G \cap G^{(n-1)}H \leq G \cap G^{(n-2)}H \leq \cdots \leq G \cap GH = G$$

And since the generating set for $G^{(i)}H$ is just the union of the generating set for $G^{(i)}$ and H, we can check for membership in $G \cap G^{(i)}H$. Thus, using Schreier's lemma and the REDUCE algorithm, we can descend the tower computing generating sets.

An additional property we need is that the index between consecutive elements of the tower is small. We leave it as an exercise to show that $\left[G \cap G^{(i-1)}H : G \cap G^{(i)}H\right] \leq n - i$. $\square$

Algorithm 1 REDUCE

```
 1: $B_0 = B$
 2: $A[\,][\,]$, an empty $n \times n$ array
 3: for $i = 0$ to $n-1$ do
 4:    for all $\psi \in B_i$ do
 5:       $j = i^{\psi}$
 6:       if $A[i][j]$ is empty then
 7:          if $j = i$ then
 8:             $B_{i+1} = B_{i+1} \cup \{\psi\}$
 9:          else
10:             $A[i][j] = \psi$
11:          end if
12:       else
13:          $\pi = A[i][j]$
14:          $B_{i+1} = B_{i+1} \cup \left\{\pi \cdot \psi^{-1}\right\}$
15:       end if
16:    end for
17: end for
18: discard all trivial elements from $\bigcup B_i$
19: return $\bigcup B_i$
```

Algorithm 2 MEMBERSHIP

```
Input: $g \in \mathrm{Sym}_n$ and a generating set $A$ for $G^{(i)} \leq \mathrm{Sym}_n$ and the index $i$
 1: if $g = id$ then
 2:     return true
 3: end if
 4: $X_i = (i+1)^{G^{(i)}}$   {use the ORBIT algorithm}
 5: compute the set $R$ of distinct coset representatives of $G^{(i+1)}$ in $G^{(i)}$
 6: $k = (i+1)^g$   {image of $i+1$ under the action of $g$}
 7: if $k \notin X_i$ then
 8:     return false
 9: else
10:     compute generating set $B$ for $G^{(i+1)}$ using Schreier's lemma
11:     REDUCE $B$
12:     pick $g_{ik}$ from $R$, the coset representative of $G^{(i)}$
13:     $g' = g \cdot g_{ik}^{-1}$
14:     return MEMBERSHIP($g'$,$B$, $i+1$)
15: end if
```