Algorithms for permutation groups III

Michal Hrbek

In this talk, we present several applications of the Schreier-Sims algorithm. Any version of the Schreier-Sims algorithm can be used, including the nearly linear Monte Carlo one. We will show how several group-theoretic tasks can be done effectively using modifications of the Schreier-Sims algorithm. An economic way of computing strong gerenating set of a group will be shown in cases when we already know a base of the group, or when we have a strong generating set of a subgroup. Finally, a memory-wise cheaper representation of elements of a group by storing just the images of a base in their action will be discussed.

1 Some basic algorithms

Let Ω be a finite set and $G \leq \operatorname{Sym}\Omega$ a permutation group given by the set of generators S. Modifying the Schreier-Sims algorithm in various ways, we get effective algorithms for the following tasks:

- Given $g \in \operatorname{Sym}\Omega$, determine wheter g lies in G or not.
- Compute the stabilizer $G_{(\Delta)}$ for some $\Delta \subseteq \Omega$.
- Given a finite set Δ and a map $\varphi\colon S \to \operatorname{Sym}\Delta$, determine wheter φ defines a group homomorphism.
- Compute the (generators of the) kernel of a homomorphism φ.
- For any $g \in G$ and $h \in \varphi(G)$, compute $\varphi(g)$ and some representative of the coset $\varphi^{-1}(h)$.

The algorithms have the same complexity as the chosen version of the Schreier-Sims algorithm.

2 Closures

Definition 1. Let $G \leq \operatorname{Sym}\Omega$ be a permutation group and suppose we already have its base $B = (b_1, \cdots, b_m)$ and a strong generating set S relative to B computed. Let $T \subseteq \operatorname{Sym}\Omega$, then the group $H = \langle S \cup T\rangle$ is called a *closure* of G by T.

We will show a way of computing an SGS of a closure efficiently, without a need to use the Schreier-Sims algorithm from scratch.

3 Base images

Suppose that we have a base B for a group $G \leq \operatorname{Sym}\Omega$. The elements of G are determined by the images of the set B under their action. Indeed, $B^g = B^h$ implies that gh^{-1} fixes every point of B and thus $gh^{-1} = 1_G$, so $g = h$. If B is smaller then Ω, this method can save some memory. We need an algorithm for recovering the elements of G from the base images:

Lemma 2. *Let S be an SGS for $G \leq \operatorname{Sym}\Omega$ relative to some base B and let t be the sum of depths of Schreier trees used in Schreier-Sims algorithm. There is an algorithm which for every injection $f\colon B \to \Omega$ finds $g \in G$ such that $B^g = f(B)$ or determines that no such g exists in $O(t|\Omega|)$ time.*

The g obtained by this algorithm will be expressed as a unique product of elements of the transversals R_i used in Schreier-Sims algorithm. If we settle just for expressing g as a word in elements of S, the algorithm can run in $O(t|B|)$ time. This can be shown by using a modified version of the *sifting* procedure used in the Schreier-Sims algorithm, which works with words in S rather then elements of a transversal. We will show that similar technique can be used to speed up the process of computing an SGS, provided that we already have a base of the group:

Theorem 3. *Suppose that B is a base for some $G = \langle S \rangle \leq \operatorname{Sym}\Omega$ with $|\Omega| = n$. Then an SGS for G can be computed in $O(n|B|^2 \, |S| \log^3 |G|)$ time.*

One example of such situation, is when we compute an SGS for some group G, and then we want an SGS for some of its subgroups. The base of G can be used as a base for any of subgroups of G.