ALGORITHMS FOR PERMUTATION GROUPS – PART II

Petr Nižnanský

In this part we show the Schreier-Sims algorithm for computing a strong generating set (SGS) for group G. An algorithm is deterministic and works in polynomial time. We denote $\mathrm{Sym}(\Omega)$ as a symmetric group on set Ω (we interested only in finite groups so $|\Omega| < \infty$). The key component is Schreir-Sims lemma (in previous abstract) and following lemma.

Lemma 1. *Let* $\{ \beta_1, \dots, \beta_k \} \subseteq \mathrm{Sym}(\Omega)$ *and* $G \leq \mathrm{Sym}(\Omega)$. *For* $1 \leq j \leq k+1$, *let* $S_j \subseteq G_{(\beta_1,\dots,\beta_{j-1})}$ *such that* $\langle S_j \rangle \geq \langle S_{j+1} \rangle$ *hold for* $j \leq k$. *If* $G = \langle S_1 \rangle$, $S_{k+1} = \emptyset$, *and*

$$\langle S_j \rangle_{\beta_j} = \langle S_{j+1} \rangle$$

holds for all $1 \leq j \leq k$ *then* $B = (\beta_1, \dots, \beta_k)$ *is a base for* G *and* $S = \bigcup_{j=1}^{k} S_j$ *is an SGS for* G *relative to* B.

We say that data structure is *up to date below level* j if $\langle S_i \rangle_{\beta_i} = \langle S_{i+1} \rangle$ holds for all i satisfying $j < i \leq m$.

An algorithm itself is recursive and core of algorithm is following: Suppose we have $B = \{ \beta_1, \dots, \beta_m \}$ a nonredundant base and data structure up to date below level j (e.g. we have SGS for $\langle S_{j+1} \rangle$). Every step of algorithm approximate SGS of G, which is denoted by S. We compute a transversal R_j for $\langle S_j \rangle \bmod \langle S_j \rangle_{\beta_j}$. Generators are known (Schreier-Sims lemma). We test all Schreier generators whether are they in $\langle S_{j+1} \rangle$. If this is satisfied then the data structure is up to date below level $j-1$. If Schreier generator g is not in $\langle S_{j+1} \rangle$, then we find siftee g' of g. We add g' to S and our data structure is up to date below $j+1$. If $j = m$ then add to B a new point not fixed by g'. The algorithm terminates when the data structure becomes up to date below level 0. Lemma 1 implies correctness.

The algorithm starts with choosing $\beta_1 \in \Omega$ that is moved by at least one generator in T (set which generates G) and setting $S_1 := T$. The data structure is up to date below level 1.

The Schreier-Sims algorithm has several improvements some of them are probabilistic.

Probably the most common used variant (probabilistic) of Schreier-Sims algorithm has following properties:

Theorem 2. *Suppose that* $G = \langle T \rangle \leq \mathrm{Sym}(\Omega)$ *and* $|\Omega| = n$. *There is an algorithm that, with error probability less than* $1/n^d$ *for a constant* d *prescribed by the user, constructs and SGS for* G *in* $O(n \log n \, \log^4(|G|) + |T| \, n \log(|G|))$ *time using* $O(n \log(|G|) + |T| \, n)$ *memory.*