Algorithms for permutation groups
Part III

Michal Hrbek

March 24, 2012

Algorithms

Let Ω be a finite set and $G = \langle S \rangle \leq \operatorname{Sym} \Omega$ a permutation group given by generators S.

Task

For a permutation $g \in \operatorname{Sym} \Omega$, determine wheter g is an element of G.

Compute a base and a strong generating set for G. Then the membership is decided by sifting.

Task

For a subset $\Delta \subseteq \Omega$, find generators of its (pointwise) stabilizer $G_{(\Delta)}$.

- Declare every element of Ω to be a base point
- Use base points from Δ first
- The $(k+1)$-th stabilizer group is $G_{(\Delta)}$

Task

Given a finite set Δ and a map $\varphi : S \to \operatorname{Sym}\Delta$, decide wheter φ defines a homomorphism $G \to \operatorname{Sym}\Delta$.

- Define $H = \langle (g, \varphi(g)) \mid g \in S \rangle$ (a subgroup of $\operatorname{Sym}\Omega \times \operatorname{Sym}\Delta$)
- Observe that φ defines a homomorphism if and only if $H_{(\Omega)}$ is trivial

Task

Let $\varphi : G \to \operatorname{Sym}\Delta$ be an action of G on Δ. Find its kernel.

- Define $\bar{G} = \{(g, \varphi(g)) \mid g \in G\}$, a subgroup of $\operatorname{Sym}\Omega \times \operatorname{Sym}\Delta$ isomorphic to G
- Observe that $g \in \operatorname{Ker}\varphi$ if and only if $(g, \varphi(g)) \in \bar{G}_{(\Delta)}$

Task

For any $g \in G$ and $h \in \varphi(G)$, compute $\varphi(g)$ and some representative of coset $\varphi^{-1}(h)$ effectively.

- Compute two strong generating sets S_1, S_2 for $\bar{G} = \{(g, \varphi(g) \mid g \in G\}$, where S_1 is relative to a base $B_1 = (\beta_1, \dots, \beta_m)$, such that $\beta_1, \dots, \beta_k \in \Omega$ and $\beta_{k+1}, \dots, \beta_m \in \Delta$ for some $1 \leq k \leq m$ and S_2 is relative to B_2 with roles of Ω and Δ inversed
- Observe that $\varphi(g)$ can be computed by sifting $(g, 1)$ in Schreier data structure corresponding to S_1 and restricting the inverse of the siftee to Δ
- Observe that representative of $\varphi^{-1}(h)$ can be computed by sifting $(1, h)$ in Schreier data structure corresponding to S_2 and restricting the inverse of the siftee to Ω

Closures

Definition

Let Ω be a finite set and $G = \langle S \rangle \leq \operatorname{Sym} \Omega$ a permutation group. Suppose that we have a strong generating set S_1 of G relative to some base B. If $T \subseteq \operatorname{Sym} \Omega$ then we call a group $H = \langle S_1 \cup T \rangle$ the closure of G by T.

Task

Compute a strong generating set of the closure of G by T without a need to construct it from scratch.

- Add T to the generating set of G and recompute the first fundamental orbit β_1^H and the coresponding transversal H modulo H_{β_1}
- Declare that our data structure is up to date below level 1 in order to initialize the Schreier-Sims algorithm

Definition

Let $H = \langle T \rangle \leq \mathrm{Sym}\,\Omega$, $G = \langle S \rangle \leq \mathrm{Sym}\,\Omega$ and suppose that G has an action on H. The algebraic closure $\langle H^G \rangle$ is called a G-closure of H.

Task

Compute a G-closure of H effectively. (We suppose that we can compute an algebraic closure of a set of generators)

- Suppose that T is an SGS of H
- Let $H_1 = H$ and for all $h \in T_1 = T, g \in S$ collect h^g such that $h^g \notin H_1$ into a list L
- Compute an algebraic closure of $T_1 \cup L$, recompute SGS T_2 of H_2
- Iterate until L is empty

Base images

Let $G \leq \operatorname{Sym}\Omega$ be a permutation group with base B. Instead of storing an element $g \in G$ as a permutation, we can remember just the images of base points in action of g. Since $B^g = B^h$ (pointwise) implies that gh^{-1} fixes B pointwise and hence $g = h$, images of base points determine g uniquely.

Task

Recover $g \in G$ effectively from its base images.

Algorithm

Let $G \leq \operatorname{Sym}\Omega$ be a permutation group with an SGS S relative to B and let t be the sum of depths of Schreier trees coding the coset represetative sets along the point stabilizer chain of G. If $f : B \to \Omega$ is an injection, it is possible to find an element $g \in G$ such that $B^g = f(B)$ or decide that no such element exists in $O(t|\Omega|)$ time.

Algorithm

Let $G \leq \operatorname{Sym}\Omega$ be a permutation group with an SGS S relative to B and let t be the sum of depths of Schreier trees coding the coset represetative sets along the point stabilizer chain of G. If $f: B \to \Omega$ is an injection, it is possible to find an element $g \in G$ such that $B^g = f(B)$ or decide that no such element exists in $O(t|\Omega|)$ time.

- Suppose that $B = (\beta_1, \ldots, \beta_m)$ and $G = G^1 \geq \cdots \geq G^{m+1}$ is the point stabilizer chain.
- If $f(\beta_1)$ lies in orbit $\beta_1^{G^1}$, take the product of edge labels along the path from $f(\beta_1)$ to β_1 in the first Schreier tree. We get $r_1 \in G$ such that $f(\beta_1)^{r_1} = \beta_1$
- Define $f_2 : B \to \Omega$ by $f_2(\beta_i) = f(\beta_i)^{r_i}$. If $f_2(\beta_2) \in \beta_2^{G^2}$, we take a product of edge labels from $f_2(\beta_2)$ to β_2 in the second Schreier tree. We get $r_3 \in G$ such that $f_3 : B \to \Omega$ defined by $f_3(\beta_i) = f(\beta_i)^{r_1 r_2}$ fixes β_1, β_2

- Iterating this process we get $g = r_1 r_2 \dots r_m \in G$ such that $f(B) = B^{(g^{-1})}$
- If $f_i(\beta_i) \notin \beta_i^{G^i}$ for some $1 \leq i \leq m$, we conclude that there is no $g \in G$ such that $f(B) = B^g$

If we decide that having such g expressed as a word in elements of an SGS (or just its existence) is enough, the algorithm can be sped up to $O(t|B|)$:

- Instead of computing the products r_i of elements along the paths in Schreier trees, we just remember it as a word w_i
- By assumption $S = S^{-1}$, we have that $g = (w_1 \dots w_m)^{-1}$ is also a word in S

This procedure is also called “sifting as a word”.

Sifting as a word has another application. If we know a base of a permutation group in advance, the computation of an SGS can be sped up.

Theorem

Given a base B for some permutation group $G = \langle S \rangle \operatorname{Sym} \Omega$, $|\Omega| = n$, an SGS for G can be computed in $O(n|B|^2|S|log^3|G|)$ time. In particular, if a nonredundant base is known then an SGS can be computed by nearly linear-time algorithm.

Black-Box group representation

- Storing elements of G as base images makes computing products slow
- We can store them as words in an SGS (obtained by sifting)
- The length of such word is bounded by a sum of depths of the Schreier trees
- We have that G is isomorphic to a group H of such words in an obvious way. Let us denote the isomorphism by ψ

Lemma

For any $g \in G, h \in H$, we can compute $\psi(g)$ in $O(\log^c |G|)$ time and $\psi^{-1}(h)$ in $O(|\Omega| \log^c |G|)$ time