Disjoint direct product decomposition

C. Jefferson, M. Chang (2022)

Mainak Roy
IIT Kharagpur
2.9.2024

The problem to solve

Theorem

Let $H \leqslant \mathfrak{S}_n$ be a permutation group with orbits $\Omega_1, \ldots, \Omega_k$. There exists a unique finest partition P of the orbits such that we can write

$$H = \prod_{c \in P} H|_c$$

where $H|_c$ is the projection of H onto the set of points in the union of orbits in c.

Goal

Find an efficient algorithm to find the partition P.

Remark: All groups from here on are permutation groups.

Theorems, definitions, and stuff

Definition

A group K is called a *subdirect product* of $G = G_1 \times G_2 \times \ldots \times G_k$ if the projection maps $\rho_i : K \to G_i$ are surjective.

Theorem

Let H be a subdirect product of $G_1 \times G_2$. $H = G_1 \times G_2$ iff $1 \times G_2 \leqslant H$.

Proof.

Forward implication is obvious.
Backward implication: $\forall (g_1, g_2) \in H, \exists (1, g_2) \implies (g_1, 1) \in H$.
Now, $\rho_1(H) = G_1 \implies G_1 \times 1 \leqslant H$. $H = \langle G_1 \times 1, 1 \times G_2 \rangle$, hence proved. □

Theorems, definitions, and stuff

Theorem (Goursat's lemma)

Let H be a subdirect product of $G = G_1 \times G_2$. Let $\rho_i : G \to G_i$ for $i \in \{1, 2\}$ be the projection maps. Then, the following hold:

1. *Let $N_1 := \rho_1(Ker(\rho_2))$ and $N_2 := \rho_2(Ker(\rho_1))$. Then $N_1 \trianglelefteq G_1$ and $N_2 \trianglelefteq G_2$.*
2. *$\theta : G_1/N_1 \to G_2/N_2$ given by $N_1h_1 \to N_2h_2 \ \forall \ (h_1, h_2) \in H$ is an isomorphism.*
3. *(asymmetrical version) $\theta : G_1 \to G_2/N_2$ given by $h_1 \to N_2h_2 \ \forall \ (h_1, h_2) \in H$ is a surjective homomorphism.*

Lemma

Let T_2 be a transversal of N_2 in G_2. Let $\hat{\theta} : G_1 \to T_2$ be given by $\hat{\theta}(g_1) = t_2$ if $\theta(g_1) = N_2t_2$. Let $\mathcal{G} = \{((g_1, \hat{\theta}(g_1))|g_1 \in G_1\}$.

Then $H = \langle \mathcal{G}, 1 \times N_2 \rangle$.

Notation

- Let Δ be a union of some H-orbits. $h|_\Delta$ for $h \in H$ is the *restriction* of h to the points in Δ. $H|_\Delta = \{h|_\Delta | h \in H\}$.
- Fix an ordering of the orbits of $H \leqslant \mathfrak{S}_n$, say $\Omega_1, \Omega_2, \ldots, \Omega_k$. Let $G_i = H|_{\Omega_i}$. H is definitely a subdirect product of $G = G_1 \times \ldots \times G_k$.
- Let $\rho_i : G \to G_i$ for $i \in \{1 \ldots k\}$ be given by $g \to g|_{\Omega_i}$.
- For $I = \{i_1, \ldots, i_r\} \subseteq \{1 \ldots k\}$, let $P_I : G \to H|_{\bigcup_{i \in I} \Omega_i}$ be given by $g \to g|_{\Omega_{i_1}} g|_{\Omega_{i_2}} \ldots g|_{\Omega_{i_r}}$.
- Let $\Delta_i = \bigcup_{j \leqslant i} \Omega_j$.
- Let $\bar{i} = \{1 \ldots i\}$.
- Let $H_{(\Delta)}$ be the *pointwise stabilizer* subgroup of H of the points in Δ.

An iterative algorithm

We will consider the orbits one by one and at each step, we will maintain the current partition of the orbits such that we have a finest direct product decomposition.

Definition

At the end of the i-th step, we should have the finest direct product decomposition of $P_{\bar{i}}(H)$, which is $\mathcal{P}_i$. Define $\mathcal{P}_i = \langle C_1 | C_2 | \ldots | C_r \rangle$ to be a set partition of $\bar{i}$, such that

$$P_{\bar{i}}(H) = P_{C_1}(H) \times P_{C_2}(H) \times \ldots \times P_{C_r}(H)$$

Now, we will consider $P_{\overline{i+1}}(H)$ to be a subdirect product of $P_{\bar{i}}(H) \times \rho_{i+1}(H)$.

An iterative algorithm

We use points 1 and 3 of Goursat's lemma, and the lemma following it. We have the following:

Proposition

- *Let* $\rho_1 : P_{\overline{i+1}}(H) \to P_{\bar{i}}(H)$. *Let* $\rho_2 : P_{\overline{i+1}}(H) \to \rho_{i+1}(H)$.
- *Let* $N_{i+1} = \rho_2(Ker(\rho_1))$. *Note that* $Ker(\rho_1) = P_{\overline{i+1}}(H_{(\Delta_i)})$. *Also note that* $\rho_2(P_{\overline{i+1}}(H_{(\Delta_i)})) = \rho_{i+1}(H_{(\Delta_i)})$.
- *So,* $N_{i+1} = \rho_{i+1}(H_{(\Delta_i)})$, *and* $N_{i+1} \trianglelefteq \rho_{i+1}(H)$.
- $\theta_i : P_{\bar{i}}(H) \to \rho_{i+1}(H)/N_{i+1}$ *given by* $h_1 \to N_{i+1}h_2 \; \forall \; (h_1, h_2) \in P_{\overline{i+1}}(H)$ *is a surjective homomorphism.*
- *Let* T_{i+1} *be a transversal of* N_{i+1} *in* $\rho_{i+1}(H)$. *Let* $\hat{\theta}_i : P_{\bar{i}}(H) \to T_{i+1}$ *be given by* $\hat{\theta}_i(h_1) = t$ *if* $\theta_i(h_1) = N_{i+1}t$. *We require that* $1 \in T_{i+1}$.
- *Let* $\mathcal{G} = \{((h_1, \hat{\theta}(h_1))|h_1 \in P_{\bar{i}}(H)\}$. *Then* $P_{\overline{i+1}}(H) = \langle \mathcal{G}, 1 \times N_{i+1} \rangle$.

An iterative algorithm

Claim

At the i-th step, let $S_i = \{C_j | 1 \leqslant j \leqslant r, P_{C_j}(H) \times 1_{\bar{i} \setminus C_j} \nsubseteq Ker(\theta_i)\}$.
Then the finest disjoint direct product decomposition of $P_{\overline{i+1}}(H)$ *is:*

$$P_{\overline{i+1}}(H) = P_C(H) \times \prod_{C_j \notin S_i} P_{C_j}(H)$$

where $C = \{i+1\} \cup \bigcup_{C_j \in S_i} C_j$.

So, at each step, we need to find the set S_i.

Determining S_i (part 1. finding θ_i)

We will compute $\theta_i(P_{C_j}(H) \times 1_{\bar{i}\setminus C_j})$.Let X be a generating set for H. Then:

$$\theta_i(P_{C_j}(H) \times 1_{\bar{i}\setminus C_j}) = \{N_{i+1}\rho_{i+1}(x) | x \in X, P_{\bar{i}}(x) \in P_{C_j}(H) \times 1_{\bar{i}\setminus C_j}\}$$

We don't want to have to loop over all of X for every C_j. So we have to choose a generating set in a smart way.

Conjecture

For each i, $\exists$ a generating set X_i for H such that $\forall x \in X_i$, if $P_{\bar{i}}(x) \neq 1$ then $\exists$ a unique C_j such that $P_{\bar{i}}(x) \in P_{C_j} \times 1_{\bar{i}\setminus C_j}$.

Then:

$$\theta_i(P_{C_j}(H) \times 1_{\bar{i}\setminus C_j}) = \{N_{i+1}\rho_{i+1}(x) | x \in X_i, P_{C_j}(x) \neq 1\}$$

Now we only have to loop over X_i once for each i.

Interlude: Bases and strong generating sets

- A *base* for a group H is a set of points $B = [\beta_1, \ldots, \beta_m]$ such that $H_{(B)} = \mathrm{id}$.
- A base defines a sequence of groups $H = H^{[1]} \geqslant \ldots \geqslant H^{[m+1]} = \mathrm{id}$, where $H^{[i+1]} = \mathrm{Stab}_{\beta_i}(H^{[i]})$. This is called a *stabilizer chain*.
- Any element $h \in H$ is uniquely identified by its *base images* $[\beta_1^h, \ldots, \beta_m^h]$.
- A *strong generating set* for a group H with respect to the base B is a generating set of H such that $H^{[i]} = \langle X \cap H^{[i]} \rangle$.
- All of the above can be efficiently found by the Schreier-Sims algorithm.

Interlude: Sifting

The *sifting* procedure takes a base B for H and a permutation $g \in \mathfrak{S}_n$ and does the following:

- Let $H^{[i]}$ be the stabilizer chain defined by B.
- Iterate from 1 to m. Let T_i be a transversal of $H^{[i+1]}$ in $H^{[i]}$.
- Attempt to find an element $t_i \in T_i$ such that $\beta_i^{t_i} = \beta_i^g$. If found, set $g := gt_i^{-1}$. Otherwise, terminate. Also terminate when all i have been considered. The final result g' is called the *siftee* of g by B.
- $g = g'h$ for some $h \in H$.

Notice that if $g \in H$, then the g' is the identity permutation. Alternatively, if the base images of g' correspond to the identity of H, then the restriction of g' to H is the identity.

Determining S_i (part 2. is $\rho_{i+1}(x) \in N_{i+1}$?)

We will pick an *orbit-ordered* base B for H. Let Y be the associated strong generating set for H. Then notice that, for each $H_{(\Delta_i)}$, $\exists j_i$ such that $B_i = [\beta_{j_i}, \beta_{j_i+1}, \ldots, \beta_m]$ is a base for $H_{(\Delta_i)}$.

Proposition

1. *$N_{i+1} = \langle \rho_{i+1}(y) | y \in Y \cap H_{(\Delta_i)} \rangle$.*
2. *Let y' be the siftee of $y \in Y$ by B_i. $\rho_{i+1}(y') = 1$ iff $\rho_{i+1}(y) \in N_{i+1}$.*

Proof.

$B_i = [\beta_{j_i}, \ldots, \beta_m]$ is a base for $H_{(\Delta_i)}$.$B_{i+1} = [\beta_{j_{i+1}}, \ldots, \beta_m]$ is a base for $H_{(\Delta_{i+1})}$. Then, since elements of a group are in bijection with the base images, we can say $B = [\beta_{j_i}, \ldots, \beta_{j_{i+1}-1}]$ is a base for $\rho_{i+1}(H_{(\Delta_i)}) = N_{i+1}$. Assuming $\rho_{i+1}(y') = 1 \implies$ all base images of y' in B correspond to the identity. Furthermore, $\rho_{i+1}(y) = \rho_{i+1}(y')p$ for some $p \in N_{i+1}$. Therefore, $\rho_{i+1}(y) \in N_{i+1}$. □

Determining X_i

To make the test work, we must make sure that the X_i are all strong generating sets with respect to B.

Proposition

Let X_1 = a strong generating set of H with respect to B. Then

$$X_{i+1} = \{x | x \in X_i, x \in H_{(\Delta_i)}\} \cup \{sift(B_i, x) | x \in X_i, x \notin H_{(\Delta_i)}\}$$

is such that:

1. X_{i+1} *is a strong generating set for* H *with respect to* B.
2. $\forall x \in X_{i+1}$, *if* $P_{\overline{i+1}}(x) \neq 1$ *then* $\exists$ *a unique* C_j *such that* $P_{\overline{i+1}}(x) \in P_{C_j} \times 1_{\overline{i+1} \setminus C_j}$.

We have determined S_i and X_i for all i. So we are done.

The final algorithm

Algorithm 1 disjoint direct product decomposition

Require: Group H with orbits $\Omega_1, \ldots, \Omega_k$

Ensure: P is the finest partition of H-orbits

$\mathcal{P} \leftarrow$ *disjoint_set*$(1 \ldots k)$
$B \leftarrow$ concatenate$(\Omega_1, \ldots, \Omega_k)$
$X \leftarrow$ strong_generating_set(B)
for $i = 1$ to $k - 1$ **do**
    compute_next_partition$(\mathcal{P}, X, B)$
    $X \leftarrow$ compute_next_sgs(X, B)
end for

The final algorithm

Algorithm 2 compute next partition

Require: $\mathcal{P}$ is partition of first i orbits
Require: a strong generating set X_i and the base B
Ensure: $\mathcal{P}$ is partition of first $i+1$ orbits

for $x \in X_i$ **do**
    if $P_{\bar{i}}(x) \neq 1$ **then**
        if $\rho_{i+1}(\text{sift}(B_i, x)) \neq 1$ **then**
            $\mathcal{P}.\text{union}(\text{find\_cell}(y), i+1)$
        end if
    end if
end for

The final algorithm

Algorithm 3 compute next sgs

Require: a strong generating set X_i and the base B
Ensure: a strong generating set X_{i+1}

$X_{i+1} = \{\}$
for $x \in X_i$ **do**
    if sift$(B_i, x) \neq 1$ **then**
        X_{i+1}.append(sift(x, B_i))
    else
        X_{i+1}.append(x)
    end if
end for
return X_{i+1}

The algorithm is available in Sage as a method of PermutationGroup named disjoint_direct_product_decomposition.
The algorithm works in time polynomial in $n \cdot |X|$.

References

[1] M. Chang, C. Jefferson, Disjoint direct product decompositions of permutation groups, Journal of Symbolic Computation, Volume 108, 2022, Pages 1-16, ISSN 0747-7171, https://doi.org/10.1016/j.jsc.2021.04.003. arxiv:2004.11618