数理解析研究所講究録 1456 巻 2005 年 35-39

置換群の可移拡大の計算法

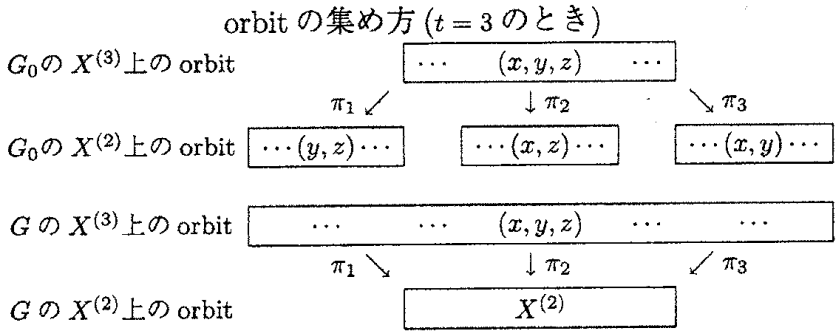
宮本 泉
MIYAMOTO IZUMI
山梨大学
UNIVERSITY OF YAMANASHI*

1 置換群の可移拡大

G は集合 $X=\{0,1,2,\cdots,n\}$ 上の可移な置換群, $G_0=\{\, g\in G\,|0^g=0\,\}$, G の点 0 の固定部分群とする. G_0 が先に与えられているとき, 群 G をその可移拡大という. G_0 は $X\backslash\{0\}$ 上に $(t-2)$-重可移で, $(t-1)$-重可移ではないとする $(t\geq 3)$. このとき, 可移拡大 G は X 上 $(t-1)$-重可移で, t-重可移にはならない.

$X^{(l)}=\{(x_1,x_2,\cdots,x_l)|x_i\in X,$ 相異なる $\}$ とおく. $g\in G$ は $(x_1,x_2,\cdots,x_l)\longrightarrow(x_1^g,x_2^g,\cdots,x_l^g)$ で $X^{(l)}$ に作用する. このとき,G は $X^{(t-1)}$ 上可移で, $X^{(t)}$ 上可移ではない. 下のアルゴリズムによって G の $X^{(t)}$ 上の可能な orbit が構成可能であることを, 次数の小さいとき [1] の実験で示した. 群論を応用して, 次数のより大きい置換群の場合にも計算可能になった.

アルゴリズム :
G_0 が $X^{(t)}$ に作用したときの orbit を集めて, G の $X^{(t)}$ 上の可能な orbit を構成する方法. その自己同型群として可移拡大 G に近い群を得る.



G_0 の $X^{(2)}$ と $X^{(3)}$ 上の任意の orbit を R'^2 と R'^3 とするとき,

$$\text{すべての } (x,y)\in R'^2 \text{ に対して } |\pi_j^{-1}((x,y))\cap R'^3|=\text{ 定数.}$$

G についても同様で, この定数に関する条件が合うように G_0 の orbit を集めて G の orbit の候補を計算する. さらに, ある orbit の $\{x,y,z\}$ の成分の順序を一斉に置換したものも, また,orbit になるということも考慮する. このような組合せ構造は次に定義する superscheme となる.

*imiyamoto@yamanashi.ac.jp

2 Superschemes

X 上の置換群が X の t 点対の集合 X^l, $1 \le l \le t$, に作用したときの orbit の与える $X^{(l)}$ の分割のなす組合せ構造を superscheme を使って表す。
定義. (X, Π) が t-superscheme であるとは、$\Longleftrightarrow$

S1. $\Pi = \{\Pi^1, \Pi^2, \cdots, \Pi^t\}$, $t \ge 2$, で Π^l は $1 \le l \le t$ において X^l の分割,

S2. $\sigma \in Sym(l)$ について, $\sigma((y_1, y_2, \cdots, y_l)) = (y_{\sigma(1)}, y_{\sigma(2)}, \cdots, y_{\sigma(l)})$ とおき, $\Pi^l = \{R_0^l, R_1^l, \cdots, R_{d_l}^l\}$, $1 \le l \le t$, とおくと, すべての R_k^l と $\sigma \in Sym(l)$, に対して、$\sigma(R_k^l) \in \Pi^l$ $\cdots$(*symmetric*)

S3. *projection* $\pi_j^l : X^l \to X^{l-1}$ を
$\pi_j^l((y_1, y_2, \cdots, y_l)) = (y_1, y_2, \cdots, y_{j-1}, y_{j+1}, \cdots, y_l)$ で定義すると,
すべての $R_k^l \in \Pi^l$, $2 \le l \le t$ に対して, $\pi_j^l(R_k^l) \in \Pi^{l-1}$,

S4. すべての R_k^l, $2 \le l \le t$, と, すべての $\boldsymbol{y} = (y_1, y_2, \cdots, y_{l-1}) \in \pi_j^l(R_k^l)$ に対して, constant $p_{k,j}^l = |(\pi_j^l)^{-1}(\boldsymbol{y}) \cap R_k^l|$ が存在. 特に, $p_{k,j}^l = |R_k^l| / |\pi_j^l(R_k^l)|$. $\cdots$(*regular*)

アソシエーションスキームは, 2-superscheme であって, $\Pi^1 = \{X\}$ かつ, Π^2 から誘導される X^3 の分割を合わせると 3-superscheme になる. 構成された superscheme が可移拡大を与えるかどうかを簡易チェックするためにアソシエーションスキームを使用している.

3 メモリ使用量の節約

superscheme の分割 Π^{l-1} から誘導される X^l の分割 = 次の同値関係で定義される分割
$\boldsymbol{x}, \boldsymbol{y} \in X^l$

$$\boldsymbol{x} \sim \boldsymbol{y} \Longleftrightarrow \pi_j^l(\boldsymbol{x}), \pi_j^l(\boldsymbol{y}) \in R_{s(j)}^{l-1} \text{ for } 1 \le j \le l$$

. G_0 と G が $X^{(t)}$ に作用したときの orbit による $X^{(t)}$ の分割を,

$$\Pi'^{(t)} = \{\underbrace{R'_1, R'_2, \cdots\cdots\cdots\cdots, R'_{r'}}\} \cdots G_0\text{の orbit (既知)}$$
$$\Pi^{(t)} = \{ \quad R_1, \quad R_2, \ \cdots \ , R_r \ \} \cdots G \text{ の orbit (未知)}$$

とすると,$\{1, 2, \cdots, r'\}$ の適当な分割 $Y = \{Y_1, Y_2, \cdots, Y_r\}$ が存在して,G_0 の $X^{(t-1)}$ 上の各 orbit $R_s'^{(t-1)}$ に対して, 次のようになる.

$$R_k = \bigcup_{l \in Y_k} R'_l \implies p_{k,j}^t = \sum_{l \in Y_{k,j,s}} p_{l,j}'^t, \quad (1 \le k \le r,\ 1 \le j \le t).$$

ただし, $Y_{k,j,s} = \{l \in Y_k | R_s'^{(t-1)} = \pi_j^t(R'_l)\}$. また, G は $(t-1)$-重可移より,

$$\begin{aligned} &\pi_j^t(R_k) = \{X^{(t-1)}\}, \\ &p_{k,j}^t = |R_k| / |X^{(t-1)}|, \end{aligned} \quad (1 \le j \le t,\ 1 \le k \le r).$$

可移拡大 G の $X^{(t)}$ 上の orbit R_k の個数 r とサイズは, 次の良く知られた置換群の命題からわかる.

命題. G_0 は $(X \backslash \{0\})^{(t-1)}$ 上、r 個の orbit $R_k'^{t-1}$, $1 \le k \le r$, をもつ. そして、$|R_k^t| = |X| \cdot |R_k'^{t-1}|$. 特に、

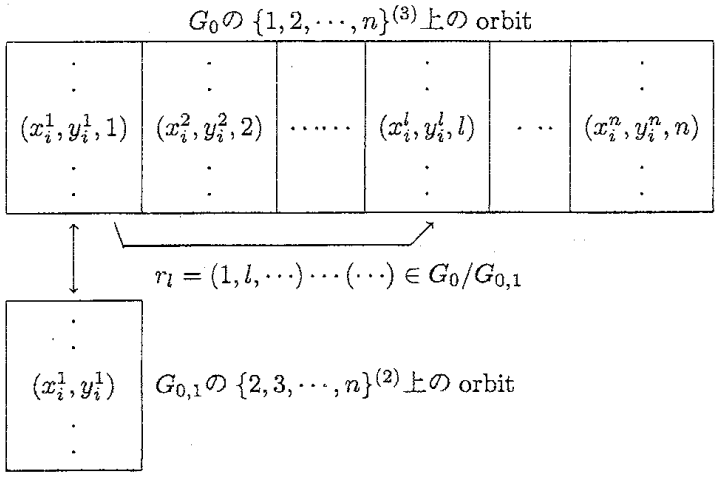
$$p_{k,j}^t = |R_k'^{t-1}| / |(X \backslash \{0\})^{(t-2)}| \text{ for } 1 \le k \le r \text{ and } 1 \le j \le t.$$

証明. orbit の適当な順序で下のようになる.

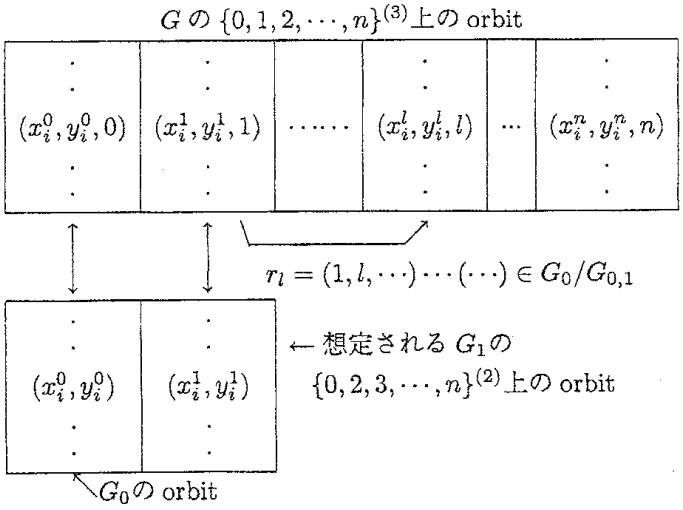
$$\{\pi_t^t(\boldsymbol{x})|\boldsymbol{x}=(x_1,x_2,\cdots,x_{t-1},0)\in R_k^t\}=R_k^{\prime t-1} \text{ for } 1\le k\le r.$$

構成アルゴリズムでは, この命題のほか,π_j^t の性質と, $Sym(t)$ の G と G_0 の orbit 番号 $\{1,2,\cdots,r\}$ と $\{1,2,\cdots,r'\}$ への作用の比較から, 可能な分割 Y を計算している.
以上より,G_0 に関するデータは, $p_{l,j}^{\prime t}$, $R_{s(j)}^{\prime t-1}=\pi_j^t(R_l^{\prime t})\in\Pi^{t-1}$, $(1\le j\le t)$, $R_{\sigma_i(l)}^{\prime t}=\sigma_i(R_l^{\prime t})\in\Pi^{\prime t}$, $(\sigma_i=(i,i+1)\in Sym(t))$ となる. 簡単に書けば, $p_{l,j}^{\prime t}$, $\pi_j^t(l)$, $\sigma_i(l)$ となる.

今回は,G_0 の $X^{(t)}$, $t\ge 3$, 上の orbit 計算に必要になるメモリ量を節約するために, 上の命題を一般化した形で適用した. 以下の説明では, $t=3$ としている.



データサイズの比較は, G_0 による $X^{(3)}$ の orbit 分割は約 n^3 のサイズ, 一方, $G_{0.1}$ による $(X\backslash\{0\})^{(2)}$ の orbit 分割は約 n^2 のサイズ, コセットの代表元たち $r_l\in G_0/G_{0,1}$, $(l=1,2,\cdots n)$, n 点上の置換が n 個は約 n^2 のサイズとなる. 以上より, データサイズの節約は $n^3\longrightarrow \text{const}\times n^2$ が可能となる.
また, G_0 の orbit 番号で G の orbit の候補が求めているところでも同様にメモリ使用の節約を行う.



データサイズは, G による $X^{(3)}$ の orbit 分割が約 n^3, G_0, G_1 による $X^{(2)}$ の orbit 分割が, それぞれ約 n^2, $G_0/G_{0,1}$ の代表元たちが約 n^2 となる.

このようなメモリの節約を行ったとき, G_0 の $X^{(3)}$ 上の orbit に関する情報の計算は, やや面倒だが, 1 回計算するだけなので, 影響は少ない. しかし, G の $X^{(3)}$ 上の orbit の作る superscheme の自己同型の計算. バックトラックで計算するので, くり返し必要な部分を G_1 の orbit から $r_l \in G_0/G_{0,1}$ を使って構成する必要がある.

アソシエーションスキームを使った簡易計算:
G_1 による $(X\backslash\{1\})^2$ の orbit 分割は, 群からつくられると想定しているので, 2-superscheme であるばかりでなく, アソシエーションスキームとなっている必要がある. この $(X\backslash\{1\})^2$ の分割が, アソシエーションスキームになっているかどうか調べる. また, その自己同型群が G_1 に近い場合は, これから可移拡大を求めることも可能となる. アソシエーションスキームの計算は n^2 のデータサイズの直接計算になる.

以上, 説明は $t=3$ で行ったが, 自己同型の計算を除き, 一般の t で計算が可能になっている. 自己同型の計算は,$t>3$ のときは何点かを固定してできる 3-superscheme の自己同型を計算している.

4 計算実験

31 次から 999 次までの Primitive な置換群のデータ (GAP のライブラリ) を G_0 として, その群の $(X\backslash\{0\})^{(t)}$ 上の orbit の個数が 50 以下の場合について, 実験を行った. 下の表で,superscheme の構成にかかった時間は, 全体で 1 日程度であった. 使用した計算機は Pentium III 800Mhz, メモリ 256MB, Linux のもとで,GAP を使用して計算した.

	全体	99 次以下	100 次以上
Primitive な群	5185	540	4645
条件を満たす群	872	210	662
答が空でない群	115	72	43
アソシエーションスキーム	31	15	16
拡張がある群	25	9	16

自己同型計算実験データ

G_0	次数	G	答の個数	計算時間	以前の時間
$U_3(5)$	175	HS	33	7 分	20 分 *
McL	275	Co.3	1	1 分	1 時間 *
$G_2(3)$	351	–	2	120 分	–
$O_{10}^+(2)$	527	$Sp_{10}(2)$	1	25 分	–
$PSL_5(2)$ on P^2	961	–	1	80 分 ♮	–
$(PSL_3(31))_0^\sharp$	992	$PSL_3(31)$	1	8 時間	–

(注) * superscheme で特定の 1 点を含む部分からできるアソシエーションスキームの自己同型を利用して拡大を構成. ♮ 拡大の候補となる自己同型 1 個または拡大が無いことのみを計算. ♯ primitive ではない. HS, Co.3 の 1 点固定群は $U_3(5).2$, McL.2.

可移拡大の正確な計算

superscheme の自己同型として可移拡大に近い群を求めている. 例えば, 可移拡大で PSL を求めるとき, superscheme の自己同型は $P\Gamma L$ になってしまう. $G_{0,1}$ の正規化群の中から Witt の Lemma を満たす元を探す方法で, 正確な可移拡大が得られる.

自己同型計算の表中の $*$ アソシエーションスキームの自己同型を利用する方法では，$PSL_3(31)$ の場合などで，アソシエーションスキームの自己同型群は計算できるが，大きくなり過ぎて，その中での正規化群の計算ができない．

Orbit の計算法

X 上の可移な群 G の $X^{(2)}$ の計算に関して，GAP による直接計算 Orbits(G, Arrangements(X, 2), OnTuples); と前に示した命題を応用する方法を比較してみたら，計算時間にかなりの差があったので，その結果を以下に示す．

群	次数	GAP の直接計算	命題利用の計算	確認
$G_2(3)$	351	14 秒	3 秒	3 秒
$O_{10}^{+}(2)$	527	63 秒	6 秒	9 秒
$3^6 \cdot 55296$	729	900 秒	11 秒	23 秒
$PSL_3(31)$	993	115 分	20 秒	47 秒

参 考 文 献

[1] 宮本泉．アソシエーションスキームの拡張と 2 重可移群の計算．*Computer Algebra – Algorithms, Implementations and Applications*, 数理解析研究所講究録 1395 185–189 (2004).