数理解析研究所講究録
第811巻 1992年 140-143

Representation of finite groups with CAYLEY

千葉大学　自然科学研究科　脇　克志 (Katsushi WAKI)

1. CAYLEY について

CAYLEY は、オーストラリアの J.Cannon によって作られた代数構造（群、環、体、ベクトル空間、etc ）を計算するためのコンピュータ言語です。　その目的は代数構造の定義とその解析および特に有限群のデータベース化にあります。　今回はこの CAYLEY の簡単な説明と実際の利用例を紹介したいと思います。（言語としての CAYLEY の詳しい説明は [1] を参照してください）　まず具体的な例から示します。　例えば次の様にして代数構造を定義することが出来ます。

```
G:symmetric(6);          : 6 次の対称群を G とする。
H:derived subgroup(G);   : G の交換子群を H とする。
F:field(3);              : 3 つの元からなる有限体を F とする。
VS:vector space(3,F);    : 体 F 上の 3 次元ベクトル空間を VS とする。
MR:matrix ring(VS);      : ベクトル空間 VS に作用する行列環を MR とする。
```

また print 命令を使うことで、定義された代数構造についてのいろいろな情報を見ることも出来きます。

```
print order(G);          : 群 G の位数を表示する。
print classes(G);        : 群 G の共役類の表を表示する。
print simple(G);         : 群 G が単純群であるかないかを判断する。
```

このように定義と解析を繰り返すことが CAYLEY の最も基本的な使い方となります。

2. コンピュータ言語としての CAYLEY の特徴

さて次に他の数式処理言語にない CAYLEY の特徴を示します。

(a) 命令語が長い（代数で使われている名称をそのまま使っている）

専門語をそのまま使っているので代数学の専門家であってコンピュータの専門家でない人でも特殊な命令を覚えずに、CAYLEY を扱うことが出来きます。

(b) of 文（全ての要素は代数構造か代数構造の元でなけらばならない）

例えば、 x=1; として print x+x; を実行すれば、 2 と答えがかえってきます。 しかし F:field(2); x=1 of F; とした場合 print x+x; の答えは 0 になってしまいます。 このようにバックグラウンドとなる代数構造の違いにより計算結果も変ってくるわけです。 また正方行列の場合はそれを含むような行列環が存在するので、問題ないのですが、行と列の長さの違う行列の場合はこの行列を含むような代数構造がないため CAYLEY では今のところ定義することが出来ません。

(c) for each 文（任意の集合や列で loop が出来る）

他の数式処理の for-next 文でもインデックスを付けることで集合や列の loop を作ることが出来きますが、これも (a) と同様にコンピュータの専門家でない人がプログラムを作る上で余計なことを考えずに済むわけです。

(d) library（よく知られている代数構造がファイルとして利用できる）

現在までのバージョンアップで caylibs と呼ばれているディレクトリにたくさんの群がファイルとして貯めこまれてきています。 そのため使用者は自分の調べたい群の名前のファイルを CAYLEY から呼び出すだけで済みます。 例えば、 caylibs の中の simgps というディレクトリの中には、群の位数が 10^6 以下の単純群がすべて納められています。

3. CAYLEY の利用例

ここでは、板内先生の問題を使って CAYLEY の具体的使用例を紹介します。 有限群の表現とは、体 F と有限群 G が与えられたとき G から $GL(n, F)$ （ただし n は自然数）への準同型写像のことです。 表現 R が一つ決まるとこれを使って F上の n 次元のベクトル空間 V に G の作用を定義でき V を G-加群と見ることが出来ます。 いま表現 R に対応する G-加群 V について自分自身と 0 以外に部分 G-加群を持たないとき、 R は既約な表現と呼びます。

体 F を複素数体にして既約な表現 R が与えられたとき G の元 x に対して複素数 $tr(R(x))$ を対応させる有限群 G から複素数体への写像を既約指標と呼びます。 また一般論から既約指標の個数は、 G の共役類の個数と等しいことが分かります。 さらに $\{C_1, ..., C_k\}$ を

G の共役類、$\{\chi_1, ..., \chi_k\}$ を G の既約指標、1_G を G の単位元とすると一般に

$$|G| = \sum_{i=1}^{k} |C_i| = \sum_{i=1}^{k} \chi_i(1_G)^2$$

が成り立ちます。(有限群の表現の詳しい説明は [2] を参照してください)

さてここで C_i, χ_i のインデックスを付け替えることで

(*) $$|C_i| = \chi_i(1_G)^2 \qquad (1 \leq i \leq k)$$

となるような群はどんな性質を持つかという問題が出来ます。(坂内先生の問題) 今までに Suzuki 2-groups の中に上の条件を満たす群があることが知られています。 そこで CAYLEY を使って、この Suzuki 2-groups 以外で上の条件を満たす群を探してみましょう。

ディレクトリ caylibs の中に twogps という名のディレクトリがあります。 ここには位数が 2 から 256 までのすべての 2-groups が5つの自然数の列を使って定義されています。

第1の数はこの群の最少の生成元の個数、第2の数は Power Commutator-Presentation (以下 PC-P と呼ぶ。) という群を定義する方法で使う生成元の個数、第3の数は群の the exponent p-class 、第4の数と第5の数は、PC-P の方法に基づいた群の関係式を数値化したものです。例えば、CAYLEY からディレクトリ twogps の中の gps64 というファイルを読み込むと gps という変数に群の位数が 64 の群たちが列として納められることになります。(ただしそれぞれの群は上で述べた5つの自然数の列の形になっている) そして5つの自然数の列は genrat と呼ばれるプログラムによって群に変換する事が出来ます。

さて有限群 G が abelian の場合には条件 (*) は常に満たされるので G は non-abelian としておきます。 ここで上で述べたディレクトリ twogps から位数が 8 から 128 までの群 (全部で 2665 個) に対し条件 (*) を満たすかどうかを調べてみましょう。

まず与えられた有限群 G の中で $|G/G'| = |Z(G)|$ となるものだけを選びます。(ここで G' は G の交換子群また $Z(G)$ は G の中心を表わす) この式は $\chi_i(1_G) = 1$ となるものと $|C_i| = 1$ となるのもの個数が等しいことを意味します。 これは次のようなプログラムで実行できます。

```
 1:     sq=empty;
 2:     for each gp in gps do
 3:          g=genrat(gp);
 4:          gn=order(g);
 5:          cn=order(center(g));
 6:          dn=order(derived subgroup(g));
 7:          if cn eq gn/dn and not(abelian(g)) do
 8:              sq=append(sq,g);
 9:          end;
10:    end;
```

1 行目で sq を空の列にします。 2 行目から gps の中の５つの自然数の列 gp で loop します。 3 行目で５つの自然数を群 G に変換します。 4,5,6 行目でそれぞれ群 G の位数 gn 、 G の中心の位数 cn 、 G の交換子群の位数 dn をそれぞれ求めます。 最後にもし cn=gn/dn のときで G が abelian でないとき列 sq に G をつけ加えます。

そしてこの sq の中の群についてその既約指標を計算して条件 (*) を満たす群を見つけるわけです。

その結果、位数 64 の群の中に 10 個、位数 128 の群の中には 41 個の条件 (*) を満たすものを見つけることが出来ました。 また位数 100 以下のすべての群を見ても条件 (*) を満たすものは上の位数 64 の群だけであることも分かりました。 さらにこの位数 64 $(= 2^6)$ の群に注目した結果、一般に位数が p^6 (p は素数) で $G' = Z(G)$ である special group はつねに条件 (*) を満たすことが分かりました。

このように群論や表現論の世界にも少しづつ計算機が利用されるようになってきています。

現在まだ本格的に計算できる群の位数が小さかったり計算に時間がかかったりしますが、アルゴリズムの改良や library 充実によりもっといろいろな利用が可能になると思います。

参考文献

[1] Cannon,J.J. , An introduction to the group theory language CAYLEY, (Atkinson,M.,ed) Computational Group Theory. London: Academic Press, 145-183,(1984).

[2] 永尾 汎、 津島 行男、有限群の表現、数学選書８、 裳華房.