リー代数の計算の楽しみ

宇佐見 公輔

2019 年 10 月 19 日

自己紹介

職業：プログラマ / 趣味：数学

関西日曜数学友の会での発表履歴：

- Generalized Onsager algebras（第 5 回 / 2019 年 8 月）
- ルート系とディンキン図形（第 4 回 / 2019 年 4 月）
- ラムダ計算の話（第 3 回 / 2018 年 11 月）
- 圏論と Haskell（第 2 回 / 2018 年 8 月）

執筆参加：

- 数学デイズ大阪編：低次元のリー代数をみる（Kindle 版発売中）

リー代数

ベクトル空間とリー代数

ベクトル空間 ＝「加法」と「スカラー倍」

リー代数 ＝ ベクトル空間 ＋ 第 3 の演算「ブラケット積」

ブラケット積が満たすべき条件

1. $[ax + by, z] = a[x, z] + b[y, z],$

   $[z, ax + by] = a[z, x] + b[z, y]$（双線型性）

2. $[x, x] = 0 \ (\Longrightarrow \ [x, y] = -[y, x])$（交代性）

3. $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$（Jacobi identity）

抽象的に与えられたリー代数

いくつかの「生成元」を用意して、加法、スカラー倍、ブラケット積をほどこして得られるものの集合を考える。

A_1：生成元と関係式で与えられたリー代数

生成元：$e,\ f,\ h$

関係式：$[e,f]=h,\quad [h,e]=2e,\quad [h,f]=-2f$

これはどのようなリー代数か？

「どのようなリー代数か」とは

何が分かったらいいのか？

- リー代数はベクトル空間なのだから、基底が知りたい。
- そして、その基底同士のブラケット積が知りたい。

特に、行列のリー代数であらわすことができれば分かりやすい。

行列のリー代数

$\mathfrak{gl}(n, \mathbb{C})$

$\mathbb{C}$ 成分の n 次正方行列がなすベクトル空間 + 次のブラケット積

$$[X, Y] := XY - YX$$

このブラケット積の定義は、リー代数の条件を満たしていることが確認できる。

抽象的なリー代数と行列のリー代数との対応

$\mathfrak{sl}(2, \mathbb{C})$

$$\mathfrak{sl}(2, \mathbb{C}) := \{X \in \mathfrak{gl}(2, \mathbb{C}) \mid \mathrm{tr}(X) = 0\}$$

これは 3 次元のベクトル空間で、以下の E, F, H を基底に持つ。

$$E := \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad F := \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad H := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$[E, F] = H, \quad [H, E] = 2E \quad [H, F] = -2F$$

$\mathfrak{sl}(2, \mathbb{C})$ は、生成元と関係式のリー代数 A_1 と同型である。

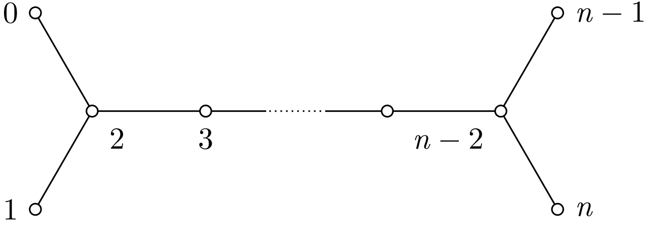
実際に調べたいリー代数

$D_n^{(1)}$ 型 Onsager 代数

生成元：$e_0, e_1, \ldots, e_n$

関係式：

- $[[e_i, e_j], e_j] = e_i$ (i と j が $D_n^{(1)}$ 型 Dynkin 図形で隣り合う)
- $[e_i, e_j] = 0$ (otherwise)



おおまかな予想

$A_n^{(1)}$ 型 Onsager 代数が、loop 代数 $\mathbb{C}[t,t^{-1}]\otimes\mathfrak{sl}(n,\mathbb{C})$ の部分代数として具体的に実現できることは、先人の結果で分かっていた。
$D_n^{(1)}$ 型 Onsager 代数は、loop 代数 $\mathbb{C}[t,t^{-1}]\otimes\mathfrak{o}(2n,\mathbb{C})$ の部分代数として書けるだろうと予想できた。

（実際、これは正しかった。
参考：関西日曜数学友の会第 5 回 Generalized Onsager algebras）

どうやって調べるのか？

これを調べる時点で理論的な背景は分からなかったので力技。

同型になる部分代数を探して、基底を見つけたい。そのために、

行列同士のブラケット積をたくさん計算する必要がある。

行列を直接計算しても良いが、以下の関係を利用してみる。

$\mathfrak{gl}(n, \mathbb{C})$ の基底のブラケット積

E_{ij} : (i, j) 成分だけ 1 で他は 0 の行列

$$[E_{ij}, E_{kl}] = \delta_{jk} E_{il} - \delta_{il} E_{kj}$$

パターンマッチ

ある式の中に $[E_{ij}, E_{kl}]$ という形を見つけたら、機械的に $\delta_{jk}E_{il} - \delta_{il}E_{kj}$ に置き換えることができる。

このルールだけあれば、E_{ij} が行列をあらわしたものであることは忘れてしまってもいい。

この置き換えは、プログラミングで言う「パターンマッチ」で処理できるのではないか？

その考えに基づいてプログラムコードを書いてみる。

Mathematica を使う

Mathematica によるパターンマッチプログラム

```
LieBracket[e[i_,j_],e[k_,l_]] :=
    KroneckerDelta[j,k] e[i,l]
    - KroneckerDelta[i,l] e[k,j];
LieBracket[e[1,2],e[2,3]] (* = e[1,3] *)
LieBracket[e[4,5],e[5,4]]
(* = e[4,4] - e[5,5] *)
```

D 型の基底

$\mathfrak{o}(2n, \mathbb{C})$

$$\mathfrak{o}(2n, \mathbb{C}) := \{X \in \mathfrak{gl}(2n, \mathbb{C}) \mid X^\intercal S + SX = 0\}$$

(S : (i, j) 成分が $i + j = 2n + 1$ のときだけ 1 で他は 0 の行列)

$\mathfrak{o}(2n, \mathbb{C})$ の基底のブラケット積

$$G_{ij} := E_{ij} - E_{2n+1-j,2n+1-i}$$

$$\begin{aligned}[G_{ij}, G_{kl}] =& \delta_{jk} G_{il} - \delta_{il} G_{kj} \\ &+ \delta_{2n+1-j,l} G_{k,2n+1-i} - \delta_{2n+1-i,k} G_{2n+1-j,l}\end{aligned}$$

D 型の計算

D 型を計算するパターンマッチプログラム

```
G[i_,j_] := 0 /; i+j == 2n+1;
G[i_,j_] := - G[2n+1-j,2n+1-i] /; i+j > 2n+1;
LieBracket[G[i_,j_],G[k_,l_]] :=
    KroneckerDelta[j,k] G[i,l]
    - KroneckerDelta[i,l] G[k,j]
    + KroneckerDelta[2n+1-j,l] G[k,2n+1-i]
    - KroneckerDelta[2n+1-k,i] G[2n+1-j,l];
LieBracket[G[1,2],G[2,3]] (* = G[1,3] *)
```

例：生成元の表現を探す

```
e[i_]:= t[1]G[2n-1,1] + t[-1]G[1,2n-1] /;i==0;
e[i_]:= G[i,i+1] + G[i+1,i] /;1<=i<=n-1;
e[i_]:= G[n-1,n+1] + G[n+1,n-1] /;i==n;
LieBracket[e[1],e[2]]
(* = G[1,2] + G[2,1] *)
LieBracket[e[1],e[2],e[3]]
(* = G[1,3] - G[3,1] *)
LieBracket[e[1],e[2],e[3],e[4]]
(* = G[1,4] + G[4,1] *)
```

まとめ

リー代数の計算の手助けとして、Mathematica プログラミングを活用した。

行列を計算する代わりにパターンマッチを活用することで、計算の見通しも良くなった。様々な組み合わせを試した結果、生成元と基底をローラン多項式＋行列で表現できた。（Date, Usami, On an analog of the Onsager algebra of Type $D_n^{(1)}$）

プログラミングを数学研究に活用するのも楽しい！