

数理漢文学のススメ

島野達雄

1. 多くを語るカタラン数

1-1. 第四の数列

はじめに等差数列, 等比数列, フィボナッチ数につぐ (?), 第四の数列「カタラン数」を紹介しよう. それぞれ次のような漸化式 (差分方程式) で定義する.

$$\text{等差数列 } a_n = a_{n-1} + d \quad (d \text{ は公差})$$

$$\text{等比数列 } b_n = r b_{n-1} \quad (r \text{ は公比})$$

$$\text{フィボナッチ数 } F_{n+2} = F_n + F_{n+1} \quad (F_0 = 0, F_1 = 1)$$

$$\text{カタラン数 } c_n = c_0 c_{n-1} + c_1 c_{n-2} + \cdots + c_{n-2} c_1 + c_{n-1} c_0 = \sum_{i=0}^{n-1} c_i c_{n-i-1} \quad (c_0 = 1)$$

前の三つの数列は, 線形の漸化式になっているが, カタラン数は二つの項の「積」を使った非線形の漸化式になっている. $\sum_{i=0}^{n-1} c_i c_{n-i-1}$ を, 数列 $\{c_n\}$ の c_0 から c_{n-1} までの (または 0 から $n-1$ までの) 畳み込みとよぶ.

実際にカタラン数を求めてゆくと,

$$c_0 = 1$$

$$c_1 = c_0 c_0 = 1 \cdot 1 = 1$$

$$c_2 = c_0 c_1 + c_1 c_0 = 1 \cdot 1 + 1 \cdot 1 = 2$$

$$c_3 = c_0 c_2 + c_1 c_1 + c_2 c_0 = 1 \cdot 2 + 1 \cdot 1 + 2 \cdot 1 = 5$$

$$c_4 = c_0 c_3 + c_1 c_2 + c_2 c_1 + c_3 c_0 = 1 \cdot 5 + 1 \cdot 2 + 2 \cdot 1 + 1 \cdot 5 = 14$$

$$c_5 = c_0 c_4 + c_1 c_3 + c_2 c_2 + c_3 c_1 + c_4 c_0 = 1 \cdot 14 + 1 \cdot 5 + 2 \cdot 2 + 5 \cdot 1 + 14 \cdot 1 = 42 \quad \dots$$

となる. つまり, 数列 $\{c_n\} = \{1, 1, 2, 5, 14, 42, \dots\}$ となる.

1-2. 返り点が生み出すパターン

じつは, カタラン数は, 「 n 字の漢字列にレ点, 一二点, 上下点などの返り点をつけるパターンの総数」になっている. ($n=0$ のときはパターン数を 1 とする).

たとえば, 漢字 1 字の A には, 返り点の付けようがなく, A のままの 1 通りしかない.

AB の 2 字には, レ点の有無によって AB と A_レB の 2 通りのパターンがある. A_レB は BA の順に読むので, ここでは A_レB = BA と書くことにする (以下, 同様).

ABC の 3 字には, ABC = ABC, AB_レC = ACB, A_レBC = BAC, A_レBC_レ = BCA, A_レB_レC = CBA の 5 パターンがある. ここで, 全部で $3 \times 2 \times 1 = 6$ 通りある ABC の並べ替えのうち CAB は, 「返り点だけからは生じない」ことを注意しておく. 豎点 (たててん, と読む. ハイフンのこと) を使い, A-B を熟語としてあつかうと, A₂-BC₁

=CABのように返り点の一二点と堅点でCABを生み出すことができる。(まぎらわしさを避けるため、以下、一二点は漢数字ではなく、算用数字であらわす。)

同じようにABCDの4字には、ABCD=ABCDからA_↓B_↓C_↓D=DCBAまでの14通りが返り点だけで生じる。

ABCDEの5字には、ABCDE=ABCDEからA_↓B_↓C_↓D_↓E=EDCBAまでの42通りが返り点だけで生じる。

こうして数列{ c_n } = {1,1,2,5,14,42,...}つまりカタラン数が生み出される。

1-3. カタラン数の母関数

一般に、数列{ a_n }に対して、各項に x^n ($n = 0,1,2,\dots$)を掛け、 n に関する総和をとった $\sum_{n=0}^{\infty} a_n x^n$ を{ a_n }の母関数 $g(x)$ とよぶ。

とくにカタラン数のばあい、母関数を $C(x)$ とし、 C と略記する。

$$C(x) = c_0 x^0 + c_1 x^1 + c_2 x^2 + \dots + c_k x^k + \dots \quad \text{とおくと,}$$

$\{C(x)\}^2 = c_0^2 + (c_0 c_1 + c_1 c_0)x + \dots + (c_0 c_{k-1} + c_1 c_{k-2} + \dots + c_{k-2} c_1 + c_{k-1} c_0)x^{k-1} + \dots$ となる。つまり、畳み込みで定義された数列の母関数を自乗すると、 x^n の係数が畳み込みになる。

次に、 $c_0 = 1$ から順に c_k に x^k を掛けて書き出してゆくと、下のようになる。

$$\begin{aligned} c_0 &= 1 \\ c_1 x &= c_0^2 x \\ c_2 x^2 &= (c_0 c_1 + c_1 c_0) x^2 \\ c_3 x^3 &= (c_0 c_2 + c_1 c_1 + c_2 c_0) x^3 \\ &\dots \\ c_k x^k &= (c_0 c_{k-1} + c_1 c_{k-2} + \dots + c_{k-2} c_1 + c_{k-1} c_0) x^k \\ &\dots \end{aligned}$$

左辺の和は定義から $C(x)$ であり、右辺の和は上の式から $x\{C(x)\}^2 + 1$ になる。

よって、

$$x\{C(x)\}^2 + 1 = C(x).$$

つまり、関数2次方程式 $x C^2 - C + 1 = 0$ をえる。

この2次方程式を解の公式で解くと、

$$C = \frac{1 \pm \sqrt{1-4x}}{2x}$$

をえるが、複号のプラスのほうは $C(x)$ の初項が定数項にならないので捨て、マイナスのほうをテイラー展開すると、下のようになり、めでたくカタラン数が x^n の係数にあらわれる。

$$C(x) = \frac{1 - \sqrt{1-4x}}{2x} = 1 + x + 2x^2 + 5x^3 + 14x^4 + 42x^5 + 132x^6 + \dots$$

1-4. カタラン数の線形差分方程式

関数 2 次方程式 $x^2C^2 - C + 1 = 0$ を x で微分すると,

$$C^2 + 2xCC' - C' = 0.$$

これに x をかけ, 微分する前の $x^2C^2 = C - 1$ を代入すると次の式をえる.

$$2x^2CC' - xC' + C - 1 = 0.$$

これに x^2C をかけ, もう一度 $x^2C^2 = C - 1$ を代入すると次の式をえる.

$$x^2CC' - 2x^2C' + (1-x)C - 1 = 0.$$

この式と上の式を連立し, x^2CC' を消去すると,

$$(1-2x)C + x(1-4x)C' - 1 = 0.$$

ここで, $(1-2x)C = \sum_{n=0}^{\infty} c_n x^n - 2 \sum_{n=0}^{\infty} c_n x^{n+1}$ だから, x^n の係数は $c_n - 2c_{n-1}$.

また $x(1-4x)C' = \sum_{n=0}^{\infty} n c_n x^n - 4 \sum_{n=0}^{\infty} n c_n x^{n+1}$ だから, x^n の係数は $n c_n - 4(n-1)c_{n-1}$.

よって $c_n - 2c_{n-1} + n c_n - 4(n-1)c_{n-1} = 0$ より, 次の線形の漸化式(差分方程式)をえる.

$$(n+1)c_n = 2(2n-1)c_{n-1}.$$

ここから, 逐次代入法により, 数列 $\{c_n\}$ の一般項,

$$c_n = \frac{(2n)!}{n!(n+1)!} = \frac{1}{n+1} \binom{2n}{n}$$

が求まる.

1-5. カタラン数の性質

線形の漸化式(差分方程式)からは, 項比の極限が容易に計算できる. すなわち,

$$(n+2)c_{n+1} = (4n+2)c_n \quad \text{であるので,}$$

$$\lim_{n \rightarrow \infty} \frac{c_n}{c_{n+1}} = \lim_{n \rightarrow \infty} \frac{n+2}{4n+2} = \frac{1}{4}.$$

ダランベールの判定法により, この $\frac{1}{4}$ は無限級数 $C(x) = \sum_{n=0}^{\infty} c_n x^n$ の収束半径になる.

また, $C(x)$ に $x = \frac{1}{4}$ を代入すると, 2 に収束することが証明できる.

$$C\left(\frac{1}{4}\right) = 1 + \frac{1}{4} + 2 \cdot \left(\frac{1}{4}\right)^2 + 5 \cdot \left(\frac{1}{4}\right)^3 + 14 \cdot \left(\frac{1}{4}\right)^4 + 42 \cdot \left(\frac{1}{4}\right)^5 + \dots = 2.$$

以上, カタラン数は多くを「語る」. なお, カタラン数の母関数に関する計算は, 形式的な議論にとどまり, 厳密な証明ではないことをお断りしておく.

2. 返り点は < > だけであらわせる

2-1. 豎点の個数と漢文数列

漢字 3 字のばあい, 返り点に加えて, 豎点(たててん. ハイフン. 漢字連結記号)を使うと, $A_2 - B C_1 = C A B$ が生み出され, 全部で 6 通りになることは, 先に述べた.

漢字 4 字では, 返り点だけの 14 通りに加えて, 豎点 1 個を使う $A_2 - B C_1 D = C A B D$, $A_2 - B C D_1 = C D A B$, $A_3 - B C_2 D_1 (= A_2 - B C_1 \vee D) = D C A B$, $A_3 B_2 - C$

$D_1 (=A \downarrow B_2 - C D_1) = D B C A$, $A B_2 - C D_1 = A D B C$, 豎点 2 個を使う $A_2 - B - C D_1 = D A B C$ の 6 通りが生じ, 全部で 20 通りが生み出される. $B D A C$, $C A D B$, $D A C B$, $D B A C$ の 4 個は, 返り点・豎点では生み出すことができない. なお, カッコ内は異点同順の例で, ここでは同じパターンとして数えている.

一般に n 字の漢字列に対して, 豎点の使用は $n - 2$ 個まで想定できる.

このようにして, n 字の漢字列に返り点および豎点 1 個を付け加えて生み出されるパターン数の数列を $\{k^1(n)\}$, 返り点および豎点 2 個までで生み出される数列を $\{k^2(n)\}$, 返り点および豎点 3 個までで生み出される数列を $\{k^3(n)\}$, ..., さらに豎点の使用を無制限つまり $n - 2$ 個までとした数列を $\{k_n\} (= \{k^{n-2}(n)\})$ とすると, つぎのようになる. ここでは, 数列 $\{a_n\}$ を $\{a(n)\}$ と書き, カタラン数 $\{c_n\}$ は返り点だけで豎点は 0 個なので $\{k^0(n)\}$ と書いている.

n	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...
$\{k^0(n)\}$	1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, ...
$\{k^1(n)\}$	1, 1, 2, 6, 19, 64, 225, 816, 3031, 11473, 44096, 171631, ...
$\{k^2(n)\}$	1, 1, 2, 6, 20, 69, 248, 919, 3485, 13461, 52778, 209513, ...
$\{k^3(n)\}$	1, 1, 2, 6, 20, 70, 253, 942, 3589, 13925, 54834, 218598, ...
...	
$\{k_n\}$	1, 1, 2, 6, 20, 70, 254, 948, 3618, 14058, 55432, 221262, ...

以上のような数列たち, $\{k^m(n)\}$ ($m = 0, 1, 2, \dots$) を漢文数列とよぶ.

2-2. カッコ表示の作り方 (返り点だけの場合)

漢文数列の各項が求められる理由は, むろん返り点・豎点のパターンを全部書き出していったからではない. このような数え上げができるのは, 漢文訓読で使用する返り点・豎点を「カッコ表示」によって数学モデル化しているからである.

一二点, 上下点, 甲乙点, 東西点, ... など, 漢字の読む順序を示す返り点を総称して順序点とよぶ.

はじめに同種の順序点を ... 3 2 1 の算用数字に書きかえる. たとえば, 師不₃必賢₂於弟子₁ (師必ずしも弟子より賢ならず) は師不₃必賢₂於弟子₁ となる. ここで 2 点以上の点を左カッコ < に置きかえ, 対応する右カッコ > を 1 点の位置に置くと, 師不<必賢<於弟子>> となる.

一二点が上下点のあいだにあるばあいは, はじめに一二点を算用数字に書きかえて, 左右のカッコ < > に置きかえたあと, 上下点を算用数字に書きかえ, さらに左右のカッコ < > に置きかえる. たとえば, 不_下以₂千里₁称_上也 (千里を以て称せざる也) は, 不_下以₂千里₁称_上也 \Rightarrow 不_下以<千里>称_上也 \Rightarrow 不₂以<千里>称₁也 \Rightarrow 不<以<千里>称>也 となる.

一二点, 上下点に加えて, 甲乙点などがあるときも, 下位の順位点を先に算用数字への書きかえ, 左右のカッコへの置きかえ, という手順を繰り返せばよい.

レ点のばあい, 有_レ朋 (朋有り) は有₂朋₁ から有<朋> と置きかえる.

レ点が続く不_レ可_レ転也 (転ばすべからず=也は不読) は、不₃可₂転₁也と書きかえ、不_レ可_レ転>>也と置きかえる。

レ点が一二点のあいだにあるばあいは、レ点を下位の順位点とみなし、うえの一二点が上下点のあいだにあるときと同様に、はじめにレ点を算用数字に書きかえ、左右のカッコに置きかえたあと、一二点を算用数字、ついで左右のカッコに置きかえる。

レ点と一二点が複合した百聞不_レ如_レ一見_レ (百聞は一見に如 (し) かず) は百聞不₃如₂一見₁から百聞不_レ如_レ一見>>と置きかえる。同様に、A_レB C_レDはA₃B C₂D₁と書きかえ、A<B C<D>>と置きかえる。A_レB C_レD_レE F_レは最後のF_レをF₁にして順にA₄B C₃D₂E F₁とし、A<B C<D<E F>>>と置きかえる。

なお、使_下学生_レ笑_上 (学生をして笑わしむ) のような使役形特有の返り点のばあい、使_下学生_レ笑_上と異点同順になるので使₂学生_レ笑₁から使_レ学生_レ笑>とする。

以上が、返り点 (順序点とレ点) だけを使うばあいの「カッコ表示」の作り方。多種多様な返り点は、このように< >だけであらわすことができる。

2-3. カッコ表示の総数

n字の漢字列A₁A₂…A_nが与えられた時、返り点だけを使う場合のカッコ表示の総数 (返り点の付け方のパターンの総数、と言いかえてもよい) はいくつだろうか。

i字の漢字列A_kA_{k+1}…A_{k+i-1}のカッコ表示をr(A_kA_{k+1}…A_{k+i-1})、その総数を|r(A_kA_{k+1}…A_{k+i-1})|と書き、c_iとする。|r(A₁A₂…A_n)|=c_nの漸化式を求めてみよう。

2-2.で出来上がったカッコ表示をよく見ると、先頭の漢字A₁のあとに返り点がない・あるの違いによって、(1)A₁A₂から始まる、(2)A₁<A₂から始まる、の二通りしかないことがわかる。

(1)A₁A₂から始まるケース

A₁以外のn-1字の漢字列のカッコ表示の数は、|r(A₂A₃…A_n)|=c_{n-1}。

(2)A₁<A₂から始まるケース

A₁の右側にある< から対応する> までの間にA₂A₃…A_{i+1}のi字(1≤i≤n-1)があるとすると、残りはA_{i+2}…A_nのn-i-1字。よって

r(A₁A₂…A_n)=A₁<r(A₂A₃…A_{i+1})>r(A_{i+2}…A_n) から、場合の数は

$$|r(A_2A_3 \dots A_{i+1})| \times |r(A_{i+2} \dots A_n)| = c_i c_{n-i-1} \quad (1 \leq i \leq n-1)$$

c_nは(1)のc_{n-1} と(2)のc_ic_{n-i-1}のiに関する総和(1≤i≤n-1)だから、

$$c_n = c_{n-1} + \sum_{i=1}^{n-1} c_i c_{n-i-1} = c_0 c_{n-1} + \sum_{i=1}^{n-1} c_i c_{n-i-1} = \sum_{i=0}^{n-1} c_i c_{n-i-1} \quad (c_0 = 1)$$

と、最初に述べた、畳み込みで定義されるカタラン数の漸化式になる。

3. カタラン数を拡張する

3-1. 堅点を認めるときの漸化式

ここからは、カッコ表示の概念を、堅点 m 個($1 \leq m \leq n-2$)までの使用をみとめる場合や再読文字の使用をみとめる場合に拡張しよう。

はじめに堅点 1 個の使用をみとめる場合を説明する。

卑₂一下之₁ (之 (これ) を卑下する), 欲₃平₂一治天下₁ (天下を平治せんと欲 (ほつ) す) のように、堅点には必ず順序点が付属している。そこで堅点をハイフンとして残し、卑一下<之>, 欲<平一治<天下>>のようにカッコ表示を決めるのが妥当であろう。

n 字の漢字列に返り点および堅点 1 個の使用をみとめるカッコ表示の総数 $k^1(n)$ を算出するには、さき返り点だけのカタラン数の、(1) A_1A_2 から始まる、(2) $A_1 < A_2$ から始まる、に加えて、(3) A_1-A_2 から始まる、というケースを付け加えればよい。

(3)は、(2)と同様に、 A_2 の右側にある< から対応する> までの間に $A_3A_4 \cdots A_{i+2}$ の i 字 ($1 \leq i \leq n-2$)があるとすると、残りは $A_{i+3} \cdots A_n$ の $n-i-2$ 字。これより、

$$k^1(n) = \sum_{i=0}^{n-1} k^1(i)k^1(n-i-1) + \sum_{i=0}^{n-2} k^1(i)k^1(n-i-2) - k^1(n-2) \quad \text{をえる.}$$

説明は省略するが、一般に、堅点 m 個($1 \leq m \leq n-2$)までの使用をみとめるときは、

$$\begin{aligned} k^m(n) &= \sum_{i=0}^{n-1} k^m(i)k^m(n-i-1) + \sum_{i=0}^{n-2} k^m(i)k^m(n-i-2) - k^m(n-2) \\ &\quad + \sum_{i=0}^{n-3} k^m(i)k^m(n-i-3) - k^m(n-3) + \cdots \\ &\quad + \sum_{i=0}^{n-m-1} k^m(i)k^m(n-m-1) - k^m(n-m-1) \quad \text{が成立つ.} \end{aligned}$$

堅点の使用に制限がない数列 $\{k^{n-2}(n)\}$ すなわち $\{k_n\}$ の漸化式は、

$$k_n = k_{n-1} + \sum_{i=1}^{n-1} \sum_{j=1}^{n-i} k_j k_{n-i-j} \quad \text{から (途中の計算を大幅に省略して)}$$

$$k_n = k_{n-1} - k_{n-2} + \sum_{i=0}^{n-1} k_i k_{n-i-1} \quad \text{をえる.}$$

3-2. 堅点を認める時の母関数の関数方程式

1-3. で述べた母関数の関数方程式を求める手順と同じように、畳み込みで定義された数列の母関数を自乗すると、 x^k の係数が畳み込みになることを利用し、漢文数列 $\{k^m(n)\}$ ($m = 0, 1, 2, \dots$)の母関数 $K_m(x) = \sum_{n=0}^{\infty} K_m(n) x^n$ の関数方程式を求めることができる。カタラン数をふくめて結果だけを書き出しておくと、

$$\begin{aligned} \{k^0(n)\} & \quad xK_0^2 - K_0 + 1 = 0 \\ \{k^1(n)\} & \quad x(1+x)K_1^2 - (1+x^2)K_1 + 1 = 0 \end{aligned}$$

$$\begin{aligned} \{k^2(n)\} & x(1+x+x^2)K_2^2 - (1+x^2+x^3)K_2 + 1 = 0 \\ \{k^3(n)\} & x(1+x+x^2+x^3)K_3^2 - (1+x^2+x^3+x^4)K_3 + 1 = 0 \\ \dots & \\ \{k_n\} & xK^2 - (1-x+x^2)K + 1 - x = 0 \end{aligned}$$

となる. 2-1.の表で見たように, 堅点の数 m が大きくなれば, $k^m(n)$ の値は k_n と一致するので, 当然と言えば当然であるが, 漢文数列 $\{k^m(n)\}$ の母関数の関数方程式たちは $m \rightarrow \infty$ としたとき, $|x| < 1$ の区間で $\{k_n\}$ の母関数の関数方程式に収束する.

また, 1-4.と同様の手順で, これらの関数方程式を微分し, $x^2K_mK_m'$ を消去し, 係数比較をおこなうと, 次の線形の差分方程式をえる. $\{k^2(n)\}$, $\{k^3(n)\}$ は長くなるので省略する.

$$\begin{aligned} \{k^0(n)\} & (n+1)k^0(n) = 2(2n-1)k^0(n-1) \\ \{k^1(n)\} & (n+1)k^1(n) = 3(n-1)k^1(n-1) + 6(n-1)k^1(n-2) + \\ & 2(n-2)k^1(n-3) - (n-5)k^1(n-4) - (n-5)k^1(n-5) \\ \{k_n\} & (n+1)k_n = 3(2n-1)k_{n-1} - 7(n-2)k_{n-2} + (2n-7)k_{n-3} - (n-5)k_{n-4} \end{aligned}$$

3-3. 再読を認めるとき

未₂完成₁ (未だ完成せず), 陛下不_レ応_レ憂_レ嶠 (陛下応 (まさ) に嶠 (きょう) を憂うべからず) の「未」と「応」のように再読をみとめるときも, カッコ表示を定義し, 堅点の数に応じた漢文数列とそれらの母関数の関数方程式を求めることができる.

未, 応, 猶などの再読文字は, 「いまだ」「まさに」「なお」などと初めに読み, 返り点の指示にしたがって「~す」「~べし」「ごとし」などと再読するので, 未- \langle 完成 \rangle , 陛下不 \langle 応 \langle 憂 \langle 嶠 $\rangle\rangle\rangle$ と「 \langle \rangle 」を用いたカッコ表示を採用する.

なお, 再読文字かどうかの判定は, 未^{いまだ}のように漢字の左右に添え仮名 (読み仮名, 送り仮名) があることを基準にすれば良いだろう. 左右の添え仮名は, 左右の傍訓ともいう.

返り点および堅点 m 個 ($1 \leq m \leq n-2$) までと再読をみとめる場合の, 漢字数 n 字ごとのカッコ表示の総数の数列を $\{p^m(n)\}$ とする. 3-1.と同様, $\{p^m(n)\}$ の数え上げは, 漢字列 $A_1A_2 \cdots A_n$ の先頭の2字に注目し, (1) A_1A_2 から始まる, (2) $A_1 < A_2$ から始まる, (3) A_1-A_2 から始まる, に加えて, (4) $A_1-\langle A_2$ から始まる, というケースを付け加えればよい.

煩雑になるので, 返り点だけ・再読ありの $\{p^0(n)\}$, 返り点・堅点1個・再読ありの $\{p^1(n)\}$, 返り点・堅点無制限・再読ありの $\{p^{n-2}(n)\} = \{p_n\}$ の三つの数列の母関数 P_0, P_1, P の関数方程式を示しておこう.

$$\begin{aligned} \{p^0(n)\} & 2xP_0^2 - (1+x)P_0 + 1 = 0 \\ \{p^1(n)\} & x(2+x)P_1^2 - (1+x+x^2)P_1 + 1 = 0 \\ \{p_n\} & x(2-x)P^2 - P + 1 - x = 0 \end{aligned}$$

ここでも漢文数列 $\{p^m(n)\}$ の母関数の関数方程式たちは $m \rightarrow \infty$ としたとき, $|x| < 1$ の区間で $\{p_n\}$ の母関数の関数方程式に収束する.

線形の差分方程式は次のようになる.

$$\{p^0(n)\} \quad (n+1)p^0(n) = 3(2n-1)p^0(n-1) - (n-2)p^0(n-2)$$

$$\begin{aligned} \{p^1(n)\} \quad & 2(n+1)p^1(n) = (11n-7)p^1(n-1) + (8n-7)p^1(n-2) - \\ & 3(n-4)p^1(n-3) - (4n-17)p^1(n-4) - (n-5)p^1(n-5) \\ \{p_n\} \quad & 2(n+1)p_n = (17n-7)p_{n-1} - 4(8n-13)p_{n-2} + 4(5n-13)p_{n-3} - \\ & 2(2n-7)p_{n-4} \end{aligned}$$

3-4. 漢文数列は拡張カタラン数

堅点の数 $m = 0, 1, 2, \dots, n-2$ に応じた漢文数列 $\{k^m(n)\}$ および $\{p^m(n)\}$ の線形差分方程式を求めるポイントを復習すると、次のようになる。

- ① カッコ表示とは、返り点（レ点と順序点）、堅点、左右の傍訓にもとづき、もとの漢字列に $\langle \rangle$ や $-$ を付け加えたもの。
- ② カッコ表示のパターンは、先頭の2漢字が(1) A_1A_2 から始まる、(2) $A_1 < A_2$ から始まる、(3) A_1-A_2 から始まる、(4) $A_1-< A_2$ から始まる、に分類して数え上げる。
- ③ 数え上げた数列は、畳み込みをふくむ非線形の漸化式を満たす。
- ④ 母関数 $G(x)$ を自乗し、非線形の漸化式を利用して、 $G(x)$ の関数2次方程式をつくる。
- ⑤ 関数2次方程式を微分し、もとの関数2次方程式を代入して x^2GG' を消去する。
- ⑥ G および G' の x^n の係数比較をおこない、線形の差分方程式を導く。

①は説明の便宜上、例文にもとづいた「カッコ表示の作り方」を説明している。現実の漢文では、堅点を四つ以上つかっている例はまだ見つからない。 $A-<B-<C>>$ のように、再読文字が入れ子になっているケースもない。ここで議論したカッコ表示は、現実をベースにしているものの、理論的なモデルであることをお断りしておく。

②の分類のうち、(1)(2)は返り点だけを認めるパターン、(1)(2)(3)は堅点の使用を認めるパターン、(1)(2)(4)は返り点・再読を認めるパターン、(1)(2)(3)(4)は返り点・堅点・再読を認めるパターンになっている。

③④⑤⑥は、すべての漢文数列に共通した計算のプロセスになっている。つまり、漢文数列はカタラン数を拡張したものである。

畳み込みをもつ数列に関しては、一般に次の【畳み込み数列の線形化定理】が成り立つ。

整数列 $\{a_n\}$ を次の差分方程式で定義する。 P_1, P_2, \dots, P_{k+1} は定数とする。

$$a_n = \sum_{i=1}^k P_i a_{n-i} + P_{k+1} \sum_{i=0}^{n-1} a_i a_{n-1-i} \quad (a_0 = 1 \text{ とする}).$$

すると、 $\{a_n\}$ の母関数 L は、多項式 $f(x), g(x), h(x)$ を係数にもつ関数2次方程式、

$$f(x)L^2 + g(x)L + h(x) = 0$$

を満たす。

また、ある自然数 N が存在し、 $\{a_n\}$ に関して、 Q_{n-j} と R_{n-j} ($0 \leq j \leq N$) を定数とする N 階線形差分方程式が成り立つ。

$$\sum_{j=0}^N (Q_{n-j}n + R_{n-j})a_{n-j} = 0.$$

4. カッコ表示を定義する

4-1. カッコ表示の標準形

ここからは再読現象はあつかわず、返り点・堅点の使用だけを認めることにする。つまり、漢文数列のうち $\{k^m(n)\}$ ($m = 0, 1, 2, \dots, n-2$)だけをあつかう。

レ点や順序点を算用数字に書きかえ、 $\langle \ \rangle$ に置きかえて構成したカッコ表示の特徴は、次のように整理できる。

- A) n 字の漢字列に $\langle \ \rangle$ を加えたものをカッコ表示とよぶ。 ($k^m(0) = 1$ とするため) $n=0$ つまり空 (empty) の漢字列だけでもカッコ表示とよび、 $n=1$ つまり漢字 1 字だけで、 $\langle \ \rangle$ を付けていないものもカッコ表示とよぶ。
- B) カッコ表示とカッコ表示を並置 (接続) したのももカッコ表示とよぶ。
- C) カッコ表示は必ず漢字 1 字またはハイフンでつながれた複数の漢字から始まる。
- D) カッコ表示の $\langle \ \rangle$ の中には、空 (empty) の漢字列以外のカッコ表示が入っている。

B)の接続 (concatenation または catenation) は、数理言語学の用語。記号交じりアルファベットの 2 つの列を並べてくっつけることをいう。ここでは、漢字・漢字列・カッコ表示を並べる (つなぎあわせる) 操作のこと。

再読現象を除いた、【カッコ表示の定義】は次のようにおこなう。

n 字の漢字列 $\mathbf{A} = A_1A_2 \cdots A_n$ に対し、カッコ表示 $\mathbf{r}_n(\mathbf{A}) = \mathbf{r}(A_1A_2 \cdots A_n)$ を次のように再帰的に定義する。再帰的とは、カッコ表示 $\mathbf{r}_n(\mathbf{A})$ の定義のなかにカッコ表示があらわれていることをいう。

$$\mathbf{r}_n(\mathbf{A}) = A_1 \cdots A_i \langle \mathbf{r}_j(A_{i+1} \cdots A_{i+j}) \rangle \mathbf{r}_{n-i-j}(A_{i+j+1} \cdots A_n)$$

ただし、 $1 \leq i \leq n-1, 0 \leq j \leq n-1, i+j \leq n$ とする。また、

$i = 1$ のとき、

(1) $j = 0$ ならば、 $\langle \mathbf{r}_j(A_{i+1} \cdots A_{i+j}) \rangle$ を空 (empty) とし、

(2) $j \geq 1$ かつ $i+j = n$ ならば、 $\mathbf{r}_{n-i-j}(A_{i+j+1} \cdots A_n)$ を空 (empty) とする。

$i \geq 2$ のとき、

(3) $i+j = n$ ならば、 $\mathbf{r}_{n-i-j}(A_{i+j+1} \cdots A_n)$ を空 (empty) とする。

この定義によれば、

(1) $i = 1, j = 0$ のとき、 A_1A_2 型の $\mathbf{r}_n(\mathbf{A}) = A_1\mathbf{r}_{n-1}(A_2 \cdots A_n)$

(2) $i = 1, j \geq 1, i+j = n$ のとき、 $\mathbf{r}_n(\mathbf{A}) = A_1 \langle \mathbf{r}_{n-1}(A_2 \cdots A_n) \rangle$

(3) $i \geq 2, i+j = n$ のとき、 $\mathbf{r}_n(\mathbf{A}) = A_1 \cdots A_i \langle \mathbf{r}_{n-i-1}(A_{i+1} \cdots A_n) \rangle$

となる。 $\mathbf{r}_n(\mathbf{A}) = A_1 \cdots A_i \langle \mathbf{r}_j(A_{i+1} \cdots A_{i+j}) \rangle \mathbf{r}_{n-i-j}(A_{i+j+1} \cdots A_n)$ を「カッコ表示の標準形」とよぶ。(1)から $n = 1$ の漢字 1 字 (単漢字) もカッコ表示となる。単漢字を接続した、 $\langle \ \rangle$ をふくまない漢字列でもカッコ表示とよぶことになる。この定義は、漢字列に $\langle \ \rangle$ をはめこむルールを規定しているだけの、静的な (スタティックな) 定義になっている。

4-2. カッコ表示の展開

カッコ表示の標準形において、「カッコ内を直前の豎点でつながれた漢字列の前に置き、その後、豎点を取り除く」という操作を「カッコ表示を展開する」または「カッコをはずす」とよび、カッコをはずした結果は元のカッコ表示に等しい、とする。すなわち、

$$\begin{aligned} r_n(A) &= A_1 \cdots - A_i < r_j(A_{i+1} \cdots A_{i+j}) > r_{n-i-j}(A_{i+j+1} \cdots A_n) \\ &= r_j(A_{i+1} \cdots A_{i+j}) A_1 \cdots - A_i r_{n-i-j}(A_{i+j+1} \cdots A_n) \\ &= r_j(A_{i+1} \cdots A_{i+j}) A_1 \cdots A_i r_{n-i-j}(A_{i+j+1} \cdots A_n) \end{aligned}$$

とする。むろん各項は条件に応じて empty または empty ではないとする。

このカッコをはずす操作を返接 (reflection) とよぶ。返接は、漢文訓読の「返読」に対応しており、返接操作の結果は、「返読した結果」とよんでもよいだろう。

たとえば、我心匪_レ石、不_レ可_レ転也 (我が心は石に匪 (あ) らず、転ばすべからず) は、
我心匪<石>、不<可<転>>也
=我心石匪、不<転可>也
=我心石匪、転可不也

とカッコ表示を展開する (返読する)。

ここで、不<可<転>>也の、可<転>を先に展開して、不<転可>也としてから、転可不也とするばあいと、不< >を先に展開して、可<転>不也としてから転可不也とするばあいの、どちらも結果が同じになることに注目していただきたい。すなわち、内側からカッコをはずしても、外側からカッコをはずしても、返読した結果は等しい。言い換えれば、漢文訓読の返読 (返接) の操作では、結合法則が成り立つ。

一二三点をつかった師不₃必賢₂於弟子₁ (師、必ずしも弟子より賢ならず) でも、
師不<必賢<於弟子>> 師不<必賢<於弟子>>
=師不<必於弟子賢> =師必賢<於弟子>不
=師必於弟子賢不 =師必於弟子賢不

のように、カッコを内側からはずしても、外側からはずしても、結果は等しい。

4-3. 語順変換とカッコ表示

元の漢字列とカッコ表示が与えられたとき、返読した結果は、元の漢字列の並び替え、つまり語順変換になっている。以下、語順変換を下敷きにした、動的 (ダイナミック) なカッコ表示の定義を述べる。

たとえば $A < B < C >> D = CBAD$ の $ABCD \rightarrow CBAD$ という語順変換は、Aが3番目に、Bが2番目に、Cが1番目に、Dが4番目に移るので、3214 という順列で表現できる。

また、 $ABCD \rightarrow CBAD$ が与えられたとき、AはC、Bよりうしろ (右) に移るので、元のカッコ表示は $A < B$ で始まり、BはCよりうしろ (右) に移るので、 $A < B < C$ となり、CはDより前 (左) に移るので、 $A < B < C >> D$ と、元のカッコ表示が復元できる。つまり、 $ABCD \rightarrow CBAD$ という語順変換とカッコ表示は 1 対 1 に対応するので、イコール (等号) でむすんでもよ

い. 以上から, カッコ表示が順列で表現できることがわかる.

4-4. 順列行列による定義

順列を順列行列 (各行, 各列に 1 がひとつだけあり, その他の成分はすべて 0 の正方行列) であらわし, たとえば, $A < B < C >> D = CBAD$ を下のようにならわす.

$$(ABCD) \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (CBAD)$$

順列行列を使うと, 数学らしく (?) カッコ表示を定義することができる.

まず, p 次の順列行列 P と q 次の順列行列 Q に対して, 直和 $P \oplus Q$ と直差 $P \ominus Q$ を下のようにならわす. どちらも $p+q$ 次の順列行列になる. 空白部分は成分がすべて 0 とする.

$$P \oplus Q = \begin{pmatrix} P & \\ & Q \end{pmatrix}, \quad P \ominus Q = \begin{pmatrix} & P \\ Q & \end{pmatrix}$$

順列行列全体の集合 $\mathbb{P} = \cup_{i=1}^{\infty} \mathbb{P}_i$ の部分集合で, 次の条件(1)(2)(3)をみたす $\mathbb{K} = \cup_{i=1}^{\infty} \mathbb{K}_i$ の元を【漢文行列】とよぶ. 各添数は順列行列の次数をあらわす. E_i は i 次の単位行列.

(1) $E_1 \in \mathbb{K}_1$

(2) $P \in \mathbb{K}$ であれば, $E_1 \oplus P \in \mathbb{K}$

(3) $P, Q \in \mathbb{K}$ であれば, 任意の i に対し, $E_i \ominus P \in \mathbb{K}$ かつ $(E_i \ominus P) \oplus Q \in \mathbb{K}$

(1)は, 漢字 1 字の E_1 による変換は元の漢字 1 字のままで, E_1 は漢文行列の一つであることを示している. 「漢字 1 字はカッコ表示である」とも言い換えられる.

(2)は, P が漢文行列であれば,

$$\begin{pmatrix} 1 & \\ & P \end{pmatrix}$$

も漢文行列であることを保証している. これは, $A_1 A_2$ タイプに対応している.

(3)は, P, Q が漢文行列であれば,

$$\begin{pmatrix} & E_i \\ P & \end{pmatrix}, \begin{pmatrix} & E_i \\ P & Q \end{pmatrix}$$

のどちらも漢文行列であることを保証している. この(3)は, 右肩にある E_i が「返読のとき後から読まれる堅点でつながれた漢字列の変換」に対応しており, 1 次の単位行列のときは (堅点のない) 返り点だけの $A_1 < A_2$ タイプに, 2 次以上の単位行列のときは $A_1 - A_2$ タイプに対応している.

0 次の順列行列として, 何もせず, 行列の次数を増やさない「空行列 Φ 」の概念を導入し, 空行列 Φ も漢文行列であるとすれば, 条件(3)の 2 つの行列は, 1 つにまとめることができる. すなわち,

(3) $P, Q \in \mathbb{K}$ であれば, 任意の i に対し, $(E_i \ominus P) \oplus Q \in \mathbb{K}$

とできる. このように定義された漢文行列の個数が堅点の個数 m に応じて数列 $\{k^m(n)\}$ にな

ることは、2-3.のカタラン数 ($m = 0$)や 3-1.の数え上げと同様に証明できる.

4-5. 漢文行列は群をなさない

順列行列は、行列式の値が ± 1 であり、転置行列が逆行列になるので、 n 次の順列行列全体の集合 \mathbb{P}_n は行列の積に関して群をなす. その位数 (個数) は $n!$ である.

また、 $\mathbb{P}_{n \oplus m} = \{A \oplus B | A \in \mathbb{P}_n, B \in \mathbb{P}_m\}$ と定義すると、 $A \oplus B, C \oplus D \in \mathbb{P}_{n \oplus m}$ のとき、 $(A \oplus B)(C \oplus D) = AC \oplus BD$ であり、 $\mathbb{P}_{n \oplus m}$ は行列の積に関して群をなす. さらに $A \oplus B = (A \oplus E_m)(E_n \oplus B)$ かつ群 $\widetilde{\mathbb{P}}_n = \{A \oplus E_m | A \in \mathbb{P}_n\}$ 、群 $\widetilde{\mathbb{P}}_m = \{E_n \oplus B | B \in \mathbb{P}_m\}$ は各々群 \mathbb{P}_n 、群 \mathbb{P}_m と同型になり、 $\mathbb{P}_{n \oplus m} \cong \widetilde{\mathbb{P}}_n \widetilde{\mathbb{P}}_m$ が成り立つ.

残念なことに、漢文行列全体の集合 \mathbb{K} は直差に関して閉じていないので、群をなさない.

また、非常に残念なことに、 n 次の漢文行列の集合 $\mathbb{K}_n (n \geq 4)$ は、行列の積に関して群をなさない. 「 n 次の漢文行列をどこまで拡張すると n 次の順列行列全体の集合 \mathbb{P}_n の部分群となるか?」は今後の数学の問題としてのこっている.

4-6. ブロック対角化とランダウ関数

一般の順列行列に関しては、【順列行列のブロック対角化定理】が成り立つ. つまり、任意の $A \in \mathbb{P}_n$ に対し、適当な $P \in \mathbb{P}_n$ を選び、 $P^T A P = J_{r_1} \oplus J_{r_2} \oplus \dots \oplus J_{r_i}$ とできる.

ここで、各 J_{r_k} は r_k 次の巡回行列で、 $1 \leq r_1 \leq r_2 \leq \dots \leq r_i \leq n$ かつ $r_1 + r_2 + \dots + r_i = n$.

このとき、 A を生成元とする巡回群の位数は、 r_1, r_2, \dots, r_i の最小公倍数である.

この定理は、適当な P を選べば、どんな順列行列でも (むしろ漢文行列でも) 巡回行列 ($J_{r_k} = E_1 \ominus E_{r_k-1}$) すなわち「もっとも単純で原始的な漢文行列」に「還元」できることを示している. この定理が数理漢文学でどのような意味をもつかは、まだ明らかではない.

さて、この定理から「自然数 n を分割し、それらの最小公倍数が最大になるような分割はどのようなものか?」、また「 n ごとの最大の最小公倍数 $g(n)$ の一般項は?」という問題が派生する. $g(n)$ をランダウ関数とよび、Wikipedia の Landau's function の項には、十分に大きな n に対して、

$$\ln g(n) < \sqrt{\text{Li}^{-1}(n)} \quad \text{ただし、} \text{Li}(x) = \int_2^x \frac{dt}{\ln t} \quad (\text{補正対数積分関数})$$

が成り立つことが、リーマン予想と同値である、とサラリと (?) 書いてある. 真偽のほどは定かでない.

5. 漢文訓読の形式言語学

5-1. リング表示とは

ノーム・チョムスキーは、1956年に Three Models for the Description of Language(言語をあらわす3つのモデル)、1959年に On Certain Formal Properties of Grammars(文法に関するある種の形式的性質について)と題した、形式文法を幾つかの型に分類する論文を発表した. 現代の情報科学ではこの分類をチョムスキー階層とよんでいる.

返り点・豎点の使用を認めるの漢文訓読のパターンを「リング表示」すると、チョムスキー階層にあてはめ、オートマトンを構成することができる。

リング表示とは、たとえば、百聞不_レ如_二一見_一-(百聞は一見に如かず)は、

(1)百聞と一見は漢字2字を続けて読み、

(2)如は一見をさきに読み、不は如_二一見_一をさきに読む、

という二つのルールにもとづいて読みください。(1)は漢字どうしを「順読」し、(2)は漢字1字をあとから「返読」する。それぞれ接続、返接とよべることは、これまでに述べた。

ここで漢字1字を丸カッコの対(ついで)として $()$ であらわし、順読する漢字2字は 00 とならば、返読する場合は、さきに読む漢字列を $()$ で包むことにすると、百聞不_レ如_二一見_一は、 $00((00))$ とあらわせる。このように返り点つき漢文を丸カッコであらわすことを、丸カッコの対の上部と下部をつなぐとリング(輪)になるので、漢文訓読の**リング表示**とよぶ(右図)。

【百聞不_レ如_二一見_一のリング表示】

$00((00))$

$00((00))$

$00 \bigcirc \bigcirc$

カタラン数になる例として、 n 対の $()$ を重ねないように配置する場合の数、という例があるとおり、返り点だけの使用を認め、豎点の使用を認めないばあい、 n 対のリング表示の総数は数列 $\{c_n\}$ になる。

すべてのリング表示の集合 $\{0,00,(0),000,(0)0,\dots\}$ を作り出す仕組みを、**形式言語学**の「書き換え規則」を用いて述べると、

規則1 $S \rightarrow ab$

規則2 $S \rightarrow aSb$

規則3 $S \rightarrow SS$

の3つになる。

S は**開始記号**とよび、この仕組みの出発点を意味するとともに、変数の役割をしている。

たとえば、規則1を使うと、 ab が導かれる。規則2と規則1を使うと $aabb$ が導かれる。規則3と規則1を使うと $abab$ が。規則1,2,3を組み合わせてを使うと、 $aabbab$ が導かれる。最後の例は、 $S \Rightarrow SS \Rightarrow aSbab \Rightarrow aabbab$ と導出される。

このようにして生まれる文字列(語(word)とよぶ)の a を左丸カッコに置き換え、 b を右丸カッコに置き換えるとリング表示になる。

規則1,2,3を繰り返し何度も適用すれば、すべてのリング表示を導くことができる。

形式言語学では、集合 $\{a, b\}$ をアルファベット(終端記号ともいう)、うへの3つのような書き換え規則をまとめて**文法**とよび、文法によって生成されるすべての語(word)の集合を**言語**とよぶ。

ここでは、規則1,2,3を文法 G_0 とし、 G_0 によって生成される言語を $L(G_0)$ と書く。

5-2. 返り点だけのオートマトン

すべての規則の左辺が開始記号 S だけになっているとき、その文法を**文脈自由文法**とよび、

言語を文脈自由言語とよぶ。文脈自由言語はプッシュダウンオートマトンに受理されることが知られている。

ここでのオートマトンとは、読み取った語(word)を内部で処理し、受理するか拒否するか、判定できる「仮想機械」をいう。たとえば、 $aabb, abaaabbb$ など $L(G_0)$ に属する「正しくカッコの付けられた語(word)」を受理し、 $aab, baab$ など $L(G_0)$ に属さない語を拒否するオートマトンは、「 $L(G_0)$ を受理する」という。

5-3. 文法の列を規定する書き換え規則

返り点のほかに豎点 1 個の使用を認める場合は、文法 G_0 の規則 1,2,3 に規則 5 を加えた文法 G_1 で生成される。アルファベットは $\{a, b, t\}$ の 3 種類。言うまでもなく t をハイフンに置き換えれば、リング表示になる。なお、規則 4 は再読をみとめるときに追加する。

規則 1 $S \rightarrow ab$

規則 2 $S \rightarrow aSb$

規則 3 $S \rightarrow SS$

規則 5 $S \rightarrow ataSbtb$

規則 5 は、「 S を(- (と)-)で囲んでもよい」ことを示している。

この文法 G_1 に、

規則 12 $S \rightarrow atataSbtb$

を付け加え、豎点 2 個までの使用を認める規則 1,2,3,5,12 を文法 G_2 とよぶ。

豎点 3 個の使用を認める場合は、さらに次の規則 13 を付け加え、文法 G_3 とする。

規則 13 $S \rightarrow atatataSbtb$

このようにして文法の列 G_0, G_1, G_2, \dots 、および言語の列 $L(G_0), L(G_1), L(G_2), \dots$ が定義できる。

これらの $L(G_m)$ ($m = 0, 1, 2, \dots$)もすべて文脈自由言語になっている。

いっぽう、「豎点が 1 つあれば、もう 1 つ追加してもよい」ことを規定する、下の規則 6 を追加すれば、無制限に豎点を使用することができる。

規則 6 $taSbt \rightarrow tataSbtb$

規則 6 は、「 ta と bt のあいだにある S は、 $taSbt$ に書き換えてもよい」ことを示しており、たとえば、漢字 1 字を豎点 2 個でつながれた漢字 3 字の熟語で返読する奴₂—僕—視₁は、規則 5,6 をつかって、 $S \Rightarrow ataSbtb \Rightarrow atataSbtb \Rightarrow atataabbtbtb$ と導出できる。

豎点 3 個以上のときも同様に規則 5,6 で導出できる。

規則 1,2,3,5,6 の、豎点の無制限の使用を認める文法を G_K とし、言語を $L(G_K)$ とする。

言語 $L(G_K)$ は、左辺が開始記号 S だけになっておらず、文脈依存言語に属している。

豎点の使用を m 個まで認める文法 G_m が生成する言語 $L(G_m)$ は、 m をどんどん大きくすると、「豎点の使用に制限のない、すべてのリング表示」にどんどん近づく。つまり $L(G_m)$ はどんどん $L(G_K)$ に近づき、文法 G_∞ と文法 G_K は等価と言える。

なお、チョムスキー階層の 4 つの言語はそれぞれ上位の言語の真部分集合であり、文脈自由言語は文脈依存言語の真部分集合だが、 G_K, P_k が文脈自由文法でなく、 $L(G_K), L(P_K)$ が

文脈自由言語でない証明はまだおこなっていない。

6. 漢文訓読の複雑度

6-1. 返読する漢字の延べ字数

たとえば、中国語の「有朋自遠方来」に対して、

①有_レ朋自_二遠方_一来（朋有り，遠方より来る） 有<朋>自<遠方>来=朋有遠方自来

②有_下朋自_二遠方_一来_上（朋の遠方より来る有り） 有<朋自<遠方>来>=朋遠方自来有
の二通りの読み方があるが、①は有より先に「朋」、自より先に「遠方」、と合計3字を返読する。②は「朋自<遠方>来」から<と>をのぞいた漢字5字を有より先に読み、「遠方」の2字を自より先に読むので、延べ7字を返読する。

返読する延べ漢字字数は、訓読の仕方の複雑度の指標のひとつと言ってもよいだろう。

元の有朋自遠方来の6つの漢字に順に123456の番号を付け、カッコ表示を展開すると、①は214536に語順が並び替わる。214536を有<朋>自<遠方>来=朋有遠方自来のK順列とよぶ。

$$\begin{bmatrix} \text{有} & \text{朋} & \text{自} & \text{遠} & \text{方} & \text{来} \\ \text{朋} & \text{有} & \text{遠} & \text{方} & \text{自} & \text{来} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 4 & 5 & 3 & 6 \end{bmatrix}$$

ここで、上段と下段の同じ漢字、同じ数字どうしを線分で結ぶと、次のようになる。

$$\begin{bmatrix} \text{有} & \text{朋} & \text{自} & \text{遠} & \text{方} & \text{来} \\ \text{朋} & \text{有} & \text{遠} & \text{方} & \text{自} & \text{来} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 4 & 5 & 3 & 6 \end{bmatrix}$$

2つの線分で生じる交点数は、返読する延べ漢字数と一致している。同様に②も、

$$\begin{bmatrix} \text{有} & \text{朋} & \text{自} & \text{遠} & \text{方} & \text{来} \\ \text{朋} & \text{遠} & \text{方} & \text{自} & \text{来} & \text{有} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 4 & 5 & 3 & 6 & 1 \end{bmatrix}$$

となり、やはり線分の交点数が返読する延べ漢字数と一致する。

6-2. 豎点を使う場合

豎点を用いた漢文の場合、先に読む< >内の漢字数に、後に読む「豎点でつながれた漢字数」を掛けたものが交点数になる。たとえば、原文の「欲平治天下」に返り点と豎点をつけ、欲₃平₂—治天下₁（天下を平治せんと欲す）と訓読するとき、カッコ表示は、欲<平—治<天下>>=天下平治欲、K順列は45231、交点数は8となる。

$$\begin{bmatrix} \text{欲} & \text{平} & \text{治} & \text{天} & \text{下} \\ \text{天} & \text{下} & \text{平} & \text{治} & \text{欲} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 2 & 3 & 1 \end{bmatrix}$$

ここでは、先に読む天—天，下—下の2つの線分を、後に読む豎点でつながれた平—平および治—治の平行線が横切って交点4つを生じている。つまり、先に読む天と下の漢字

数 2 に、堅点でつながれた平と治の漢字数 2 を掛けた $2 \times 2 = 4$ が交点の数になる。さらに欲一欲がその他の漢字 4 つの線分をすべて横切って交点 4 つを生じ、つごう交点数 $4 + 4 = 8$ となっている。

これは、堅点でつながれた漢字それぞれが、先に読む $\langle \quad \rangle$ 内の漢字を返読していると解釈できる。卑₂一_下之₁（これを卑下す）、カッコ表示、卑一_下 \langle 之 \rangle は、卑と下の 2 漢字がそれぞれ之（これ）1 漢字を返読している、と考えるわけである。

6-3. 重要な注意事項

(1) 原文の漢字が重複している場合

原文のなかに同じ漢字があるばあい、読み下し文のなかの対応する漢字どうしを線分でつなぎ、交点数を求める。下の例は、原文が「龍池躍龍龍已飛」、読み下しが「龍池（りゅうち）龍を躍（おど）らせて龍已（すで）に飛べり」。 (交点を生じない線分を省略)

$$\left[\begin{array}{cccccc} \text{龍} & \text{池} & \text{躍} & \text{龍} & \text{龍} & \text{已} & \text{飛} \\ & & \times & & & & \\ \text{龍} & \text{池} & \text{龍} & \text{躍} & \text{龍} & \text{已} & \text{飛} \end{array} \right] = \left[\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ & & & \times & & & \\ 1 & 2 & 4 & 3 & 5 & 6 & 7 \end{array} \right] \text{ 交点数 } 1$$

どの「龍」と、どの「龍」とを線分でつなぐかの判断は、主観的にならざるをえない。

(2) 3 本以上の線分は 1 点で交わらない

3 つの線分が 1 点で交わる時は、線分のひとつをずらして、3 点で交わるを考える。原文「難可到」、読み下し「到るべきこと難（かた）し」の交点数は下のようになる。

$$\left[\begin{array}{ccc} \text{難} & \text{可} & \text{到} \\ \diagdown & \diagup & \diagdown \\ \text{到} & \text{可} & \text{難} \end{array} \right] = \left[\begin{array}{ccc} 1 & 2 & 3 \\ \diagdown & \diagup & \diagdown \\ 3 & 2 & 1 \end{array} \right]$$

この図では、到一到、可一可の 2 線分の交点が 1 個、残りの難-難が、到-到、可-可の 2 線分と交点 2 個を生み出すので、つごう交点数は $1 + 2 = 3$ になる。

(3) 交点数の最大値

3 線分より多いときも同様に 1 点で交わらないように、線分をずらして交点数を求める。

4 字の K 順列 4321 のばあい、1-1、2-2、3-3 の 3 線分で生じる交点 3 個に加え、4-4 がこれら 3 線分とすべて交点をもつので、交点数は $1 + 2 + 3 = 6$ 。

以下同様に n 字の K 順列 $n(n-1) \dots 21$ の交点数は、

$$1 + 2 + \dots + (n-1) = \frac{n(n-1)}{2}$$

となる。この K 順列 $n(n-1) \dots 21$ を生み出すのは、 $A \prec B \prec \dots \prec N$ のような、先頭の A 以外のすべての漢字の左下（縦書きでは左肩）にレ点がついている場合で、このとき訓読文は、 $N \dots B A$ となり、原文の逆の順番になっている。

交点数の最小値は、すべての字数において、上段と下段の漢字列が等しく、交点を生じないとき、すなわち K 順列が $12 \dots n$ のとき最小値 0 をとる。

6-4. 数学的な補足

(1)逆ポーランド記法

2つの順列 α, β をこの順に使って、元の漢字列を並べ替えるとき、 α, β の積 $\alpha * \beta$ を元の漢字列から最後に並べ替わった漢字列に変換する順列として定義する。このように右から次の順列を掛けるやり方は、逆ポーランド記法とよばれている。

たとえば、 $\alpha = 3241, \beta = 4321$ のとき、

$$BCAD \xrightarrow{\alpha} DCBA \xrightarrow{\beta} ABCD, \text{つまり } BCAD \xrightarrow{3241} DCBA \xrightarrow{4321} ABCD$$

であるが、

$$BCAD \xrightarrow{\alpha * \beta} ABCD, \text{つまり } BCAD \xrightarrow{3241 * 4321} ABCD$$

だから、 $\alpha * \beta = 3241 * 4321 = 2314$ となる。

(2)逆元 = K 順列

カッコ表示、たとえば $ABCD \rightarrow A < BC > D = BCAD$ の順列 $\sigma = 3124$ に対する逆元 σ^{-1} は、 $BCAD \rightarrow ABCD$ と変換する順列 $\rho = 2314$ になる。この逆元は $1 < 23 > 4 = 2314$ として、順列 $\varepsilon = 1234$ に、 $A < BC > D$ のように $< >$ を付け加え、それを展開すると求めることができる。

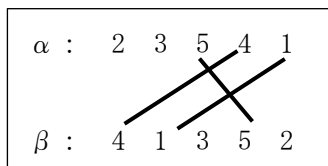
これまでK順列とよんだものは、この順列 $\sigma = 3124$ の逆元の順列 $\rho = 2314$ のことである。

この場合、 $\alpha = (\alpha^{-1})^{-1}$ が成り立つので、K順列 $\rho = 2314$ の逆元が $ABCD \rightarrow A < BC > D = BCAD$ の順列 $\sigma = 3124$ になる。つまり、 $\sigma^{-1} = \rho$ であり、 $\rho^{-1} = \sigma$ である。

(3)2つの順列の交点数

次に2つの順列の「順列間距離」(※末尾の追記を参照)の一種である交点数を定義しよう。

順列 $\alpha = e_1 e_2 \dots e_n$ の要素 $e_i (i = 1, 2, \dots, n)$ の位置 i を $p(e_i, \alpha)$ とあらわす。たとえば、 $\alpha = 23541$ のとき、 $p(2, \alpha) = 1, p(3, \alpha) = 2, p(5, \alpha) = 3, p(4, \alpha) = 4, p(1, \alpha) = 5$ である。 α において要素5の右側にある要素4, 1の集合 $\{4, 1\}$ は、 $\{x | p(5, \alpha) < p(x, \alpha)\}$ となる。



同様に順列 $\beta = 41352$ の要素5の左側にある要素4, 1, 3の

集合は $\{x | p(5, \beta) > p(x, \beta)\}$.

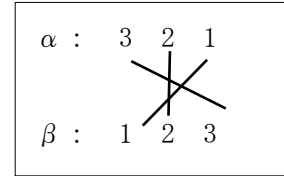
α において要素5の右側にあり、同時に β において要素5の左側にある要素の集合は、 $\{x | p(5, \alpha) < p(x, \alpha), p(5, \beta) > p(x, \beta)\}$ となる。これを $k(5, \alpha, \beta)$ と書く。この集合の大きさ(要素の個数)を $|k(5, \alpha, \beta)|$ と書く。 $\alpha = 23541, \beta = 41352$ のとき、要素1, 4の2個が α の要素5の右側にあつて同時に β の要素5の左側にあるので、 $|k(5, \alpha, \beta)| = 2$ である。

このようにして得られる、すべての要素に関する $|k(i, \alpha, \beta)|$ の総和を $d(\alpha, \beta)$ と書き、「順列 α と順列 β の交点数」とよぶ。

$$d(\alpha, \beta) = \sum_{i=1}^n |k(i, \alpha, \beta)|$$

$\alpha = 23541$ と $\beta = 41352$ では, $|k(1, \alpha, \beta)| = 0, |k(2, \alpha, \beta)| = 4, |k(3, \alpha, \beta)| = 2, |k(4, \alpha, \beta)| = 0, |k(5, \alpha, \beta)| = 2$ であるので, $d(\alpha, \beta) = 8$ となる.

なお, この定義では, $\alpha = 123, \beta = 321$ のように同じ要素どうしを結んだ3つの線分が1点で交わるような場合, 交点数は3になる. 4つの線分が1点で交わる場合, 交点数が6になることは, すでに述べた.



また, α と β を入れ替えても交点数は変わらないことは明らかで, $d(\alpha, \beta) = d(\beta, \alpha)$ が成り立つ. α の要素 i の「左側」にあって同時に β の要素 i の「右側」にある要素の個数を $|k(i, \alpha, \beta)|$ としても交点数は変わらない.

(4) K 順列の交点数

ここでは, K 順列 ρ と n 次 (字) の単位元 $\varepsilon = 12 \cdots n$ との交点数 $d(\rho, \varepsilon)$ を「K 順列 ρ の交点数」とよんでいる.

2つの順列の交点数の定義から, $d(\rho, \varepsilon) = d(\varepsilon, \rho) = d(\rho^{-1}, \varepsilon) = d(\varepsilon, \rho^{-1})$ が成り立つ.

$d(\varepsilon, \rho) = d(\rho^{-1}, \varepsilon)$ の証明は, $\rho = 3124$ のとき, $\rho^{-1} = 2314$ だから,

$$\left[\begin{array}{ccc|c} 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 4 \end{array} \right] \text{ において}$$

1 を 2 に, 2 を 3 に, 3 を 1 に, 4 を 4 に書きかえると,

$$\left[\begin{array}{ccc|c} 2 & 3 & 1 & 4 \\ 1 & 2 & 3 & 4 \end{array} \right] \text{ となることからわかる.}$$

(5) カッコ表示の標準形と交点数

n 字の漢字列 $A_1 A_2 \cdots A_n$ のカッコ表示を $r(A_1 A_2 \cdots A_n)$, その展開を $A_{e_1} A_{e_2} \cdots A_{e_n}$ と書き, その交点数を $\|r(A_1 A_2 \cdots A_n)\| = \|A_{e_1} A_{e_2} \cdots A_{e_n}\|$ と書く. $n=1$ の時, すなわち単漢字 A_1 のカッコ表示は A_1 であり, $\|A_1\| = 0$ と決める.

2つのカッコ表示を並置 (連接) したとき, 交点数は2つのカッコ表示の交点数の和となることは明らか.

$$\|r(A_1 A_2 \cdots A_m) r(A_{m+1} A_{m+2} \cdots A_n)\| = \|r(A_1 A_2 \cdots A_m)\| + \|r(A_{m+1} A_{m+2} \cdots A_n)\|$$

ここで, 返読現象が起こる返接 (reflection) の一般形とその展開,

$$A_1 - A_2 - \cdots A_i < r(A_{i+1} A_{i+2} \cdots A_n) >$$

$$= r(A_{i+1} A_{i+2} \cdots A_n) A_1 A_2 \cdots A_i$$

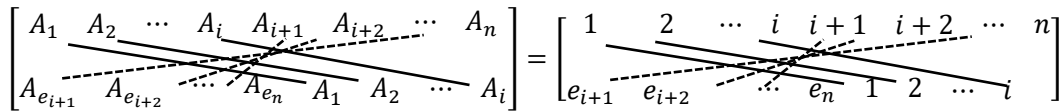
$$= A_{e_{i+1}} A_{e_{i+2}} \cdots A_{e_n} A_1 A_2 \cdots A_i \quad \text{ただし } 1 \leq i \leq n-1$$

を考える. 上段に $A_1 A_2 \cdots A_n$, 下段に $A_{e_{i+1}} A_{e_{i+2}} \cdots A_{e_n} A_1 A_2 \cdots A_i$ を置き, 同じ漢字どうしを線分でつなぐ (3本以上の線分は1点で交わらないようにする) と, $A_1 A_1$ の線分は $A_{i+1} A_{i+1}, A_{i+2} A_{i+2}, \dots, A_n A_n$ の $(n-1)$ 本の線分と交点を生じる.

同様に、 A_2A_2, \dots, A_iA_i の線分も $A_{i+1}A_{i+1}, A_{i+2}A_{i+2}, \dots, A_nA_n$ の $(n-1)$ 本の線分と交点を生じる。また、 $A_1A_1, A_2A_2, \dots, A_iA_i$ の i 本の線分は互いに交わらない（次頁の図）。よって、

$$\|A_1 - A_2 - \dots - A_i < r(A_{i+1}A_{i+2} \dots A_n) >\| = i(n-1) + \|A_{e_{i+1}}A_{e_{i+2}} \dots A_{e_n}\|$$

を得る。この式は、一般に、ハイフンで連結された i 字 ($1 \leq i \leq n-1$) が返読されるときの交点数が「返読する漢字の延べ字数」に等しいことをあらわしている。



6-5. K順列と交点数

K順列を利用する最大のメリットは、原文の中国語と日本語の読み下し文があれば、簡単にK順列が求められるという点にある。近年のインターネットはもちろん、漢文に関する出版物には、レ点、一二点、豎点などの記号を表記せず、原文と読み下し文だけを示しているものが多い。

例として、荻生徂徠の高弟・服部南郭の『唐詩選国字解』にある李白の五言絶句「静夜思」の各句のK順列と交点数を求めてみよう。（旧字、旧かなづかいは改めた）

牀前看月光	牀前 月の光を看る
疑是地上霜	疑うらくは是れ地上の霜かと
挙頭望山月	頭を挙げて山月を望み
低頭思故郷	頭を低れて故郷を思う

まず、読み下し文「牀前 月の光を看る」の漢字を順に見て、原文の第一句（起句）の「牀前看月光」の何番目にあるかを書き出してゆく。

牀は原文「牀前看月光」の1番目の漢字、前は原文の2番目、月は原文の4番目、光は原文の5番目、看は原文の3番目の漢字である。これらの数字を並べた12453が「牀前看月光」を「牀前月の光を看る」に並べ替えるK順列である。

第二句（承句）のK順列は、原文と読み下し文の漢字の語順は同じだから、12345。

第三句（転句）の「頭を挙げて山月を望み」の頭は原文の2番目の漢字、挙は原文の1番目、山は4番目、月は5番目、望は3番目の漢字だから、K順列は21453になる。

第四句（絶句）の「頭を低れて故郷を思う」は、第三句（転句）とまったく同じ構造をしており、K順列は第三句（転句）と同じ21453になる。

K順列が求めれば、上段に12345、下段にK順列を並べ、同じ数字を線分をつなぎ、実際に交点数を求めればよい。李白の「静夜思」の交点数は、起句2、承句0、転句と絶句3、とわかる。

全語数に対する交点数の合計の比、つまり全語数のうち返読している漢字の割合を返読率とよぶことにすると、李白の五言絶句「静夜思」の返読率は次のように計算できる。

$$(2 + 0 + 3 + 3) \div (5 \times 4) = 0.4$$

6-6. 計量漢文学の可能性

以上の論点を整理すると、次のようになる。

- ① 返読の複雑さを「返読する漢字の延べ字数」で定義する。

ただし、カッコ表示の< >が並置されているときや、< >が入れ子になっている場合は、各< >内の漢字数の和を返読する漢字の延べ字数とする。堅点がある場合は、堅点でつながれた漢字数と< >内の漢字数との積を返読する漢字の延べ字数とする。

- ② 返読する漢字の延べ字数は、K順列と交点数を利用して求めることができる。

K順列は、元の中国語の漢字列を日本語訓読文に変換する順列の逆元になっている。

- ③ K順列は、返り点・堅点つき漢文のカッコ表示の漢字列を $12 \dots n$ に置きかえ、カッコを展開して得られる数字列として求めることができる。

これまで、漢文訓読の計量的な研究はほとんどなされていなかったと思う。今後、このK順列と交点数を利用して、中国人や日本人が書いた漢文の計量的研究がおこなわれることを期待する。

7. まとめ

以上、数理漢文学という新分野の概要とおもな問題や応用例を紹介した。

漢文訓読の返り点・堅点システムは、古典中国語（文言文）を日本語（古文）へ翻訳する、いわばメタ言語であり、歴史的産物という意味から自然メタ言語としてとらえることができる。残された数学の問題解決はもちろん、今後、情報科学としての数理漢文学が発展することを期待している。

ここに紹介したカッコ表示や語順変換、オートマトン、交点数などの概念は、中国語から日本語への翻訳ばかりでなく、英語と日本語、中国語と英語などの翻訳にも応用することができると思う。むしろ、将来をにやう若者の情報教育にもいかせるだろう。

こころある若い人々が数学の素養と創造力をやしない、自分の力で更なる数理漢文学を切りひらくことを期待している。

本稿の執筆には、安藤洋美、太田博史、大西正男、小坂葉子、古田島洋介、小寺裕、斉藤明、坂田定久、田村三郎（2013年逝去）、田村直之、田村誠、西本実苗、張替俊夫、平井崇晴、前川太市（2019年逝去）、三浦伸夫、山内みや子、湯城吉信、Douglas G. Rogers、J. Marshall Ungerの各氏に協力いただいた。記して感謝申しあげる。

※追記

2つの順列どうしの距離 $d(\alpha, \beta)$ が距離の3条件を満たすことは、LR論法を用いて証明する。和算序林の計量国語学会予稿シリーズの2019年「漢文訓読の複雑度について—歴史情報学の試み—」を参照してほしい。（2024.2.24）