Ｊ研究会資料　2001/6/23

組合わせ数学（Combinatorics）へのＪのツール
－並べ替え（順列，置換）の関数Ａ．とＣ．について

西川 利男　(Toshio.Nishikawa@kiu.ne.jp)

コンピュータのディジタル志向に呼応して，組合わせ数学（Combinatorics）はいまや現代数学において確固たる位置を占めつつあるといえよう．その中ではいろいろな並べ替え（Permutation，順列，置換）操作が行われるが，要素の数が多くなると仲々大変である．

Ｊの関数Ａ．とＣ．とはそのための強力なツールとなり，いろいろな並べ替え操作を手軽に行うことができる．また，ＢＡＳＩＣ，Ｃ，ＦＯＲＴＲＡＮなどを見渡してもプリミティブ・レベルでこのような機能を備えている言語は見当たらない．ここにもＪのユニークな特徴がある．

１．関数Ａ．とＣ．

単項のＡ．　辞書順のインデクス

２項のＡ．　辞書順の順列の生成

単項のＣ．　直接置換と巡回置換の切り替え

２項のＣ．　直接および巡回置換の操作

２．並べ替え，順列，置換などの用語

ごく一般的なことば「並べ替え」に対して，数学の用語「順列」と「置換」とがやや混乱して使われているのではないか．筆者の感じでは「並べ替える操作」を「置換」，「並べ替えられた状態」を「順列」という意味であろうと思う．そして両方とも英語では「permutation」，もっとくだけて「Anagram」という語が使われる．

３．関数Ａ．の使い方の実際

３つの文字 a，b，c から相異なる３つを取り出して，abc... の辞書順に並べた順列を作ってみるとつぎのようになる．また，この順列の数は 3! = 6 通りになる．

```
abc
acb
bac
bca
cab
cba
```

上のすべての順列はつぎのようにして生成される．

```
   D =. 0 1 2 3 4 5 A. 'abc'
```

ここで例えば，順列の１つ bca は ０-オリジンの３番目としてとり出される．

```
   3 { D
bca
```

これをもとの abc から得るには，つぎのようにする．

```
   3 A. 'abc'
```

つまり，２項関数Ａ．は左引数のインデックスに応じた順列を返す．

これは考え方によっては「置換」の操作によって生成したものと考えることもできる．これを行うのが関数Ｃ．である．

```
   1 2 0 C. 'abc'
```

このような意味から，左引数は置換ベクトルとも呼ばれる．

ところで，bca が何番目かというインデックスを知るにはどうしたらよいだろう．そのための検索には，ｉ．の２項関数の左引数に'キー'として'abc'を指定し，つぎのようにしてまず置換ベクトルを得る．

```
   'abc' i. 'bca'
1 2 0
```

つぎに単項の関数Ａ．により

```
   A. 1 2 0
3
```

とインデックスが求められる．

また，置換ベクトルのすべて，すなわち置換配列Ｐはつぎのようにして得られる．

```
   P =. 0 1 2 3 4 5 A. 0 1 2
   P
0 1 2
0 2 1
1 0 2
1 2 0
2 0 1
2 1 0
```

お分かりのように置換配列は ０ １ ２ なる３つの数値の値順の順列に他ならない．いうまでもないが，並べ替えの要素は文字でも数値でもよい．しかし，わかり易さのために最初の例では文字を要素とした．数値の場合，順列の要素なのか，インデックスなのか，さらには置換ベクトルなのかを混同しないようによく注意することが必要である．

関数Ａ．の最も便利な利用として

```
   perm =: (i.@(!@#)) A. ]
```

なるイディオムがある．これは最初にあげたような全順列を列挙する関数であるが，要素数が大きくなったときにはとうてい手作業ではできないが，そのとき威力を発揮する．

４．関数Ｃ．の使い方の実際

つぎには５文字から成るデータ

```
   'abcde'
```

を用いて，関数Ｃ．による置換操作を説明しよう．

まず，関数 take { により取り出す．

```
   1 2 3 0 4 { 'abcde'
```

同じことは，上の左引数の値を置換ベクトルとして，関数Ｃ．により置換操作として

行うこともできる.

```
      1 2 3 0 4 C. 'abcde'
    bcdae
```

数値データについて同じようにするとつぎのようになる.

```
      1 2 3 0 4 C. 0 1 2 3 4
    1 2 3 0 4
```

これは数学の置換の表記法に従えばつぎのように書かれる.

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 0 & 4 \end{pmatrix}$$

これをことばで言えば

（０を１に），（１を２に），（２を３に），（３を０に），（４を４に）

置換（直接置換）する，という.

注）Ｊでは０－オリジンなので上のようになるが，通常，数学では１－オリジンであるのでつぎに対応する.

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

一般に置換操作はこのような直接置換だけではなく，巡回置換によって行うこともできる.

巡回置換（Cyclic Permutation）とは要素を次々と巡回的（Cyclic）に置き換えていく操作である．これにもＪの関数が使われるのである.

単項関数Ｃ．は直接置換ベクトルを巡回置換ベクトルに変換する.

```
   C. 1 2 3 0 4
```

```
+--------+-+
|3 0 1 2|4|
+--------+-+
```

このように巡回置換ベクトルはボックス化された値として示される．

この例で言えばこれは２つの環から成り，第１の環では

　（３は０へ，０は１へ，１は２へ，２は３へ）と

次々とサイクリックに置き換えられる．第２の環としては（４は４へ）つまり移動しない．

　これを数学での表記法ではつぎのように書かれる．

$$\begin{pmatrix} 0\ 1\ 2\ 3\ 4 \\ 1\ 2\ 3\ 0\ 4 \end{pmatrix} = (3\ 0\ 1\ 2)(4)$$

なお，巡回置換ベクトルに単項関数Ｃ．を作用させれば，直接置換ベクトルが得られる．つまり単項関数Ｃ．は直接置換と巡回置換との切り替えを行う．

　２項関数Ｃ．ではこのような巡回置換ベクトルとしてボックス化した値を用いて巡回置換も可能である．

　(3 0 1 2; 4) C. 'abcde'

bcdae

　もう少し要素の多い場合として，高木貞治先生の古典的名著「代数学講義」p. 174 より例をとりやってみよう．

　C. ４５８９２３１６７

```
+-+---+-----+-------+
|0|5 2|8 6 3|9 7 1 4|
+-+---+-----+-------+
```

$$\begin{pmatrix} 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ 4\ 5\ 8\ 9\ 2\ 3\ 1\ 6\ 7 \end{pmatrix} = (1\ 4\ 9\ 7)(2\ 5)(3\ 8\ 6)$$

前にも述べたが，一般の数学書では１－オリジンであるが，不変の０の環が追加されるだけであって，Ｊの関数はそのまま有効である．文字データの場合は空白文字を追加すれば良い．

```
   0 4 5 8 9 2 3 1 6 7 C. ’abcdefghi’
 dehibcafg
```

また，巡回置換は順次（in succession）行っても変わらない．

```
   (<9 7 1 4) C. ’abcdefghi’
 dbciefahg
   (<8 6 3) C. ’dbciefahg’
 dbhiecafg
   (<5 2) C. ’dbhiecafg’
 dehibcafg
```

別のデータではつぎのようになる．

```
   C. 3 4 5 6 7 8 9 1 2
+-+-----------------+
|0|9 2 4 6 8 1 3 5 7|
+-+-----------------+
```

$$\begin{pmatrix} 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 1\ 2 \end{pmatrix} = (1\ 3\ 5\ 7\ 9\ 2\ 4\ 6\ 8)$$

```
   C. 9 6 3 2 1 8 5 4 7
+-+-+-------+-------+
|0|3|8 4 2 6|9 7 5 1|
+-+-+-------+-------+
```

$$\begin{pmatrix} 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ \\ 9\ 6\ 3\ 2\ 1\ 8\ 5\ 4\ 7 \end{pmatrix} = (1\ 9\ 7\ 5)(2\ 6\ 8\ 4)(3)$$

このようにＪは組合わせ数学の演算操作においても便利なツールである.

参考書　高木貞治「改訂－代数学講義」p. 174 共立出版（再版2000）.

Robert R. Korfhage, “Discrete Computational Structures - Chapter 4,Combinatorics”, Academic Press (1984).