

I. を用いた Merge

Masato Shimura

2006 年 4 月 19 日

1 I. Indices, Interval index

J5 から I. がサポートされた。次のような機能がある。

単項	両項
ベクトルの指標ナンバーを求める a 1 1 0 1 0 1 0 1 I. a 0 1 3 5 7 次と同じであるが、些か簡潔に書ける。 a # i. # a 0 1 3 5 7	x. で示した数値との各個の比較を行い、 結果を合わせて示す。 10?.20 6 3 19 15 10 14 0 7 12 17 3 10 </ 10?.20 1 0 1 1 1 1 0 1 1 1 0 0 1 1 0 1 0 0 1 1 3 10 I. 10?.20 1 0 2 2 1 2 0 1 2 2

*1

2 Expand

これを用いて、*expamd* と *merge* を作ってみた。

expand は 両項の *i*. で指標の位置を求めるところと、指標から外れた箇所に何かを挿入するところが肝要である。

expand は APL にあって J でサポートされていない関数である。

```
a=: 1 1 0 1 0 1 0 1
```

```
a expand i. 5 3
```

```
0 1 2
3 4 5
0 0 0
6 7 8
0 0 0
9 10 11
0 0 0
12 13 14
```

3 merge

マージは *expand* で 0 のベクトルに 0 等を入れていたのを正規のデータを順に挿入するようにしたものである。

```
a1=: (i. 5 3); 100 +i. 3 3
```

*1 I. は古いバージョンで積分の隠し関数として用いられたことがある。積分はライブラリーに収録された

```

a1
+-----+-----+
| 0  1  2|100 101 102|
| 3  4  5|103 104 105|
| 6  7  8|106 107 108|
| 9 10 11|           |
|12 13 14|           |
+-----+-----+

```

```
a merge a1
```

```

0  1  2
3  4  5
100 101 102
6  7  8
103 104 105
9  10 11
106 107 108
12 13 14

```

merge はここでは 01 のベクトルを指標としているが、CODE で管理する方法もある。

3.1 Script

```
expand=:4 : '((I. a) i. i. # x.) { y.,0'
```

```
merge=: 4 : '(((I. x.),I. -. x.) i. i. +/;#L:0 y.){ ;,. y.'
```

```
NB. Usage: a expand i. 5 3
```

```
NB. Usage: a merge a1
```

```
a1=: (i. 5 3); 100 +i. 3 3
```

```
a=: 1 1 0 1 0 1 0 1
```