

DIALOG

Elsinore 2023

Leading Axis Theory and Practice

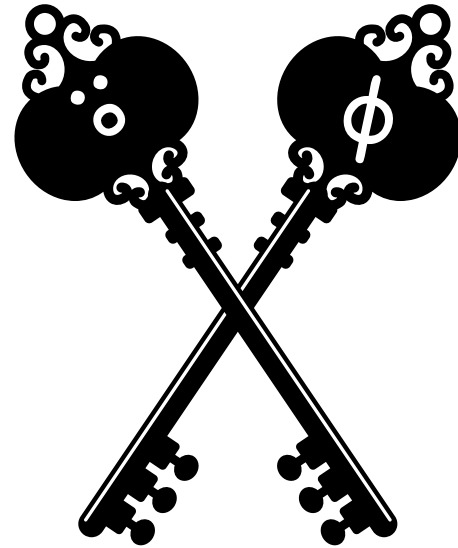
Rich Park

Adám Brudzewsky



asst. Josh David

Rank and Dyadic Transpose are the Keys to the Array Kingdom™



Leading Axis Overview

Theory

The Rank Operator

Leading Axis Functions

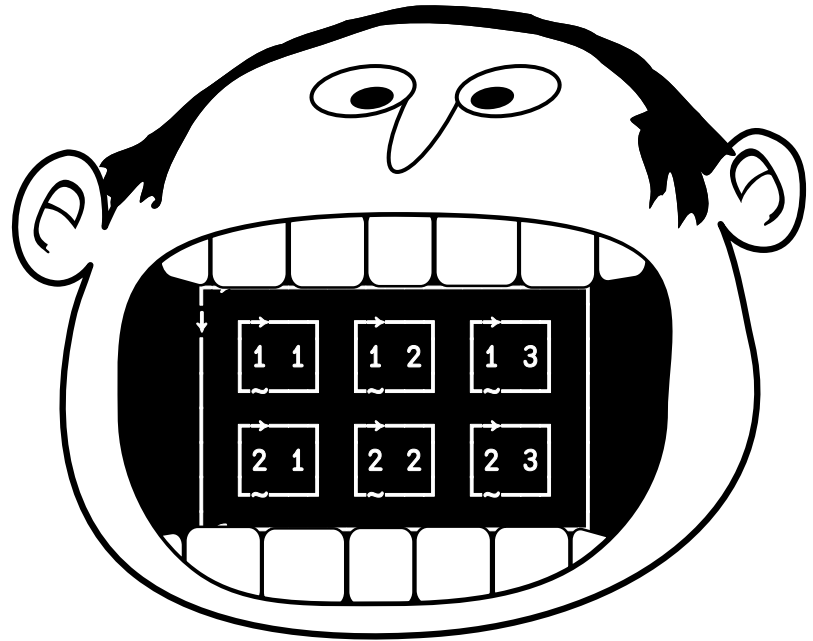
Leading Axis Operators

Dyadic Transpose

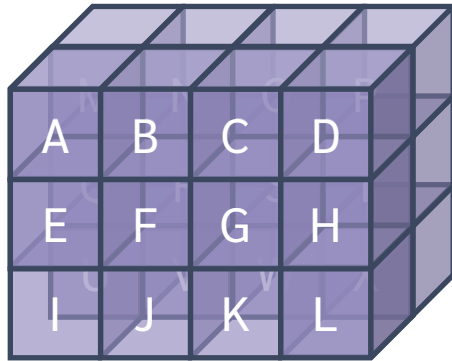
Performance



Make Functions apply to Whole Arrays™



Theory

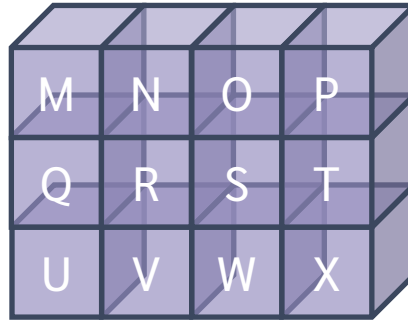
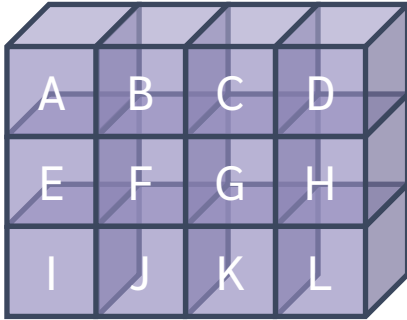


Array of rank 3

$A \leftarrow 2 \ 3 \ 4 \rho \square A$



Theory



Array of rank 3

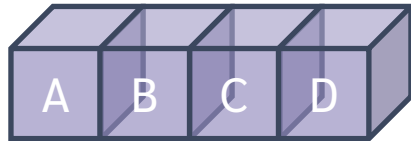
2-Cell: layer*

A ← 2 3 4p□A

* Major cell



Theory



A ← 2 **3** 4p□A

Array of rank 3

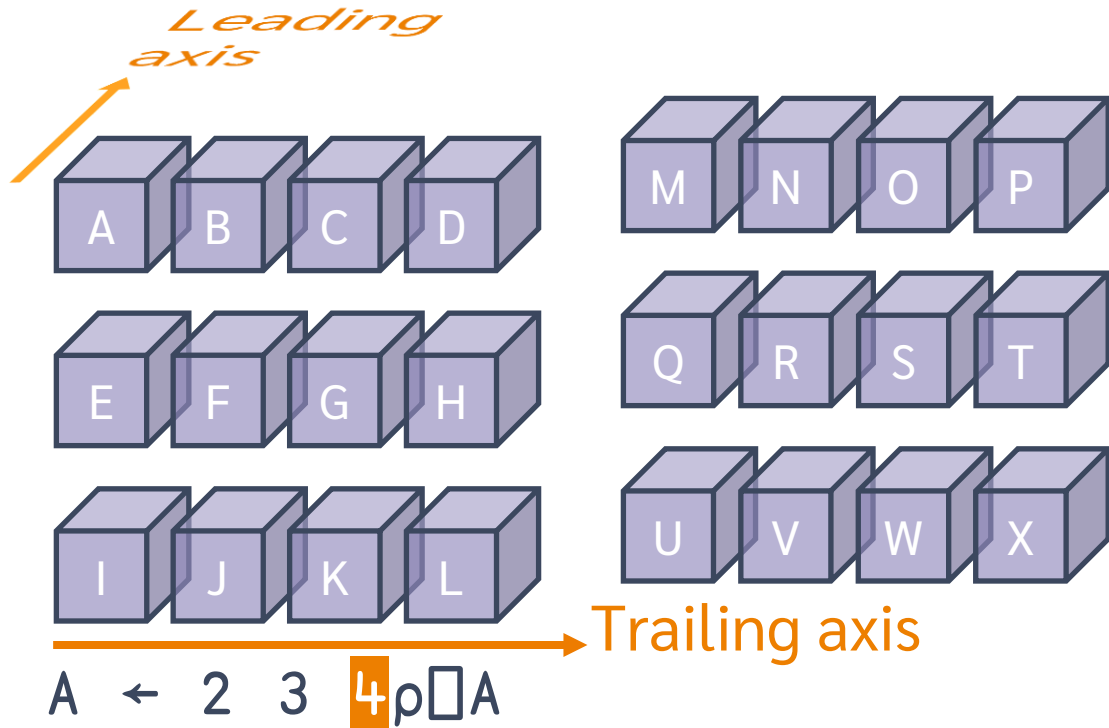
2-Cell: layer*

1-Cell: row

* Major cell



Theory



Array of rank 3

2-Cell: layer*

1-Cell: row

0-Cell: element

* Major cell



Theory

Array of rank N

collection of arrays of rank $N-1$

K -Cell

sub-array of rank K

Major Cell

$N-1$ -Cell

Leading axis

first dimension; “list” of major cells

Trailing axis

last dimension; “list” of scalar elements



Discussion: Leading-Axis Primitives

Functions

$X \neq Y$ $X \neq Y$

$X \uparrow Y$ $X \downarrow Y$ $X \bar{=} Y$

$\ominus Y$ $X \ominus Y$

$\neq Y$ $X \square Y$ ΔY ΨY

$\neq Y$ $\cup Y$ $X \wr Y$ $X \underline{\wr} Y$

Operators

\neq \neq

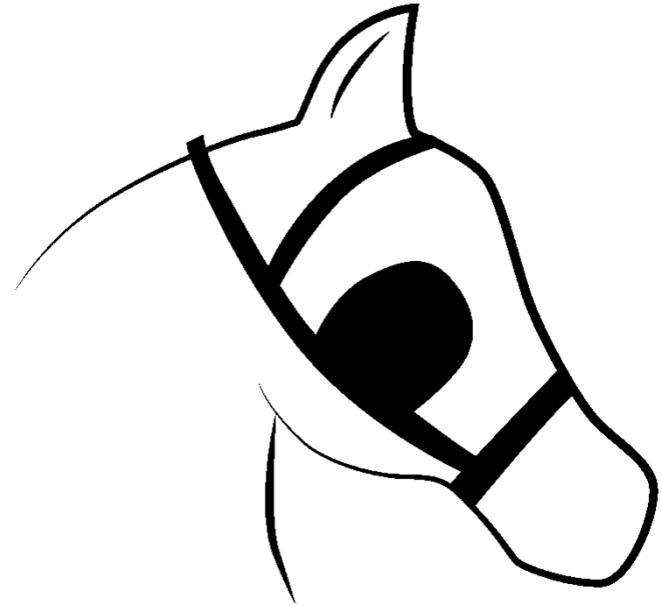
@

≡

⊠



The Rank Operator is just Blinkers™



The Rank Operator ($f \circ M$)

- f sees **arg** of **max rank M**
- If necessary, f will be called multiple times

Pro tip to avoid stranding of M and Y : $(f \circ M) Y$ or
 $f (\circ M) Y$ or
 $f \circ M \vdash Y$



Rank Values

Sub-Arrays	Rank
Elements/Scalars	0
Rows/Vectors	1
Layers/Matrices	2
Blocks	3
⋮	⋮



Task: Exchange the First Two...

Consider 'ABCdef' → 'BACdef'

- ◆ layers $A \leftarrow 3 \quad 5 \quad 4 \quad 1 \quad 6 \quad 0$
- ◆ rows of each layer
- ◆ columns (elements of each row)

Bonus question: How would you do it without $\ddot{?}$?



The Rank Operator ($f \circ L \ R$)

- f sees **left arg** of max rank L
- f sees **right arg** of max rank R
- If necessary, sub-arrays will be used multiple times

Pro tips: If L and R are the same then $f \circ R$ is enough
If confused, try $\{\alpha\omega\} \circ L \ R$



Discussion: Whole Array Primitives

Functions

$$X \equiv Y$$

$$X \neq Y$$

$$\subset Y$$

$$\rho Y$$

$$\uparrow Y$$

Operators

$$A \circ f$$

$$f \circ B$$

$$A \rightsquigarrow$$



Task: Using \equiv \circ L R ...

`names ← ↑(, 'Hey' 'Jay' ◦ {↑αω} 'ley' 'den') [? 1 0 ρ 4]`

- Boolean vector indicating layers that are 2 3 ρ 'Hey ley'
- Boolean matrix indicating rows that are 'Hey'
- Boolean rank-3 array indicating elements that are 'e'

Bonus: Think about what the relationship is between \equiv and $=$



Double-Rank $((f \circ L_2 \ R_2) \circ L_1 \ R_1)$

'ABC' (, 0 1) 'xy'

Axy

Bxy

Cxy

'ABC' ((, 0) 0 1) 'xy'

Ax

Ay

Bx

By

Cx

Cy

Remember: $L=R$



Task: Using (\times \circ L2 R2) \circ L1 R1 ...

A \leftarrow 1 10 100
B \leftarrow 2 3 4 ρ τ 24

Produce:

	1	2	3	4
	50	60	70	80
	900	1000	1100	1200
	13	14	15	16
	170	180	190	200
	2100	2200	2300	2400



Dyadic Transpose ($X\phi Y$): Why?

- ◆ Dyadic Transpose reorders axes
- ◆ Move axes to end, then use Rank on any function, operating on only those trailing axes



Dyadic Transpose ($X\phi Y$): How?

Reorders axes by the left argument:

Each number is the destination index where we *send* that axis.



Tasks: Merging Layers...

2 3 4p□A

ABCD
EFGH
IJKL

MNOP
QRST
UVWX

1. horizontally:

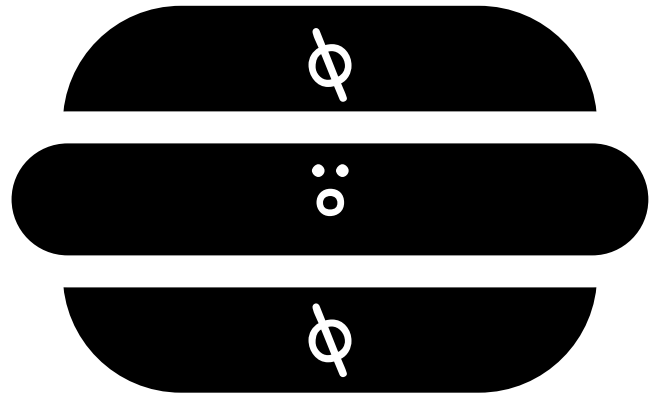
ABCDMNOP
EFGHQRST
IJKLUVWX

2. vertically:

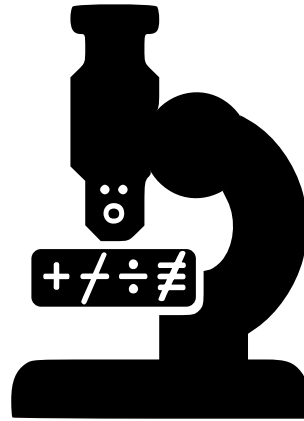
ABCD
EFGH
IJKL
MNOP
QRST
UVWX



Transpose, Apply-with-Rank, Transpose™



Keep Rank Close to the Primitives™



Task: Speed it Up!

```
n ← ? 1 10 100 1000 ρ 10  
a ← '{ + / ω / ~ 1 0 ρ ~ ≠ ω } ö d † n '  
b ← '... your code here ...'
```

```
'cmpx' □ CY 'dfns'
```

```
d ← 1  ◇  0  0  0  0.1  cmpx  a  b  
d ← 2  ◇  0  0  0  0.1  cmpx  a  b  
d ← 3  ◇  0  0  0  0.1  cmpx  a  b  
d ← 4  ◇  0  0  0  0.1  cmpx  a  b
```



Lessons to Take Home

Rank and Dyadic Transpose are the Keys to the Array Kingdom

Make Functions apply to Whole Arrays

The Rank Operator is just Blinkers

Transpose, Rank-Apply-with-Rank, Transpose

Keep Rank Close to the Primitives

