Einstein summation and reductions

READ THE OUTPUT INDICES

Retain the listed axes in order. Reduce the rest. `:=` uses `sum`. Repeated names pair equal positions on axes of equal length. `!` selects a singleton axis.

Select, reorder and broadcast

$A = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 1 & 4 \end{bmatrix}$, $D = \begin{bmatrix} 2 & 5 \\ 7 & 3 \end{bmatrix}$, $u = [1, 2]$, $U = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

```
square[.i, .j] := A[.i, .j]^2
```
$\begin{bmatrix} 1 & 4 & 0 \\ 9 & 1 & 16 \end{bmatrix}$

```
rows[.i] := A[.i, .j]
```
$[3, 8]$

```
cols[.j] := A[.i, .j]
```
$[4, 3, 4]$

```
total[] := A[.i, .j]
```
11

```
swap[.j, .i] := A[.i, .j]
```
$\begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 0 & 4 \end{bmatrix}$

```
second[.i] := A[.i, 2]
```
$[2, 1]$

```
diag_cells[.i] := D[.i, .i]
```
$[2, 3]$

```
diag_total[] := D[.i, .i]
```
5

```
add_col[.i, .j] := A[.i, .j] + u[.i]
```
$\begin{bmatrix} 2 & 3 & 1 \\ 5 & 3 & 6 \end{bmatrix}$

```
scale_rows[.i, .j] := (
    A[.i, .j] * U[.i, !])
```
$\begin{bmatrix} 1 & 2 & 0 \\ 6 & 2 & 8 \end{bmatrix}$

Multiply and contract

$B = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$, $C = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$, $u = [1, 2]$, $v = [3, 4]$, $w = [2, 1]$

```
dot_uv[] := u[.i] * v[.i]
```
11

```
outer_uv[.i, .j] := u[.i] * v[.j]
```
$\begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$

```
mv[.i] := B[.i, .j] * v[.j]
```
$[11, 13]$

```
mm[.i, .k] := B[.i, .j] * C[.j, .k]
```
$\begin{bmatrix} 4 & 6 \\ 7 & 3 \end{bmatrix}$

```
hadamard[.i, .j] := B[.i, .j] * C[.i, .j]
```
$\begin{bmatrix} 2 & 0 \\ 3 & 3 \end{bmatrix}$

```
row_dot[.i] := B[.i, .j] * C[.i, .j]
```
$[2, 6]$

```
frobenius[] := B[.i, .j] * C[.i, .j]
```
8

```
mm_w[.i, .k] := (
    B[.i, .j] * w[.j] * C[.j, .k])
```
$\begin{bmatrix} 6 & 6 \\ 13 & 3 \end{bmatrix}$

```
loss[.i, .k] := (
    (B[.i, .j] - C[.j, .k])^2 * w[.j])
```
$\begin{bmatrix} 3 & 3 \\ 2 & 22 \end{bmatrix}$

```
bilinear[] := u[.i] * B[.i, .j] * v[.j]
```
37

Read one cell of the matrix product

$$\underbrace{\begin{bmatrix} 3 & 1 \end{bmatrix}}_{\text{row 2 of } B} \cdot \underbrace{\begin{bmatrix} 2 \\ 1 \end{bmatrix}}_{\text{column 1 of } C} = 3 \cdot 2 + 1 \cdot 1 = 7$$

Fix `.i` at 2 and `.k` at 1. Pair entries at the same `.j`, then sum over `.j`.

Check the output axes

Each output name occurs in the expression and appears once on the left. With no omitted axes, `:=` returns one value per output position.

Here A is 2×3. Keeping `.i` returns two row sums. Keeping `.j` returns three column sums. Keeping `.j, .i` returns a 3×2 transpose.

```
rows = aggregate(sum, [.i], A[.i, .j])
```

Index rank-three arrays

$X_1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $X_2 = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$, $V = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$

$H_1 = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$, $H_2 = \begin{bmatrix} 4 & 5 & 6 \end{bmatrix}$, $q = [1, 0, 1]$. See C at left.

X is $2 \times 2 \times 2$. H is $2 \times 1 \times 3$. Paired matrices are successive slices along the first axis.

```
first_slice[.i, .j] := X[1, .i, .j]
```
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

```
slice_sums[.b] := X[.b, .i, .j]
```
$[10, 26]$

```
middle[.b, .j] := X[.b, .i, .j]
```
$\begin{bmatrix} 4 & 6 \\ 12 & 14 \end{bmatrix}$

```
traces[.b] := X[.b, .i, .i]
```
$[5, 13]$

```
rotate[.j, .b, .i] := X[.b, .i, .j]
```
$\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$ $\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$

```
batch_mv[.b, .i] := (
    X[.b, .i, .j] * V[.b, .j])
```
$\begin{bmatrix} 5 & 11 \\ 16 & 22 \end{bmatrix}$

```
batch_mm[.b, .i, .k] := (
    X[.b, .i, .j] * C[.j, .k])
```
$\begin{bmatrix} 4 & 6 \\ 10 & 12 \end{bmatrix}\begin{bmatrix} 16 & 18 \\ 22 & 24 \end{bmatrix}$

```
norm_sq[] := X[.b, .i, .j]^2
```
204

```
rect_sum[.b, .j] := H[.b, .i, .j]
```
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

```
rect_mv[.b, .i] := H[.b, .i, .j] * q[.j]
```
$\begin{bmatrix} 4 \\ 10 \end{bmatrix}$

Choose a reduction

$W = \begin{bmatrix} 1 & 2 & 6 \\ 2 & 4 & 6 \end{bmatrix}$. Sample `var` uses $n - 1$. `std` is its square root. T and F denote Boolean values.

Substitute each reducer name for `f` to reduce by row.

```
aggregate(f, [.i], W[.i, .j])
```

`sum`	$[9, 12]$	`prod`	$[12, 48]$
`minimum`	$[1, 2]$	`maximum`	$[6, 6]$
`mean`	$[3, 4]$	`median`	$[2, 4]$
`var`	$[7, 4]$	`std`	$[\sqrt{7}, 2]$

```
aggregate(lany, [.j], W[.i, .j] >= 4)
```
$[F, T, T]$

```
aggregate(lall, [.j], W[.i, .j] >= 4)
```
$[F, F, T]$

```
aggregate(maximum, [], W[.i, .j])
```
6

```
aggregate(lall, [], W[.i, .j] < 6)
```
F

```
hits[.i] := ifelse(
    W[.i, .j] >= 4, 1, 0)
```
$[1, 2]$

```
pair_prod = aggregate(prod, [.i, .k],
    B[.i, .j] + C[.j, .k])
```
$\begin{bmatrix} 9 & 5 \\ 10 & 12 \end{bmatrix}$

Reduce in stages

$R = \begin{bmatrix} 1 & 6 \\ 3 & 2 \end{bmatrix}$. Sample covariance here divides by $n - 1 = 1$. Mark all tied column maxima in W.

```
mu = aggregate(mean, [.j], R[.i, .j])
```
$[2, 4]$

```
E[.i, .j] := R[.i, .j] - mu[.j]
```
$\begin{bmatrix} -1 & 2 \\ 1 & -2 \\ 2 & -4 \\ -4 & 8 \end{bmatrix}$

```
cov[.j, .k] := E[.i, .j] * E[.i, .k]
```

```
peak = aggregate(maximum, [.j],
    W[.i, .j])
```
$[2, 4, 6]$

```
ties = aggregate(lany, [.i, .j],
    W[.i, .j] >= peak[.j])
```
$\begin{bmatrix} F & F & T \\ T & T & T \end{bmatrix}$

Gather by position

$p = [2, 1, 2]$, $c = [3, 1]$. Repeated positions repeat entries. `:` selects a full axis. Positions start at 1.

```
pick = u[p]
```
$[2, 1, 2]$

```
take = A[:, c]
```
$\begin{bmatrix} 0 & 1 \\ 4 & 3 \end{bmatrix}$

```
along[.i] := take[.i, 1]
```
$[0, 4]$

```
subgrid = A[p, c]
```
$\begin{bmatrix} 4 & 3 \\ 0 & 1 \\ 4 & 3 \end{bmatrix}$

```
by_get[.j] := get(A, 1, .j)
```
$[1, 2, 0]$

Reduce by group

$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ $x = [3, 2, 5, 4]$, $a = [1, 1, 2, 1]$. Columns of M mark membership. Use nonzero group counts and weight sums for the means.

```
ng[.k] := M[.i, .k]
```
$[2, 2]$

```
totg[.k] := M[.i, .k] * x[.i]
```
$[8, 6]$

```
wg[.k] := M[.i, .k] * a[.i]
```
$[3, 2]$

```
atot[.k] := M[.i, .k] * a[.i] * x[.i]
```
$[13, 6]$

```
avg = totg ./ ng
```
$[4, 3]$

```
wavg = atot ./ wg
```
$[\frac{13}{3}, 3]$

```
topg = aggregate(maximum, [.k],
    ifelse(M[.i, .k] >= 1, x[.i], -inf))
```
$[5, 4]$

Change the pair rule or reduction

$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $Q = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ `paths` tests for a two-step path through any `.j`.

```
minplus = aggregate(minimum, [.i, .k],
    B[.i, .j] + C[.j, .k])
```
$\begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix}$

```
maxplus = aggregate(maximum, [.i, .k],
    B[.i, .j] + C[.j, .k])
```
$\begin{bmatrix} 3 & 5 \\ 5 & 4 \end{bmatrix}$

```
maxprod = aggregate(maximum, [.i, .k],
    B[.i, .j] * C[.j, .k])
```
$\begin{bmatrix} 2 & 6 \\ 6 & 3 \end{bmatrix}$

```
paths = aggregate(lany, [.i, .k],
    P[.i, .j] * Q[.j, .k] >= 1)
```
$\begin{bmatrix} T & T \\ T & F \end{bmatrix}$

```
triple[.i, .l] := (
    B[.i, .j] * C[.j, .k] * B[.k, .l])
```
$\begin{bmatrix} 22 & 14 \\ 16 & 17 \end{bmatrix}$

```
loop3[] := (
    B[.i, .j] * C[.j, .k] * B[.k, .i])
```
39

Retain four axes

K uses B and C. The four 2×2 blocks are indexed by `(.i, .j)`. Each block shows `(.k, .l)`.

```
K[.i, .j, .k, .l] := (
    B[.i, .j] * C[.k, .l])
```
$\begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$ $\begin{bmatrix} 4 & 0 \\ 2 & 6 \end{bmatrix}$

$\begin{bmatrix} 6 & 0 \\ 3 & 9 \end{bmatrix}$ $\begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$

```
kplane[.k, .l] := K[2, 1, .k, .l]
```
$\begin{bmatrix} 6 & 0 \\ 3 & 9 \end{bmatrix}$

```
ksum[.i, .j] := K[.i, .j, .k, .l]
```
$\begin{bmatrix} 6 & 12 \\ 18 & 6 \end{bmatrix}$

```
ktrace[] := K[.i, .i, .k, .k]
```
10

Jessica Cox · bcox@mpp.mpg.de