Matrices are Arrows! an AOP perspective on (typed) linear algebra

H.S. Macedo
(Advisor) J.N. Oliveira

Dept. Informática,
Universidade do Minho
Braga, Portugal

May 8, 2009

Context and Motivation

- The advent of on-chip **parallelism** poses many challenges to current programming languages.
- Traditional approaches based on compiler + hand-coded optimization are giving place to trendy generative techniques, based on DSLs for high-level program **transformation**.
- In areas such as scientific computing, image/video processing, the bulk of the work performed by so-called **kernel** functions.
- Examples of kernels are matrix-matrix multiplication (MMM), the discrete Fourier transform (DFT), etc.
- Kernel **optimization** has become extraordinarily difficult due to the complexity of current computing platforms.

Teaching computers to write fast numerical code

In the SPIRAL Group (CMU), a DSL has been defined (**OL**) [1] to specify kernels in a data-independent way.

- **OL** is derived from mathematics (thus declarative) and describes the structure of a computation in an implementation-independent form. **Divide-and-conquer** algorithms are described as **OL** breakdown rules.
- By recursively applying these rules a space of algorithms for a desired kernel can be generated.

Rationale behind SPIRAL:

- Target imperative code is too late for numeric processing kernel optimization.
- Such optimization can be elegantly and efficiently performed well above in the design chain once the maths themselves are expressed in an **index-free** style.

Synergy

- Parallel between the pointfree notation of **OL** and **relational algebra** is obvious.
- Rich calculus of algebraic rules.
- Relational calculus is typed once relations are regarded as arrows in the **Rel** allegory.
- What about the matrix calculus?

Sample of **OL** (Operator language) [1]

name	definition
Linear, arity (1,1)	
identity	$I_n : \mathbb{C}^n \rightarrow \mathbb{C}^n;\ \mathbf{x} \mapsto \mathbf{x}$
vector flip	$J_n : \mathbb{C}^n \rightarrow \mathbb{C}^n;\ (x_i) \mapsto (x_{n-i})$
transposition of an $m \times n$ matrix	$L^{mn}_m : \mathbb{C}^{mn} \rightarrow \mathbb{C}^{mn};\ \mathbf{A} \mapsto \mathbf{A}^T$
matrix $M \in \mathbb{C}^{m \times n}$	$\mathrm{M} : \mathbb{C}^n \rightarrow \mathbb{C}^m; \mathbf{x} \mapsto M\mathbf{x}$
Multilinear, arity (2,1)	
Point-wise product	$\mathrm{P}_n : \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{C}^n;\ ((x_i),(y_i)) \mapsto (x_i y_i)$
Scalar product	$\mathrm{S}_n : \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{C};\ ((x_i),(y_i)) \mapsto \Sigma(x_i y_i)$
Kronecker product	$\mathrm{K}_{m \times n} : \mathbb{C}^m \times \mathbb{C}^n \rightarrow \mathbb{C}^{mn};\ ((x_i),\mathbf{y})) \mapsto (x_i \mathbf{y})$
Others	
Fork	$\mathrm{Fork}_n : \mathbb{C}^n \rightarrow \mathbb{C}^n \times \mathbb{C}^n;\ \mathbf{x} \mapsto (\mathbf{x},\mathbf{x})$
Split	$\mathrm{Split}_n : \mathbb{C}^n \rightarrow \mathbb{C}^{n/2} \times \mathbb{C}^{n/2};\ \mathbf{x} \mapsto (\mathbf{x}^U, \mathbf{x}^L)$
Concatenate	$\oplus_n : \mathbb{C}^n \times \mathbb{C}^m \rightarrow \mathbb{C}^{n+m};\ (\mathbf{x}\,,\mathbf{y}) \mapsto \mathbf{x} \oplus \mathbf{y}$
Duplication	$\mathrm{dup}^m_n : \mathbb{C}^n \rightarrow \mathbb{C}^{nm};\ (\mathbf{x} \mapsto \mathbf{x} \otimes I_m$
Min	$\mathrm{min}_n : \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{C}^n;\ (\mathbf{x}\,,\mathbf{y}) \mapsto (\min(x_i, y_i))$
Max	$\mathrm{max}_n : \mathbb{C}^n \times \mathbb{C}^n \rightarrow \mathbb{C}^n;\ (\mathbf{x}\,,\mathbf{y}) \mapsto (\max(x_i, y_i))$

Table 1. Definition of basic operators. The operators are assumed to operate on complex numbers but other base sets are possible. Boldface fonts represent vectors or matrices linearized in memory. Superscripts U and L represent the upper and lower half of a vector. A vector is sometimes written as $\mathbf{x} = (x_i)$ to identify the components.

Comments

We believe a number of improvements can be performed concerning

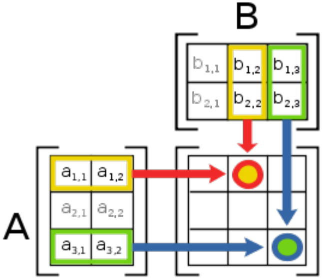
- the OL notation itself
- the layout of its calculus
- Its effectiveness for calculation/transformation purposes

By the way:

- Looking at linear algebra textbooks we see a diversity of approaches, ways of defining/describing kernel algorithms.
- Textbooks often explain matrix operations by resorting to `for`-loop notation.
- How "algebraic" is this?

MMM as inspiration about what to do

From the Wikipedia:



Index-wise definition

$$C_{ij} = \sum_{k=1}^{2} A_{ik} \times B_{kj}$$

Hiding indices i, j, k:

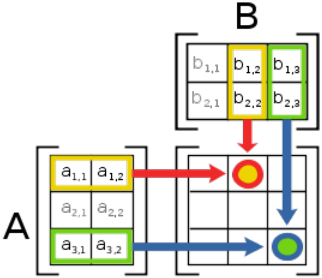
$$\begin{tikzcd} 3 & 2 \arrow[l, "A"'] & 3 \arrow[l, "B"'] \arrow[ll, bend left, "A \cdot B"] \end{tikzcd}$$

Index-free

$$C = A \cdot B$$

MMM as inspiration about what to do

From the Wikipedia:



Index-wise definition

$$C_{ij} = \sum_{k=1}^{2} A_{ik} \times B_{kj}$$

Hiding indices i, j, k:

$$\begin{tikzcd} 3 & 2 \arrow[l, "A"'] & 3 \arrow[l, "B"'] \arrow[ll, bend left, "A \cdot B"] \end{tikzcd}$$

Index-free

$$C = A \cdot B$$

Matrices are Arrows

Given

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}_{m \times n} \qquad m \xleftarrow{A} n$$

$$B = \begin{bmatrix} b_{11} & \dots & b_{1k} \\ \vdots & \ddots & \vdots \\ b_{u1} & \dots & b_{nk} \end{bmatrix}_{n \times k} \qquad n \xleftarrow{B} k$$

Define

$$\begin{tikzcd} m & n \arrow[l, "A"'] & k \arrow[l, "B"'] \arrow[ll, bend left, "A \cdot B"] \end{tikzcd}$$

Matrices are Arrows

Given

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}_{m \times n}$$

$$m \xleftarrow{A} n$$

$$B = \begin{bmatrix} b_{11} & \dots & b_{1k} \\ \vdots & \ddots & \vdots \\ b_{u1} & \dots & b_{nk} \end{bmatrix}_{n \times k}$$

$$n \xleftarrow{B} k$$

Define

$$\begin{tikzcd} m & n \arrow[l, "A"'] & k \arrow[l, "B"'] \arrow[ll, bend left, "A \cdot B"] \end{tikzcd}$$

Category of matrices

As guessed above:

- Under MMM $(A \cdot B)$, matrices form a **category** whose **objects** are matrix dimensions and whose **morphisms** $m \xleftarrow{A} n$, $n \xleftarrow{B} k$ are the matrices themselves.
- Every identity $n \xleftarrow{id} n$ is the diagonal of size n, that is, $id(r, c) \triangleq r = c$ under the $(0, 1)$ encoding of the Booleans:

$$id_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{n \times n} \qquad n \xleftarrow{id_n} n$$

Category of matrices

Looking closer:

- Such a category is **Abelian** — every homset forms an aditive Abelian group (**Ab**-category) such that composition is bilinear relative to $+$:

$$M \cdot (N + L) = M \cdot N + M \cdot L \tag{1}$$
$$(N + L) \cdot K = N \cdot K + L \cdot K \tag{2}$$

- It has **biproducts**, where a biproduct diagram

$$\begin{tikzcd} a \arrow[r, shift right, "i_1"'] & c \arrow[l, shift right, "\pi_1"'] \arrow[r, shift left, "\pi_2"] & b \arrow[l, shift left, "i_2"] \end{tikzcd} \tag{3}$$

is such that

$$\pi_1 \cdot i_1 = id_a \tag{4}$$
$$\pi_2 \cdot i_2 = id_b \tag{5}$$
$$i_1 \cdot \pi_1 + i_2 \cdot \pi_2 = id_c \tag{6}$$

Biproduct=product+coproduct

Theorem:

" Two objects a and b in Ab-category A have a **product** in A iff they have a biproduct in A. Specifically, given a biproduct diagram, the object c with the projections π_1 and π_2 is a product of a and b, while, dually, c with i_1 and i_2 is a **coproduct**."
(MacLane [2], pg. 194)

"Deja vu"?

Yes, in relation algebra, for $\pi_1 = i_1^\circ$ and $\pi_2 = i_2^\circ$:

$[R\ ,S]=(R \cdot i_1^\circ) \cup (S \cdot i_2^\circ)$ cf.

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[dr, "R"'] & A+B \arrow[d, "{[R\ ,S]}"] & B \arrow[l, "i_2"'] \arrow[dl, "S"] \\
& C &
\end{tikzcd}$$

Biproduct=product+coproduct

Theorem:

" Two objects a and b in Ab-category A have a **product** in A iff they have a biproduct in A. Specifically, given a biproduct diagram, the object c with the projections π_1 and π_2 is a product of a and b, while, dually, c with i_1 and i_2 is a **coproduct**."
(MacLane [2], pg. 194)

"Deja vu"?

Yes, in relation algebra, for $\pi_1 = i_1^\circ$ and $\pi_2 = i_2^\circ$:

$[R \ , S] = (R \cdot i_1^\circ) \cup (S \cdot i_2^\circ)$ cf.

$$\begin{tikzcd} A \arrow[r, "i_1"] \arrow[dr, "R"'] & A+B \arrow[d, "{[R\ ,S]}"] & B \arrow[l, "i_2"'] \arrow[dl, "S"] \\ & C & \end{tikzcd}$$

Deja vu

In fact, within relations

$$\begin{aligned} i_1^\circ \cdot i_1 &= id \\ i_2^\circ \cdot i_2 &= id \end{aligned}$$

meaning that $i_{k=1,2}$ are injections (**kernels** both reflexive and coreflexive) and

$$i_1 \cdot i_1^\circ \cup i_2 \cdot i_2^\circ = id$$

meaning that they are jointly surjective (**images** together are reflexive and coreflexive).
In linear algebra, however, **biproducts** are far many and more interesting! Let us see why.

The Puzzle

The biproduct definition is declarative, in order to build upon this concept, we need to re-interpret its axioms constructively. To do so is to solve the following equation system:

$$\left\{\begin{array}{lcl} \pi_1 \cdot i_1 & = & id_a \\ \pi_2 \cdot i_2 & = & id_b \\ i_1 \cdot \pi_1 + i_2 \cdot \pi_2 & = & id_c \end{array}\right.$$

"In other words, the equations [above] contain the familiar calculus of matrices." *Mac Lane*

The "standard" biproduct

In this biproduct, coproduct is column-wise join $\left[\begin{array}{c|c} A & B \end{array}\right]$ and product is row-wise join $\left\langle\frac{A}{B}\right\rangle$. As in relation algebra, $\pi_1 = i_1^\circ$ and $\pi_2 = i_2^\circ$, where M° is the **transpose** of M, and:

$$\left[\begin{array}{c|c} A & B \end{array}\right] = A \cdot \pi_1 + B \cdot \pi_2 \tag{7}$$

$$\left\langle\frac{A}{B}\right\rangle = \left[\begin{array}{c|c} A^\circ & B^\circ \end{array}\right]^\circ \tag{8}$$

Thus

$$\left\langle\frac{A}{B}\right\rangle = i_1 \cdot A + i_2 \cdot B \tag{9}$$

Diagram

$$
\begin{tikzcd}
 & k \arrow[dl, "A"'] \arrow[d, "\left\langle \frac{A}{B} \right\rangle"] \arrow[dr, "B"] & \\
m \arrow[r, "i_1"] \arrow[dr, "C"'] & m+n \arrow[d, "\left[\, C \mid D \,\right]"] & n \arrow[l, "i_2"'] \arrow[dl, "D"] \\
 & j &
\end{tikzcd}
$$

Universal properties

Product:

$$X = \left\langle \frac{A}{B} \right\rangle \equiv \begin{cases} \pi_1 \cdot X = A \\ \pi_2 \cdot X = B \end{cases} \tag{10}$$

Coproduct:

$$X = \left[\; A \,\middle|\, B \;\right] \equiv \begin{cases} X \cdot i_1 = A \\ X \cdot i_2 = B \end{cases} \tag{11}$$

Both:

$$X = \left(\begin{array}{c|c} A & C \\ \hline B & D \end{array} \right) \equiv \begin{cases} \pi_1 \cdot X \cdot i_1 = A \\ \pi_1 \cdot X \cdot i_2 = C \\ \pi_2 \cdot X \cdot i_1 = B \\ \pi_2 \cdot X \cdot i_2 = D \end{cases} \tag{12}$$

Elementary matrix AOP

Reflection

$$\left\langle \frac{\pi_1}{\pi_2} \right\rangle = id \tag{13}$$

$$\left[\; i_1 \;\middle|\; i_2 \;\right] = id \tag{14}$$

Fusion

$$\left\langle \frac{A}{B} \right\rangle \cdot C = \left\langle \frac{A \cdot C}{B \cdot C} \right\rangle \tag{15}$$

$$C \cdot \left[\; A \;\middle|\; B \;\right] = \left[\; C \cdot A \;\middle|\; C \cdot B \;\right] \tag{16}$$

Abide laws

Not only the **exchange law**

$$\left\langle \frac{\left[\, A \mid B \,\right]}{\left[\, C \mid D \,\right]} \right\rangle = \left[\, \left\langle \frac{A}{C} \right\rangle \middle| \left\langle \frac{B}{D} \right\rangle \,\right] = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right) \tag{17}$$

but also

$$\left\langle \frac{A}{B} \right\rangle + \left\langle \frac{C}{D} \right\rangle = \left\langle \frac{A+C}{B+D} \right\rangle \tag{18}$$

$$\left[\, A \mid B \,\right] + \left[\, C \mid D \,\right] = \left[\, A+C \mid B+D \,\right] \tag{19}$$

Parentheses $\left[\ \ \mid\ \ \right]$ and $\left\langle \frac{\ }{\ } \right\rangle$ will be saved wherever unnecessary.

Putting things to work

Elementary **divide and conquer** matrix multiplication:

$$\left[\; R \;\middle|\; S \;\right] \cdot \left\langle \frac{U}{V} \right\rangle \;=\; R \cdot U + S \cdot V \tag{20}$$

Calculation:

$$
\begin{array}{cl}
 & \left[\; R \;\middle|\; S \;\right] \cdot \left\langle \frac{U}{V} \right\rangle \\
= & \qquad \{ \;\; (9) \;\; \} \\
 & \left[\; R \;\middle|\; S \;\right] \cdot (i_1 \cdot U + i_2 \cdot V) \\
= & \qquad \{ \;\; \text{bilinearity (1)} \;\; \} \\
 & \left[\; R \;\middle|\; S \;\right] \cdot i_1 \cdot U + \left[\; R \;\middle|\; S \;\right] \cdot i_2 \cdot V \\
= & \qquad \{ \;\; +\text{-cancellation} \;\; \} \\
 & R \cdot U + S \cdot V
\end{array}
$$

Putting things to work

Blockwise MMM:

$$\left(\begin{array}{c|c} R & S \\ \hline U & V \end{array}\right) \cdot \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right) = \left(\begin{array}{c|c} RA + SC & RB + SD \\ \hline UA + VC & UB + VD \end{array}\right) \tag{21}$$

Calculation:

$$\left[\left\langle \frac{R}{U} \right\rangle \middle| \left\langle \frac{S}{V} \right\rangle \right] \cdot \left\langle \frac{[\, A \mid B \,]}{[\, C \mid D \,]} \right\rangle$$

$=$ { divide and conquer (20) }

$$\frac{R}{U} \cdot [\, A \mid B \,] + \frac{S}{V} \cdot [\, C \mid D \,]$$

Putting things to work

$$
\begin{aligned}
&= \quad \{\ \text{fusion-}\times\ \} \\
&\left\langle \frac{R \cdot \left[\begin{array}{c|c} A & B \end{array}\right]}{U \cdot \left[\begin{array}{c|c} A & B \end{array}\right]} \right\rangle + \left\langle \frac{S \cdot \left[\begin{array}{c|c} C & D \end{array}\right]}{V \cdot \left[\begin{array}{c|c} C & D \end{array}\right]} \right\rangle \\
&= \quad \{\ \text{fusion-}+\ \} \\
&\left\langle \frac{\left[\begin{array}{c|c} R \cdot A & R \cdot B \end{array}\right]}{\left[\begin{array}{c|c} U \cdot A & U \cdot B \end{array}\right]} \right\rangle + \left\langle \frac{\left[\begin{array}{c|c} S \cdot C & S \cdot D \end{array}\right]}{\left[\begin{array}{c|c} V \cdot C & V \cdot D \end{array}\right]} \right\rangle \\
&= \quad \{\ \text{abide-}\times\ \} \\
&\left\langle \frac{\left[\begin{array}{c|c} R \cdot A & R \cdot B \end{array}\right] + \left[\begin{array}{c|c} S \cdot C & S \cdot D \end{array}\right]}{\left[\begin{array}{c|c} U \cdot A & U \cdot B \end{array}\right] + \left[\begin{array}{c|c} V \cdot C & V \cdot D \end{array}\right]} \right\rangle \\
&= \quad \{\ \text{abide-}+\ \}
\end{aligned}
$$

Putting things to work

$$
\begin{aligned}
& \left\langle \frac{\left[\begin{array}{c|c} R\cdot A + S\cdot C & R\cdot B + S\cdot D \end{array}\right]}{\left[\begin{array}{c|c} U\cdot A + V\cdot C & U\cdot B + V\cdot D \end{array}\right]} \right\rangle \\
= & \qquad \{ \text{ exchange law (17) } \} \\
& \left(\begin{array}{c|c} RA + SC & RB + SD \\ \hline UA + VC & UB + VD \end{array} \right)
\end{aligned}
$$

(Categorial) sum = product

Definitions:

$$A \oplus B \;=\; \left[\; i_1 \cdot A \;\middle|\; i_2 \cdot B \;\right] \tag{22}$$

$$A \odot B \;=\; \left\langle \frac{A \cdot \pi_1}{B \cdot \pi_2} \right\rangle \tag{23}$$

Fact:

$$A \oplus B \;=\; A \odot B \tag{24}$$

Calculation:

$$\begin{aligned}
A \oplus B &= \left[\; i_1 \cdot A \;\middle|\; i_2 \cdot B \;\right] \\
&= \left(\begin{array}{c|c} A & 0 \\ \hline 0 & B \end{array}\right) \\
&= i_1 \cdot A \cdot \pi_1 + i_2 \cdot B \cdot \pi_2 \\
&= \left\langle \frac{A \cdot \pi_1}{B \cdot \pi_2} \right\rangle \\
&= A \odot B
\end{aligned}$$

(Bi)functors

That $A \oplus B$ ($=A \odot B$) is a (bi)functor is immediate from the universal properties. A number of standard properties arise:

$$id \oplus id = id \tag{25}$$

$$(A \oplus B) \cdot (C \oplus D) = (A \cdot C \oplus B \cdot D) \tag{26}$$

$$\left[\, A \,\middle|\, B \,\right] \cdot (C \oplus D) = \left[\, A \cdot C \,\middle|\, B \cdot D \,\right] \tag{27}$$

$$(A \odot B) \cdot \frac{C}{D} = \frac{A \cdot C}{B \cdot D} \tag{28}$$

Polymorphic matrices

Matrices i_1, i_2, π_1, π_2 are polymorphic, as exhibited by "free theorems":

$$A \cdot i_1 = i_1 \cdot (A \oplus B) \tag{29}$$

$$A \cdot \pi_1 = \pi_1 \cdot (A \odot B) \tag{30}$$

and so on. But there are more examples of matrix polymorphism: id itself is polymorphic, cf. "free theorem":

$$A \cdot id = id \cdot A = A \tag{31}$$

Polymorphic matrices

Matrix id is a special case of a diagonal matrix: given scalar a, we define

$$a_n = \begin{bmatrix} a & 0 & \cdots & 0 \\ 0 & a & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a \end{bmatrix}_{n \times n} \qquad n \xleftarrow{a_n} n$$

Clearly, $id = 1$. Also note that a can be defined blockwise, since $a = a \oplus a$, for $n > 1$ and arbitrary choice of block sizes.

Polymorphic matrices

(Pointfree) scalar product: define

$$aA = a \cdot A \tag{32}$$

"Free theorem":

$$A \cdot a = a \cdot A \qquad (= aA) \tag{33}$$

Flipping

$$\mathit{flip}\ X \quad = \quad \mathit{swap} \cdot X \tag{34}$$

where

$$\mathit{swap}\ 1 = 1$$

$$\mathit{swap}\ (n+m) = \left(\begin{array}{c|c} 0 & \mathit{swap}\ m \\ \hline \mathit{swap}\ n & 0 \end{array}\right)$$

Fact:

$$\mathit{swap}\ (n+m) \cdot \mathit{swap}\ (m+n) \quad = \quad \mathit{id} \tag{35}$$

Thus:

$$\mathit{flip}(\mathit{flip}\ X) \quad = \quad X$$

Gaussian elimination

Motivation:

$$ge\left(\begin{array}{c|c} x & y \\ \hline z & k \end{array}\right) = \left(\begin{array}{c|c} x & y \\ \hline z - \frac{1}{x}zx & k - \frac{1}{x}zy \end{array}\right)$$

Generalization:

$$\begin{aligned} ge\left(\begin{array}{c|c} x & M \\ \hline N & Q \end{array}\right) &= \left[\begin{array}{c|c} x & M \\ \hline 0 & ge(Q - \frac{N}{x} \cdot M) \end{array}\right] \\ ge\ x &= x \end{aligned}$$

Types:

$$\begin{tikzcd} & 1 & \\ 1 \arrow[ur, "x"] \arrow[dr, "N"'] & & m \arrow[ul, "M"'] \arrow[dl, "Q"] \\ & n & \end{tikzcd} = \begin{tikzcd} 1 \arrow[d, "x"] \arrow[drr, "N"'] & 1+m & m \arrow[d, "Q"] \arrow[dll, "M"] \\ 1 & 1+n & n \end{tikzcd}$$

Another solution

Step of gaussian elimination?

$$\pi_1' \cdot \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \end{bmatrix}$$

$$\pi_2' \cdot \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \alpha a_{11} + a_{21} & \alpha a_{12} + a_{22} \end{bmatrix}$$

Building Elementary Matrices?

$$\left\langle \frac{\pi_1'}{\pi_2'} \right\rangle = \left\langle \frac{\begin{bmatrix} 1 & 0 \end{bmatrix}}{\begin{bmatrix} \alpha & 1 \end{bmatrix}} \right\rangle = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix}$$

Tensor/Kroneker product

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{bmatrix}$$

$$x \otimes A = xA$$

$$\frac{C}{D} \otimes A = \frac{C \otimes A}{D \otimes A}$$

$$\left[\ C \mid D\ \right] \otimes A = \ C \otimes A \,\big|\, D \otimes A$$

Striding

Kronecker notation:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \qquad \overline{A} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}$$

A kind of exponential?

Universal Law

$$K = \overline{A} \quad \equiv \quad A = ap \cdot (K \otimes id)$$

Indeed

J. Magnus book (1999)

$$
\begin{array}{lcl}
\mathbf{vec}\, ab^T & = & b \otimes a \\
\mathbf{vec}\, A & = & (A^T \otimes I_m) \times \mathbf{vec}\, I_m \\
\mathbf{vec}\, A & = & (I_n \otimes A) \times \mathbf{vec}\, I_n
\end{array}
$$

Comments?

Why Abstract Nonsense

> *"A category can be seen as a structure that formalizes a mathematician's description of a type of structure"*
>
> *Barr's et al*

Could We Pay Back Mathematics?

Franz Franchetti, Frédéric de Mesmay, Daniel McFarlin, and Markus Püschel.
Operator language: A program generation framework for fast kernels.
In *IFIP Working Conference on Domain Specific Languages (DSL WC)*, 2009.

S. MacLane.
Categories for the Working Mathematician.
Springer-Verlag, New-York, 1971.