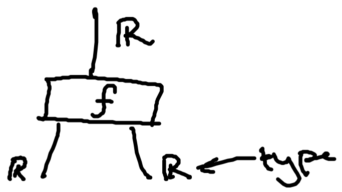
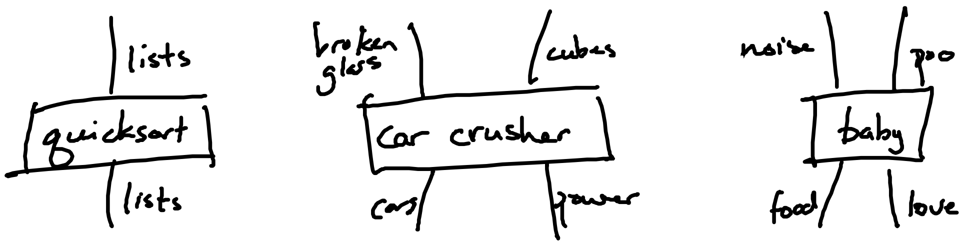
Lecture 1

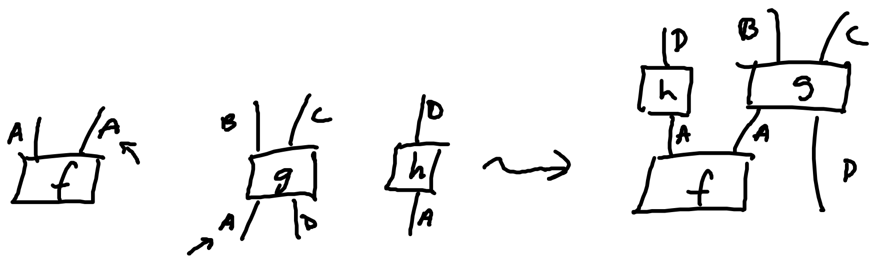
3.1 Processes

Def A process is anything with 0 or more inputs and outputs.

Ex $f(x,y) = x^2 + y$





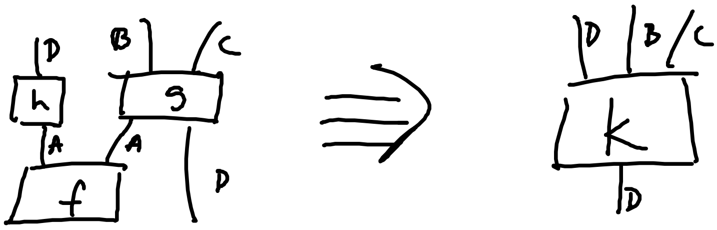


Def A process theory consists of:

(i) a collection T of system-types

(ii) a collection P of processes

→ (iii) a means of composing diagrams of processes.



EXAMPLES

- functions
  - types: Sets $A, B, C, \dots$
  - procs: $f: A \to B$ functions
- linear maps
  - types: vector spaces (or $\mathbb{N} \leftrightarrow$ dimension)
  - procs: linear maps (or matrices: $A: m \to n$ is an $n \times m$ dim'l matrix)

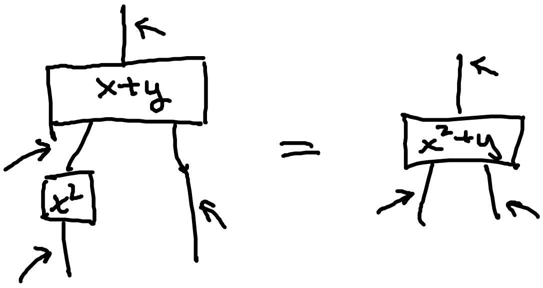
classical prob thy   quantum theory

The Golden Rule of Process Theories:

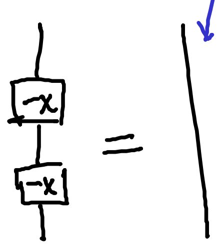
ONLY CONNECTIVITY MATTERS (ocm)

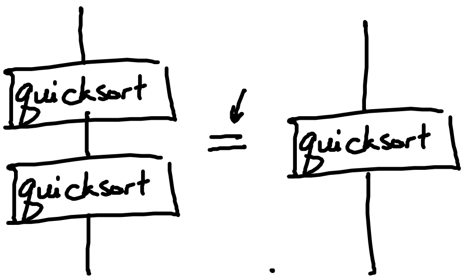
- These processes are equal because their diagrams are equal. However, we can also have multiple diagrams that describe the same process.

$f(x,y) = x+y$



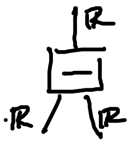
"do nothing"/identity process



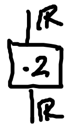


DIAGRAMMATIC REASONING := "using equations between diagrams to prove stuff"

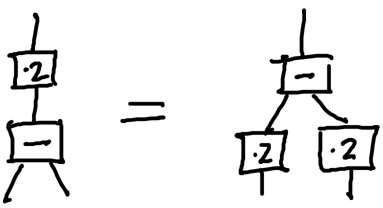
Ex



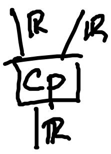
$:: (m,n) \mapsto m-n$



$:: m \mapsto 2m$

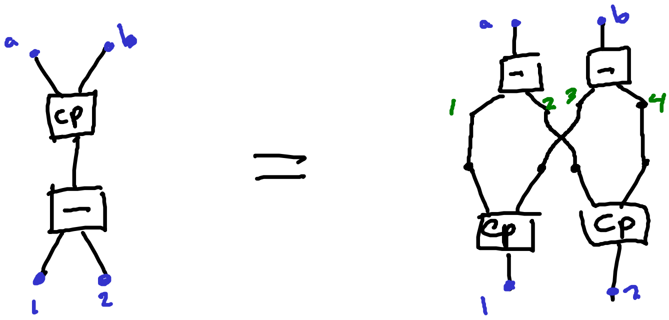


✓

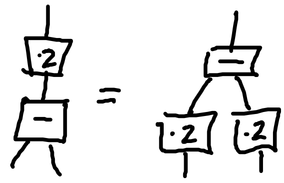


$:: n \mapsto (n,n)$

(1)



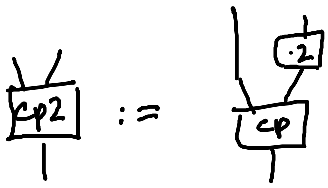
(2)



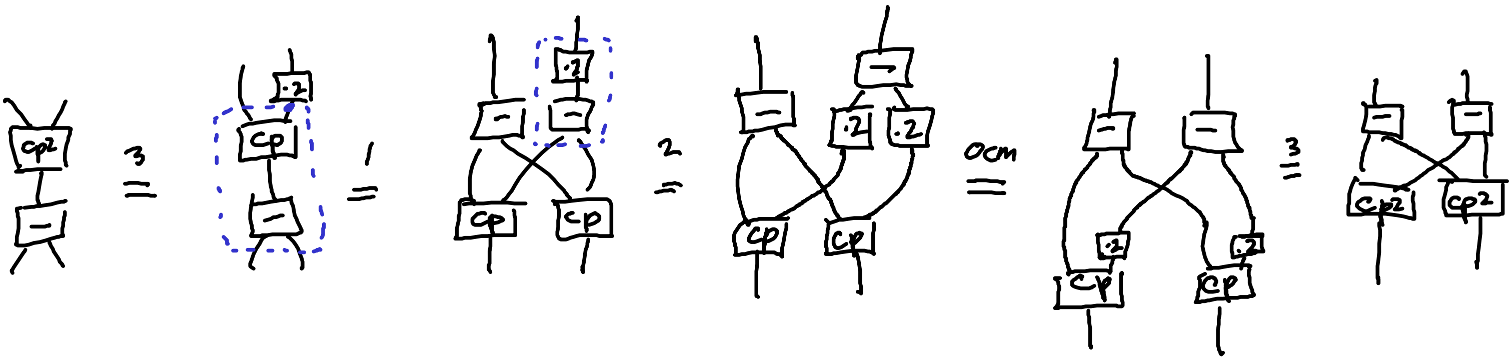
$(m,n) \mapsto m-n \mapsto (m-n, m-n)$

$(m,n) \mapsto (m,m,n,n) \mapsto (m-n, m-n)$

(3)



$n \mapsto (n, 2n)$



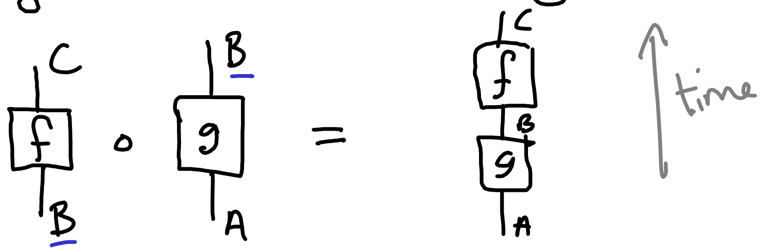
(4)

3.2 Circuit diagrams

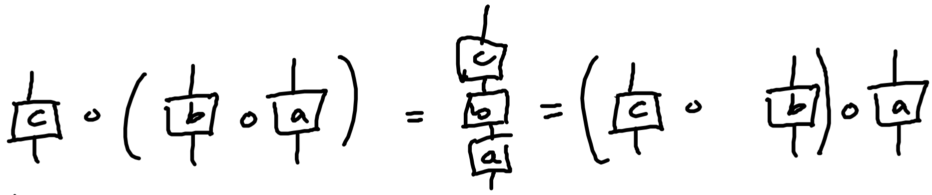
Sequential composition

time ←

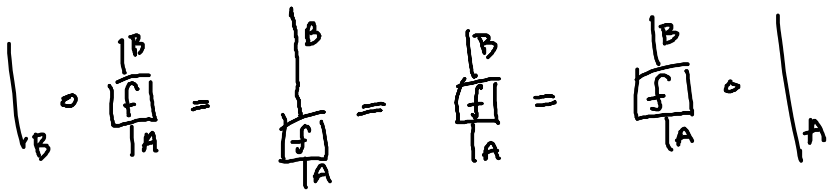
$f \circ g :=$ " f after g "



- associative



- has a unit :  ← "do nothing / identity" process



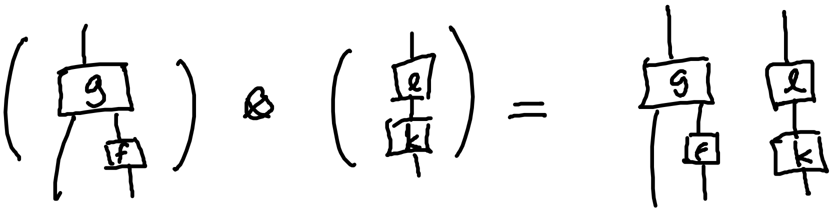
* For functions: $(f \circ g)(x) = f(g(x))$

function composition

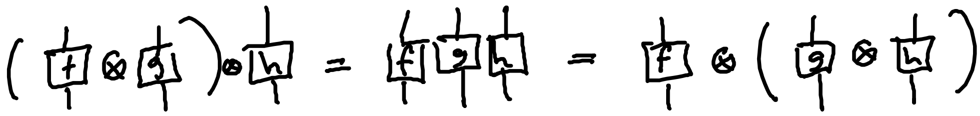
* For matrices: $A \circ B = AB$

matrix multiplication

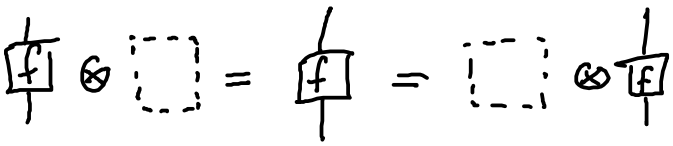
Parallel composition : $f \otimes g$ "f while g"



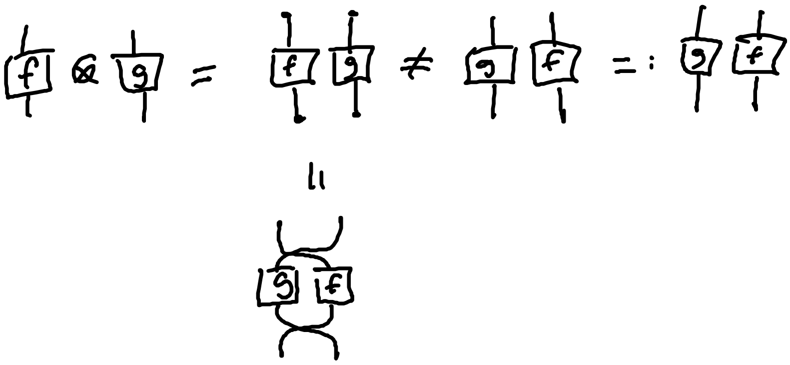
· associative



· unit

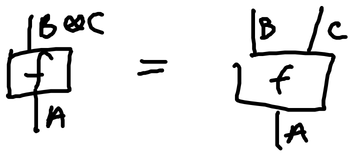


· The order matters.

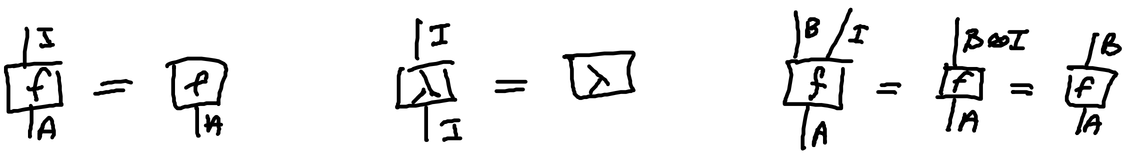


* We can also form joint systems.

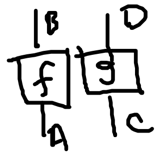
B, C system-types $\Rightarrow$ $B \otimes C$ is a system-type



* We write the trivial system as I. $A \otimes I = A = I \otimes A$



Lecture 2

Parallel composition: 

$f \otimes g : A \otimes C \to B \otimes D$

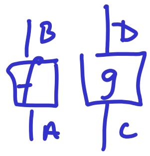
For functions, parallel composition is Cartesian product:

types (sets) : $A \otimes B = A \times B = \{(a,b) \mid a \in A, b \in B\}$

$I = \{*\}$ ← one-element set. (not $\{\}$!)

$(A \times B) \times C \cong A \times (B \times C) \cong \{(a,b,c) \mid a \in A, b \in B, c \in C\}$

$A \times \{*\} = \{(a,*) \mid a \in A\} \cong A \cong \{*\} \times A$

procs (functions): 

$f \times g : A \times C \to B \times D$

$(f \times g)(a,c) = (f(a), g(c))$

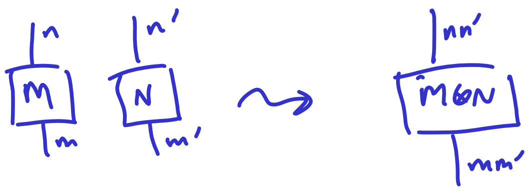
For linear maps, $\otimes$ is the tensor product $\otimes$.

Let $\mathbb{C}^n$ be the n-dimensional space $\left\{ \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \middle| \; v_1, \dots, v_n \in \mathbb{C} \right\}$

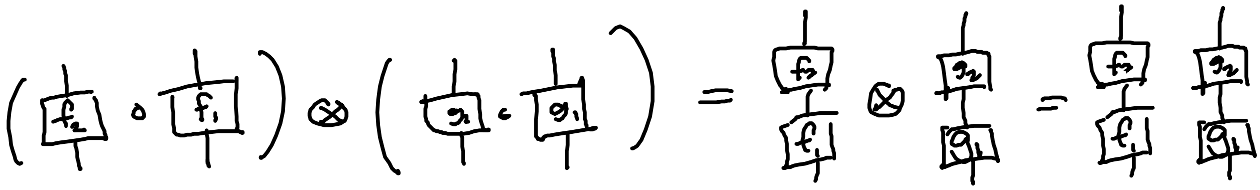
types (dimension): $\mathbb{C}^m \otimes \mathbb{C}^n := \mathbb{C}^{m \cdot n}$ $\quad I = \mathbb{C}^1 \cong \mathbb{C}$ (not 0!)

$$M \otimes N = \overbrace{\underbrace{\begin{pmatrix} a & b \\ c & d \end{pmatrix}}_{2\times 2}}^{M} \otimes \overbrace{\underbrace{\begin{pmatrix} e & f \\ g & h \end{pmatrix}}_{2\times 2}}^{N}$$

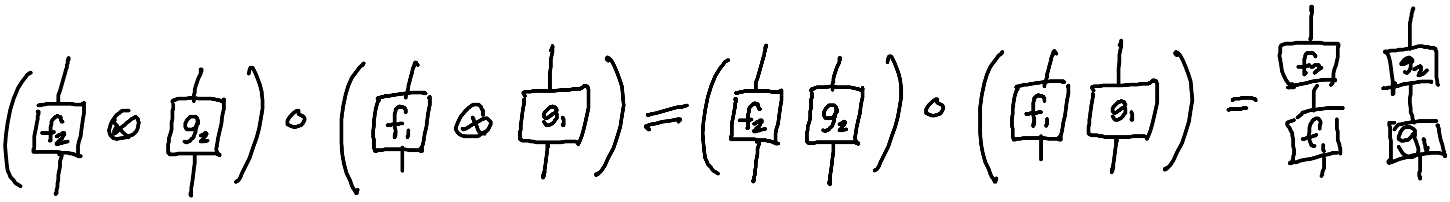
$$= \begin{pmatrix} aM & bM \\ cM & dM \end{pmatrix} = \underbrace{\begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix}}_{4\times 4}$$



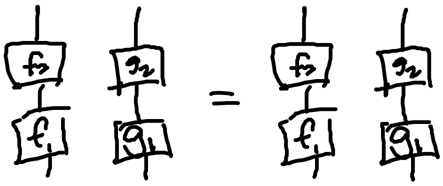
TOGETHER:



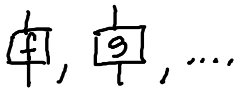
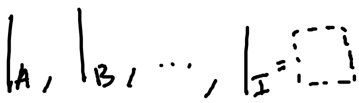
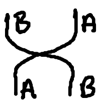
interchange law



Non-trivial equation in 1D: $(f_2 \circ f_1) \otimes (g_2 \circ g_1) = (f_2 \otimes g_2) \circ (f_1 \otimes g_1)$

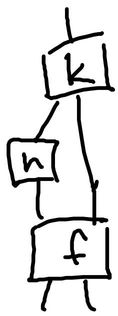
is free (ocm) in 2D: 

<u>DEF</u> A circuit diagram is any diagram built from

- boxes  , ...
- (identity) wires 
- <u>swap</u> processes 

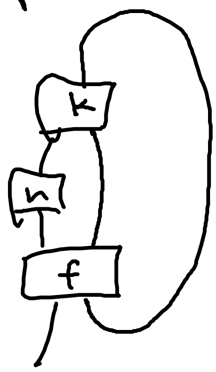
using <u>only</u> $\otimes$ and $\circ$.

Equivalently circuit diagrams are any diagram that doesn't have feedback loops.



✓

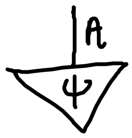
D.A.G.



×

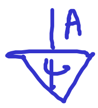
3.4 Special processes

States



$\psi : I \to A$ "preparation"

For functions states are the same a elements of a set.



$\psi : \{*\} \to A \qquad \psi(*) = a \in A.$

For linear maps: $\psi : \underbrace{\mathbb{C}}_{\dim = 1} \to \underbrace{\mathbb{C}^n}_{\dim = n}$ } $n \times 1$ matrix (column vector)

$$\psi = \begin{pmatrix} \psi^1 \\ \vdots \\ \psi^n \end{pmatrix}$$

For classical thy $\subseteq$ linear maps, states are prob distrs:

 $= \begin{pmatrix} 2/3 \\ 1/3 \\ 0 \end{pmatrix}$

$\text{Prob}(0) = 2/3$
$\text{Prob}(1) = 1/3$
$\text{Prob}(2) = 0.$

Effects

 $: A \to I$ "question" or a "test"

For functions, effects are.... trivial! $\underset{\substack{\uparrow \\ \text{only one}}}{\pi} : A \to \{*\}$

For linear maps, effects $\pi : \underbrace{\mathbb{C}^n}_{\dim = n} \to \underbrace{\mathbb{C}}_{\dim = 1}$, are $1 \times n$ matrices: row vectors!

In classical thy, these are row vectors w/ entries $0 \leq r \leq 1$.
(sometimes called "fuzzy predicates")

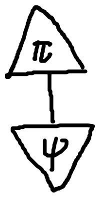
 $= (1 \;\; 0 \;\; 0)$ "in state $\emptyset$"

 $= (1 \;\; 1 \;\; 0)$ "in state $\emptyset$ or 1"

 $= \left(\frac{99}{100} \;\; \frac{1}{100} \;\; \frac{1}{100} \right)$ "got pretty reliable info that the state is 0"

Numbers: processes $\lambda : I \to I$

... for linear maps : $\lambda : \underbrace{\mathbb{C}}_{dim=1} \to \underbrace{\mathbb{C}}_{dim=1}$ 1x1 matrix (λ)

effect { state {  } number ⇜ prediction

.... in classical thy:

predicate { prob. dist. {  $= p$ ⇜ probability. "classical Born rule"

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

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

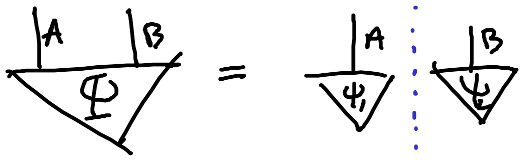
$$\left(\tfrac{99}{100}\ \tfrac{1}{100}\ \tfrac{1}{100}\right)\begin{pmatrix} 2/3 \\ 1/3 \\ 0 \end{pmatrix} = \frac{199}{300} \approx 2/3$$

Chapter 4: String diagrams

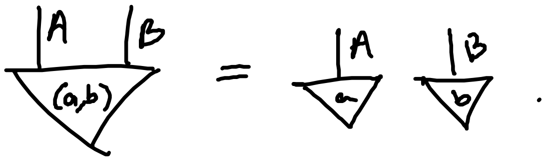
"separable vs. non-separable"

4.1

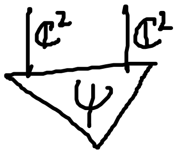
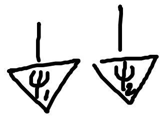
DEF a $\otimes$-separable state $\Psi: I \to A \otimes B$ is a state s.t. there exist $\psi_1: I \to A$, $\psi_2: I \to B$ where:



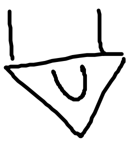
In functions, states are elements. $\Rightarrow$ all states are $\otimes$-sep'l.



In linear maps, some states are separable:

 $= \begin{pmatrix} 1 \\ 2 \\ -1 \\ -2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 2 \end{pmatrix} =$ 

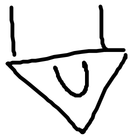
..., but some are not!

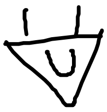
 $= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \neq \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix}$

$ac = 1 \Rightarrow a \neq 0, b \neq 0$ ... but $ad = 0$!

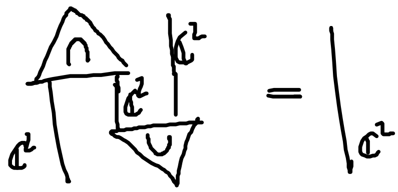
$bd = 1 \Rightarrow b \neq 0, d \neq 0$ $\quad\quad$ $bc = 0$!

Lecture 3

 $= \begin{pmatrix}1\\0\\0\\1\end{pmatrix} \neq \begin{pmatrix}ac\\ad\\bc\\bd\end{pmatrix} = \begin{pmatrix}a\\b\end{pmatrix}\otimes\begin{pmatrix}c\\d\end{pmatrix}$

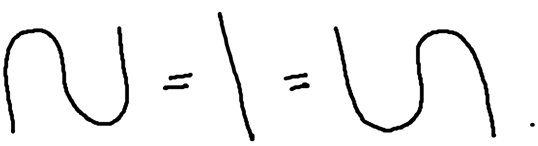
In fact,  is a special state, called the Bell state, or simply the cup.

Its transpose  $= (1\ 0\ 0\ 1)$ is called the cap.



New notation: 

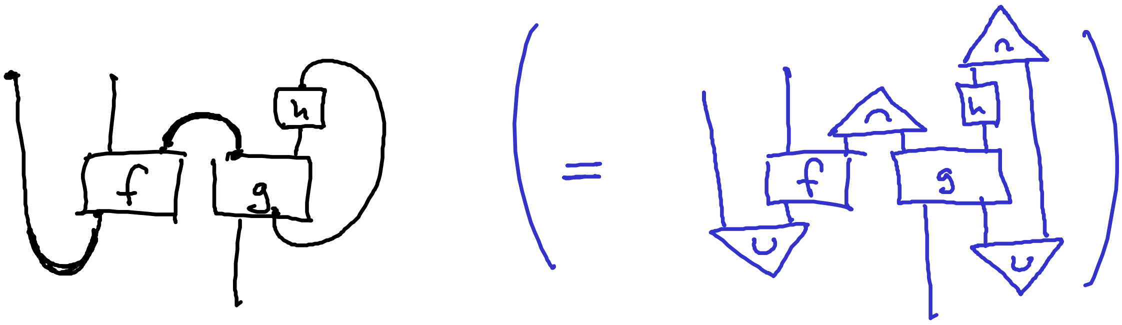
Def A process theory admits string diagrams if every system A has a special state  and effect  such that:

... and 

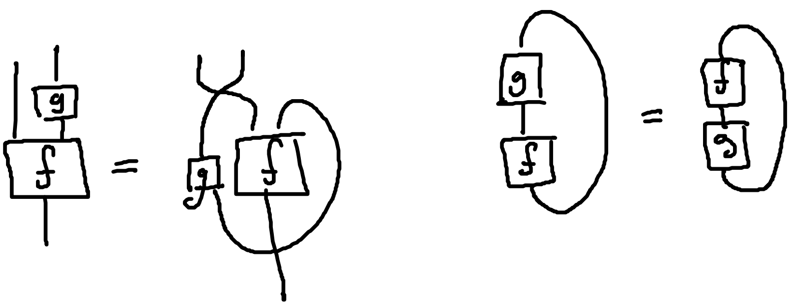
$(\cap\otimes 1_A)\circ(1_A\otimes \cup) = 1_A = (1_A\otimes\cap)\circ(\cup\otimes 1_A)$

Def A string diagram is a circuit diagram with cups & caps.

Equivalently, it is a diagram where loops and in-in/out-out connections are allowed.



Thm String diagrams satisfy OCM!

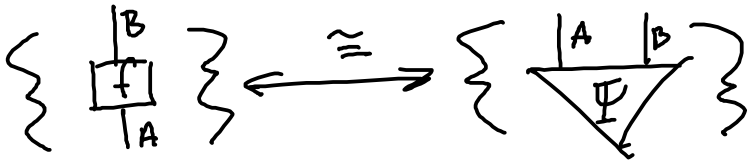


etc.

(Pf uses OCM for circuit diagrams plus cap/cup rules.)

n.b. functions does NOT admit string diagrams!
(but relations does, see Ex 4.16 in Dodo)

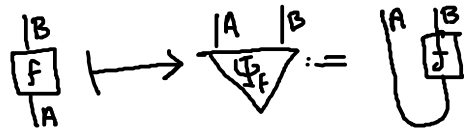
Process-State Duality



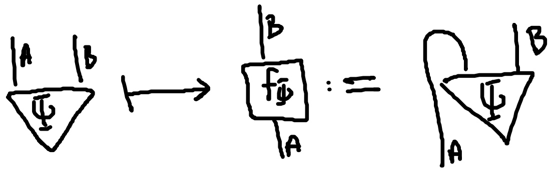
(in Q.T. Choi-Jamoitkowski isomorphism)

Thm In a process theory that admits string diagrams, the processes $f: A \to B$ are in 1-to-1 corresp. with states $\Psi: I \to A \otimes B$.

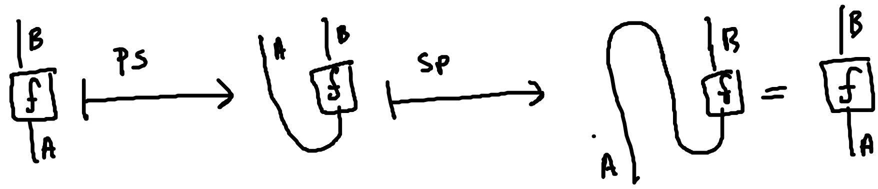
Pf Let:

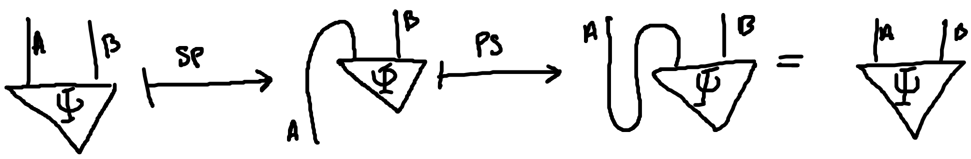


and:



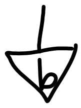
Then:



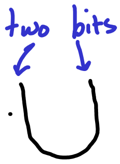


$PS = SP^{-1}$, so $\{f: A \to B\} \cong \{\Psi: I \to A \otimes B\}$. ∎

"Classical" teleportation.

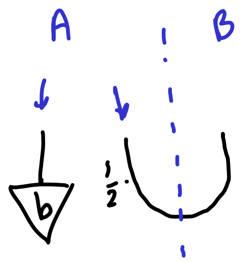
A has a bit  that she wants to send to B.

- They also share a random bit:

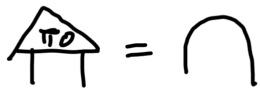
$\frac{1}{2}\cdot$ 

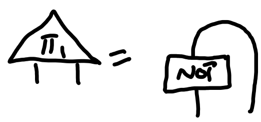
$= \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array}\begin{pmatrix} 1/2 \\ 0 \\ 0 \\ 1/2 \end{pmatrix}$ ← 50% chance 00 ; ← 50% chance 11

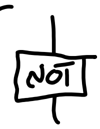
- A will observe her 2 bits:



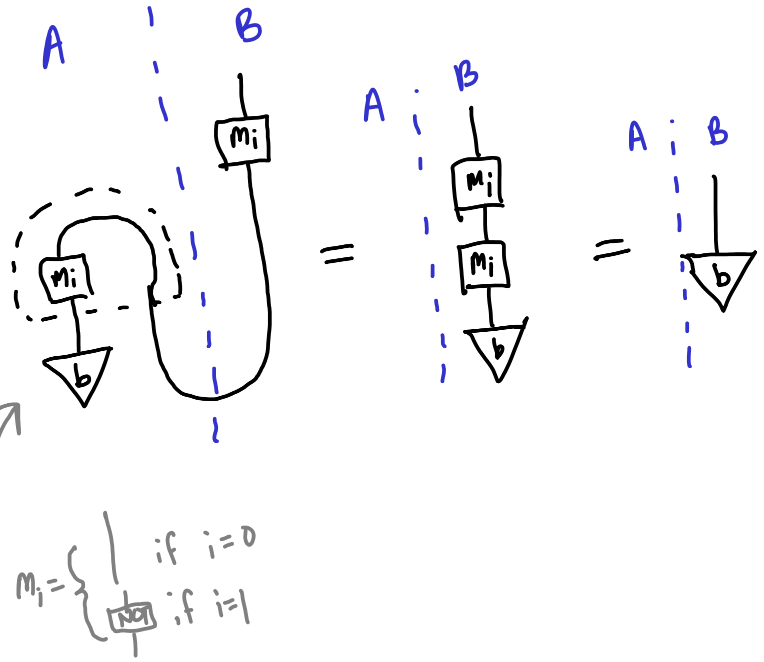
and make one of 2 obs:

(0) the bits are the same  $= (1\ 0\ 0\ 1)$

(1) the bits are different  $= (0\ 1\ 1\ 0)$

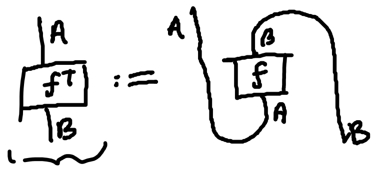
( $= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$)

- She tells B to flip his bit only if they are different.



This is known as one-time pad crypto.

4.2 Transpose & adjoint of a process

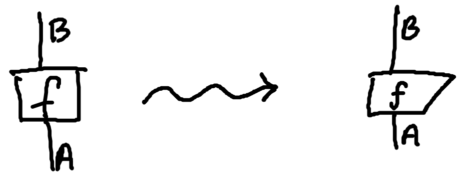
DEF (Transpose) 

Thm For linear maps, the matrix of  is the transpose of the matrix of .

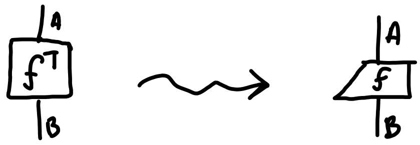
Pf (later).

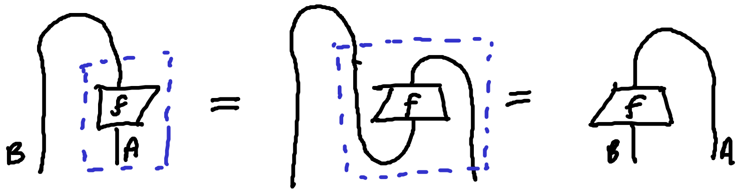
Alternate notation:

$f : A \to B$



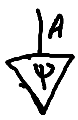
$f^T : B \to A$





4.3.1 ADJOINTS.

$\psi : I \to A$



"being in state ψ"

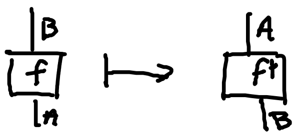
"process that prepares ψ"

$\longmapsto$

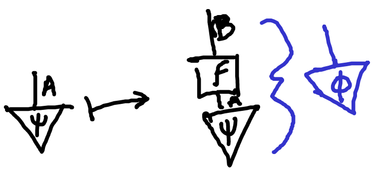
$\psi^\dagger : A \to I$

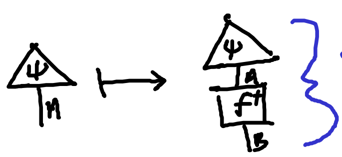
"testing for ψ"

EXTENDS TO PROCESSES:



$=: f^\dagger : B \to A$



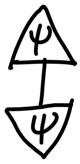


"testing for ϕ"

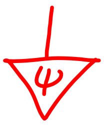
For linear maps, the adjoint is defined as the conjugate-transpose matrix.

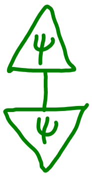
Q: why not the transpose?

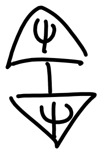
A: if we test a non-zero state ψ for itself, the result should be > 0. (positive-definiteness)

 > 0.

• The transpose does not have this property, e.g.

 $= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix} \Rightarrow$  $= \frac{1}{2}(1 \;\; i)\begin{pmatrix} 1 \\ i \end{pmatrix} = 1 + i^2 = 0.$ ☹

whereas:  $= \psi^{\dagger} \circ \psi = \frac{1}{2}(1 \;\; -i)\begin{pmatrix} 1 \\ i \end{pmatrix} = \frac{1}{2}(1 - i^2) = 1.$ ☺

Def A state is called normalised if  $= 1$.

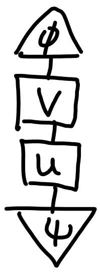
– Generally we will assume states are normalised (we'll see why next week.)

New (old) notation: $\psi : I \to A$  $|\psi\rangle$ ← ket

$\psi^{\dagger} : A \to I$  $\langle\psi|$ ← bra

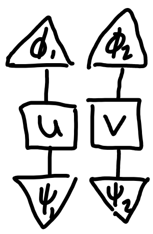
$\phi^{\dagger} \circ \psi$  $\langle\phi|\psi\rangle$ ← bra-ket (inner product

Bra-ket (a.k.a "Dirac") notation is used extensively in QC literature. It is just graphical notation sideways:



$\leftrightsquigarrow$ $\langle \phi | V U | \psi \rangle$

shorthand for $\otimes$ of states/effects



$\leftrightsquigarrow$

$$\langle \phi_1, \phi_2 | (U \otimes V) | \psi_1, \psi_2 \rangle$$

$$=$$

$$(\langle \phi_1 | \otimes \langle \phi_2 |)(U \otimes V)(|\psi_1\rangle \otimes |\psi_2\rangle)$$

interchange law

$$=$$

$$\langle \phi_1 | U | \psi_1 \rangle \langle \phi_2 | V | \psi_2 \rangle$$

Ch 5. Hilbert spaces,

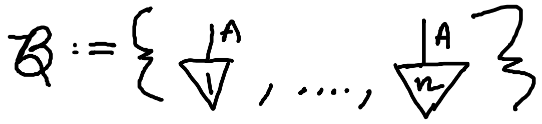
... or doing linear algebra with pictures.

Q: How much do we need to add to diagrams to calculate everything we can w/ matrices?

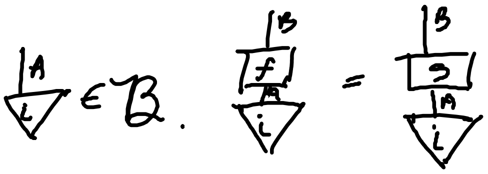
A: 2 extra things.

1. (Orthonormal) bases

DEF A basis for a type A is a minimal set of states:



such that:

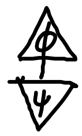
if for all 

then $f = g$.

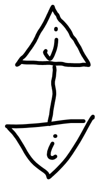
INTUITION basis states are "reference points" for a process

DEF The dimension of a system A is the size of any basis for A.

THM (Dimension theorem) For linear maps, all bases are the same size.

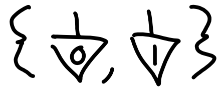
DEF Two states are <u>orthogonal</u> if  $= 0$.

DEF An orthonormal basis $^{(ONB)}$ $\mathcal{B} = \{$  $\}_{i=1\dots n}$ is a basis

where  $= \underbrace{\begin{cases} 1 & i=j \\ 0 & i\neq j \end{cases}}_{\delta_i^j}$ $\Leftarrow$ Kronecker δ.

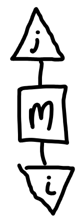
(In bra-ket:

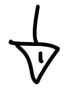
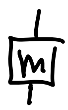
$\mathcal{B} = \{|1\rangle, |2\rangle, \dots, |n\rangle\}$ $\quad \langle j|i\rangle = \delta_i^j$)

For $\mathbb{C}^2$, we normally write the standard basis  $= \{|0\rangle, |1\rangle\}$

(a.k.a computational basis)

Using ONB's, we can recover the matrix of a process:

$m_i^j :=$  number in the j-th row and i-th column.

e.g in 2D:  $= \begin{pmatrix}1\\0\end{pmatrix}$  $= \begin{pmatrix}0\\1\end{pmatrix}$  $= \begin{pmatrix}a & b\\ c & d\end{pmatrix}$

 $= (0 \;\; 1)\begin{pmatrix}a & b\\ c & d\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = (0 \;\; 1)\begin{pmatrix}a\\c\end{pmatrix} = c.$

2. Sums (of diagrams)

<u>DEF</u> A process theory <u>has sums</u> if, for any two processes of the same type

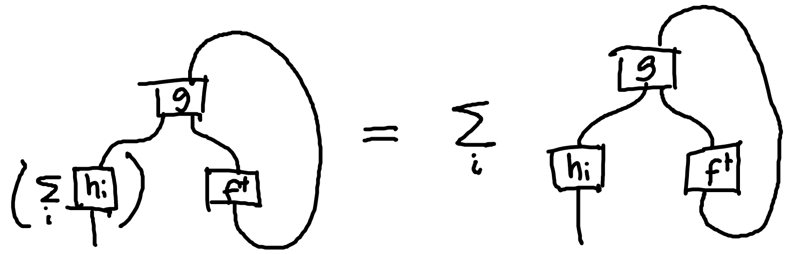
$$f : A \to B,\quad g : A \to B \quad \rightsquigarrow \quad f+g : A \to B$$

1. associative, commutative, and has unit 0.

   $f+(g+h) = f+(g+h)$ $\qquad$ $f+g = g+f$ $\qquad$ $f : A \to B,\ 0 : A \to B \rightsquigarrow f+0 = f.$

   Let $\sum_i f_i := f_1 + \dots + f_N$ $\quad (f_i : A \to B)$

2. Sums <u>distribute</u> over diagrams

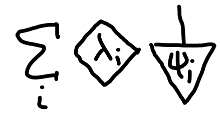


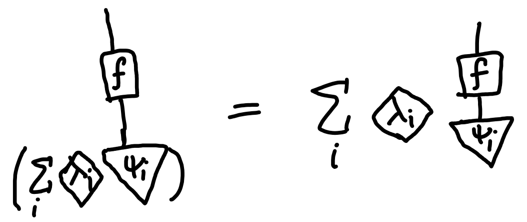
3. Sums <u>preserve</u> adjoints

$$\left(\sum_i f_i : A \to B\right)^\dagger = \sum_i f_i^\dagger : B \to A$$

Examples of sums distr. over diagrams.

(i) linearity.

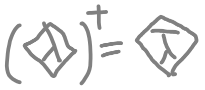
Linear combinations: $\sum_i \lambda_i |\psi_i\rangle =$ 

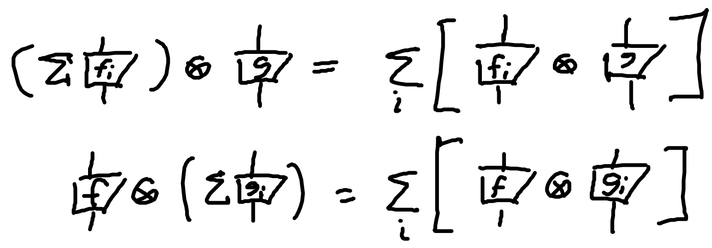
$f(\sum_i \lambda_i |\psi_i\rangle) =$  $= \sum_i \lambda_i f(|\psi_i\rangle)$

(ii) linearity of $\langle - | - \rangle$ $\quad$ $\langle \omega | v_1 + v_2 \rangle = \langle \omega | v_1 \rangle + \langle \omega | v_2 \rangle$

$\langle \omega | \sum_i v_i \rangle = \sum_i \langle \omega | v_i \rangle$

(iii) conjugate-linearity of $\langle - | - \rangle$ $\quad$ $\langle \sum_i v_i | \omega \rangle = \sum_i \langle v_i | \omega \rangle$

 $\Longrightarrow \langle \sum_i \lambda_i v_i | \omega \rangle = \sum_i \bar{\lambda}_i \langle v_i | \omega \rangle$

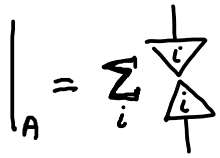
(iv) bilinearity of $\otimes$. 

Def A (finite-dimensional) Hilbert space is a vector space with an operation $\langle - | - \rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{C}$ that is:

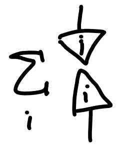
* linear in the 2nd arg.
* conj-linear in the 1st arg.
* conj-symmetric $\overline{\langle \psi | \phi \rangle} = \langle \phi | \psi \rangle$
* positive semi-def. $\langle \psi | \psi \rangle \geq 0$ and $\langle \psi | \psi \rangle = 0 \Rightarrow \psi = 0$.

Thm The types of linear maps are Hilbert

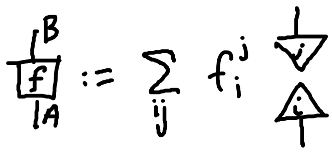
5.2 Matrix calculations with diagrams.

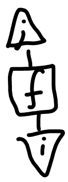
THM 

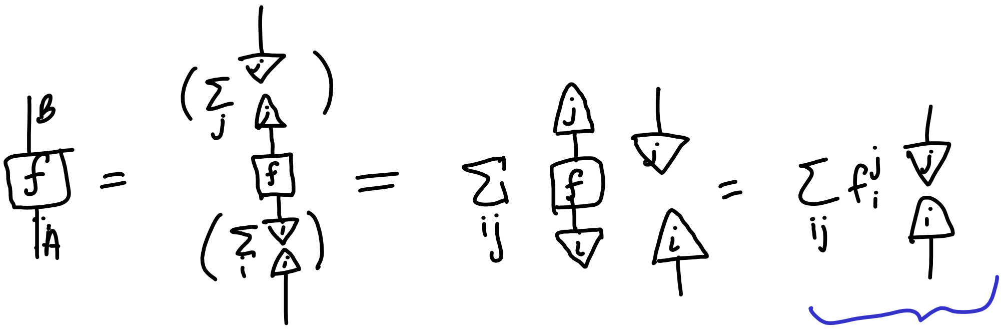
Pf The matrix of  is $\begin{pmatrix}0\\ \vdots\\ 0\\ 1\\ 0\\ \vdots\\ 0\end{pmatrix}(0..010..0) = \begin{array}{c} i^{th}\text{ col} \\ \downarrow \\ \begin{pmatrix} 0 & & 0\\ & 0\,1\,0 & \\ 0 & & 0\end{pmatrix} \end{array} \leftarrow i^{th}\text{ row}$

$\Rightarrow$  $= \begin{pmatrix}1 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots 1\end{pmatrix} = \;|\;.$ ∎

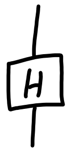
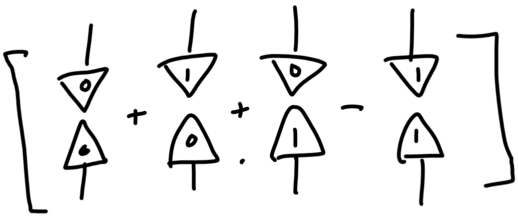
Cor For any linear map f, we have:



where $f_i^j =$  are the matrix entries.

Pf  ∎

"matrix form" of f.

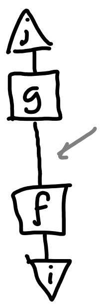
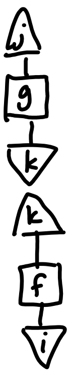
e.g.  $= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}$  $= (*)$

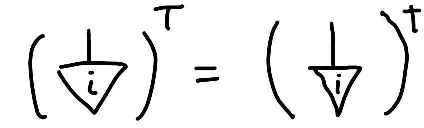
n.b. $H_i^j = \frac{1}{\sqrt{2}} \cdot (-1)^{ij}$, so $(*) = \frac{1}{\sqrt{2}} \sum_{ij} (-1)^{ij}$  .

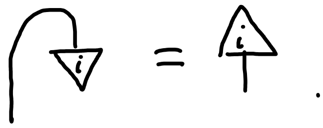
In bra-ket : $f = \sum_{ij} f_i^j \underbrace{|j\rangle\langle i|}_{\text{ket-bra}}$

Calculations involving ket-bra's are very common. These are essentially expanding a map in terms of it's matrix.

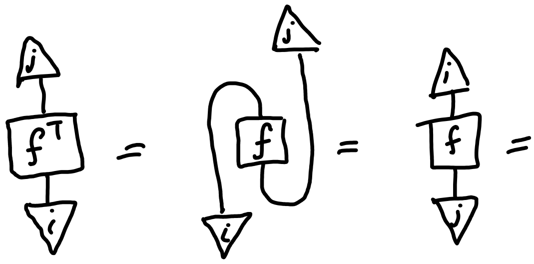
<u>Other</u> <u>consequences</u>:

$\underbrace{(g \circ f)_i^j}_{\text{matrix of } g\circ f} =$  $= \sum_k$  $= \underbrace{\sum_k g_k^j f_i^k}_{\text{product of matrix of } g \text{ w/ matrix of } f.}$

For the standard basis:  , i.e.



Consequence:

$(f^T)^j_i$ =  = f^i_j

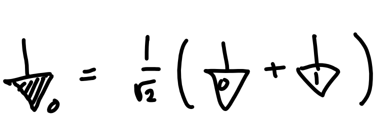
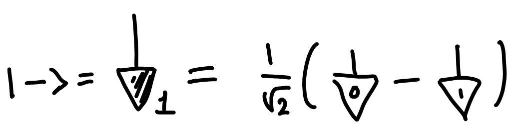
$f^j_i \underset{\text{transpose}}{\rightsquigarrow} f^i_j$

Other bases:

- Standard basis, a.k.a. Computational basis or Z-basis.

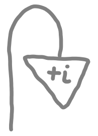
$|0\rangle$ =  = $\begin{pmatrix}1\\0\end{pmatrix}$ $\qquad$ $|1\rangle$ =  = $\begin{pmatrix}0\\1\end{pmatrix}$

- "Plus" basis, or X-basis

$|+\rangle$ =  $\qquad$ $|-\rangle$ = 

- Y-basis:

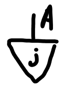
$|+i\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix} \qquad |-i\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$

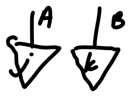
(Careful!  $= \frac{1}{\sqrt{2}} \cdot (1 \quad i) =$ )

adjoint

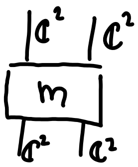
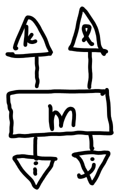
Product bases:

dim $A = m$ dim $B = n$

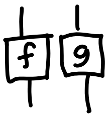
- If $\Big\{$  $\Big\}_{j=1}^{m}$ and $\Big\{$  $\Big\}_{k=1}^{n}$ are ONB's then

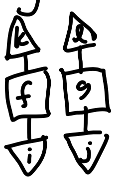
$\Big\{$  $\Big|\ j=1\ldots m,\ k=1\ldots n \Big\}$ is an ONB for $A \otimes B$.

$\dim(A \otimes B) = m \cdot n$.

 $\leftarrow$ 4×4 matrix with entries $M_{ij}^{kl} :=$ 

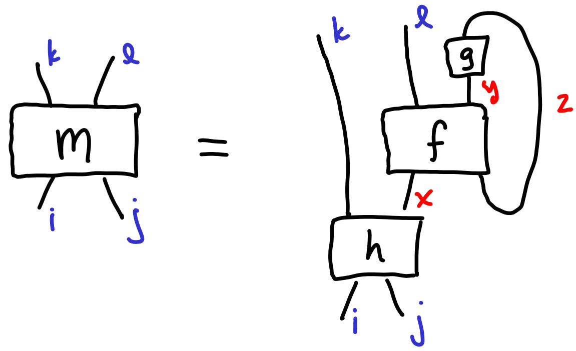
$$M = \begin{pmatrix} M_{00}^{00} & M_{01}^{00} & M_{10}^{00} & M_{11}^{00} \\ M_{00}^{01} & M_{01}^{01} & M_{10}^{01} & M_{11}^{01} \\ M_{00}^{10} & M_{01}^{10} & M_{10}^{10} & M_{11}^{10} \\ M_{00}^{11} & M_{01}^{11} & M_{10}^{11} & M_{11}^{11} \end{pmatrix}$$

__Thm__ The matrix of  is the Kronecker product of the matrices of f & g.

__Pf__ $(f \otimes g)_{ij}^{kl} :=$  $= \underbrace{f_i^k \cdot g_j^l}$ $\leftarrow$ Kron. product ■

Tensor contraction

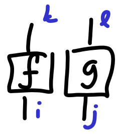
There is a recipe for computing the matrix of any string diagram in a single calculation.



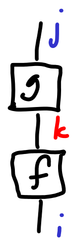
1. label every wire with a unique index
2. multiply the matrix entries of every box
3. sum over the connected wires.

$$m^{kl}_{ij} = \sum_{xyz} f^{ly}_{x}\, g^{z}_{y}\, h^{kx}_{ij}$$

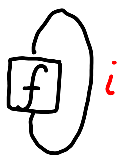
Special cases:



$\rightsquigarrow f^{k}_{i}\, g^{l}_{j}$



$\rightsquigarrow \sum_{k} f^{k}_{i}\, g^{j}_{k}$



$\rightsquigarrow \sum_{i} f^{i}_{i}$