# Y you should ZX

## An Introduction to ZX Calculus

Mahathi Vempati

# This talk

- What is ZX Calculus?
- What is diagrammatic reasoning?
- Process Theories
- Artefacts of Process Theories
- Correspondence between Linear Algebraic structures and *linear maps* process theory
- Proofs
- Quantum Information Protocol in ZX Calculus

# What is ZX Calculus?

- ZX Calculus is a rigorous graphical language for reasoning about linear maps between qubits.
- It was first introduced by Bob Coecke and Ross Duncan in 2008.
- It has applications in quantum circuit optimization (eg: PyZX) and quantum error correction (eg: surface codes).
- It is based on the principle of diagrammatic reasoning.

## Diagrammatic Reasoning

- Consider a set $S$ with 4 elements
- $a, b, c, d$ are representations of elements in $S$
- $S$ is closed under the binary operation MULTIPLY
- If representations of inputs to MULTIPLY are $p$ and $q$, then a representation of the output is $p * q$

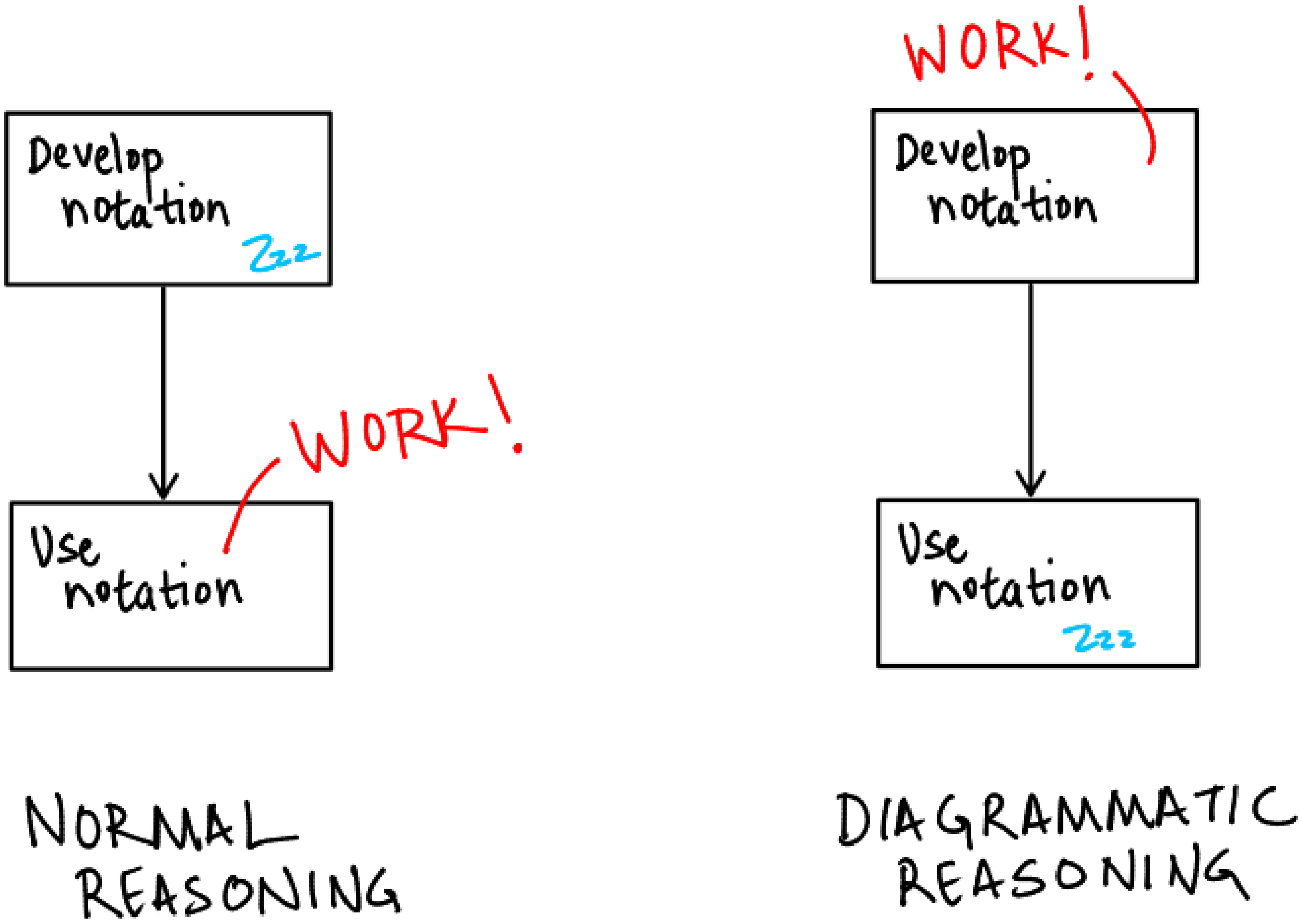
- MULTIPLY $(a, b) = a * b$
- MULTIPLY $(a * b, c) = a * b * c$
- MULTIPLY $(b, c) = b * c$
- MULTIPLY $(a, b * c) = a * b * c$

- MULTIPLY $(a, b)$ = $a * b$
- MULTIPLY $(a * b, c)$ = $a * b * c$
- MULTIPLY $(b, c)$ = $b * c$
- MULTIPLY $(a, b * c)$ = $a * b * c$

Just by the representation, you infer that MULTIPLY is an associative operation!

In diagrammatic reasoning, the axioms are precise statements about the representation of mathematical objects.

The mathematical properties can be derived from the axioms, if required.



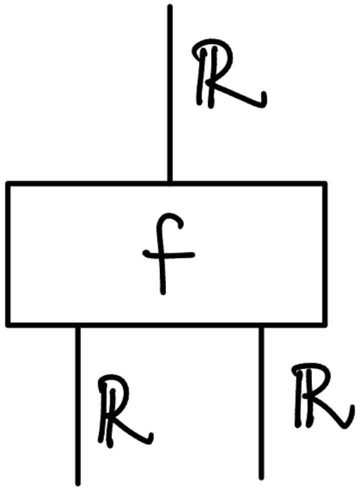
Develop notation
Zzz
Use notation
WORK!
NORMAL REASONING
WORK!
Develop notation
Use notation
Zzz
DIAGRAMMATIC REASONING

## Processes

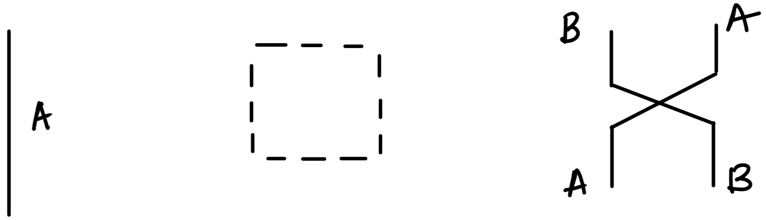
- *Anything* with zero or more inputs and zero or more outputs is known as a process. Example: functions, relations
- The inputs and outputs have *system types* associated with them.
- A process is represented by a named box, with input wires at the bottom, and output wires at its top, marked with types.

## Example

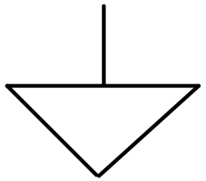
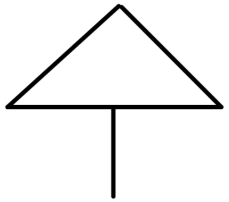
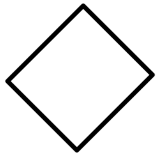
A function $f : \mathbb{R}^2 \mapsto \mathbb{R} :: (x, y) \mapsto x^2 + y$




## Identity, empty process and swap

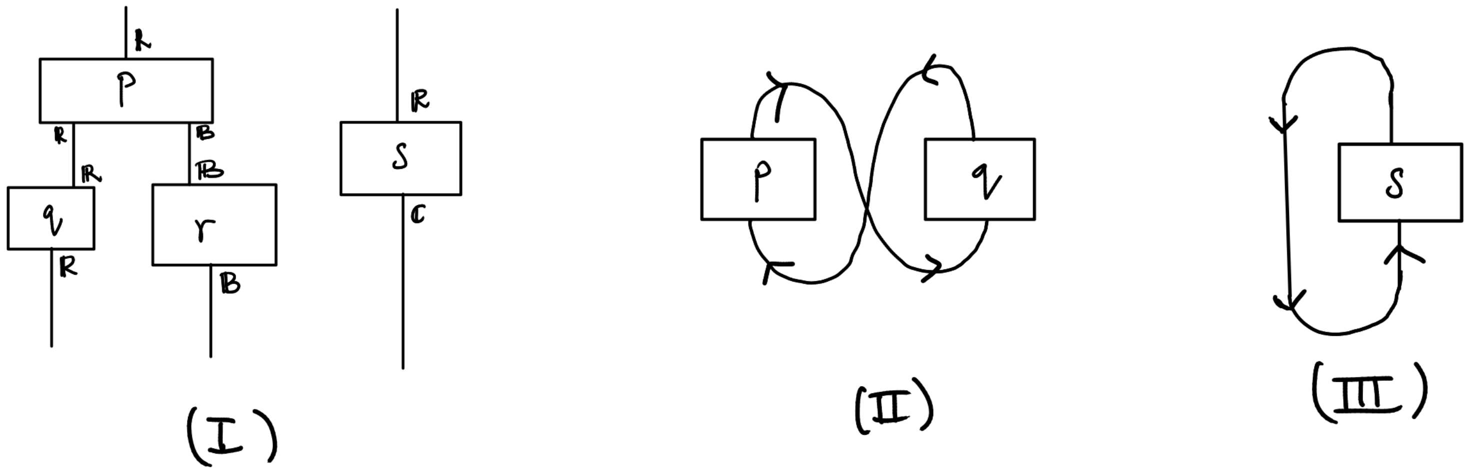


## Special processes

| Name | Process | Representation |
|---|---|---|
| State | A process with no inputs |  |
| Effect | A process with no outputs |  |
| Number | A process with no inputs and no outputs |  |

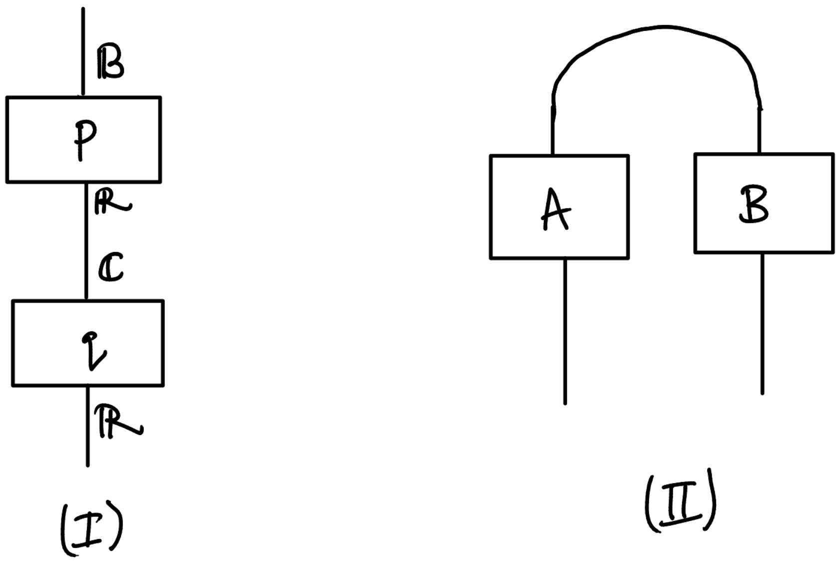
# Diagrams

A collection of processes that may have some outputs connected to inputs of the same type is called a diagram. **Examples:**




(I): All processes need not be connected, and note that only outputs and inputs of same types are connected.
(II) and (III): Types have been omitted, arrows drawn for clarity. Note that diagrams can have *directed cycles* of wires.

# Diagrams: non-examples



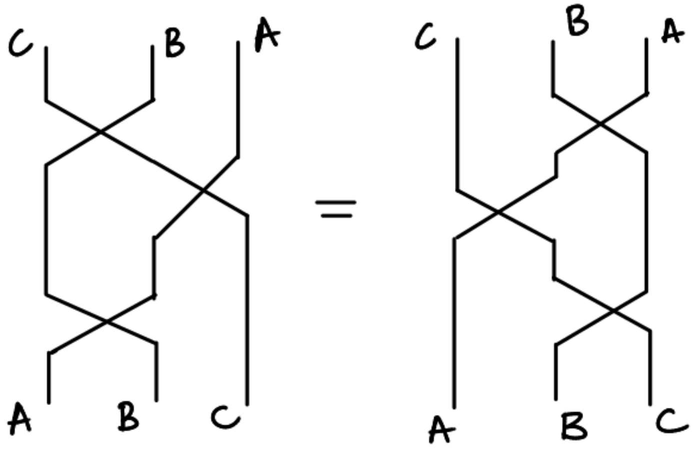

(I): Types don't match, and (II) output connects to output

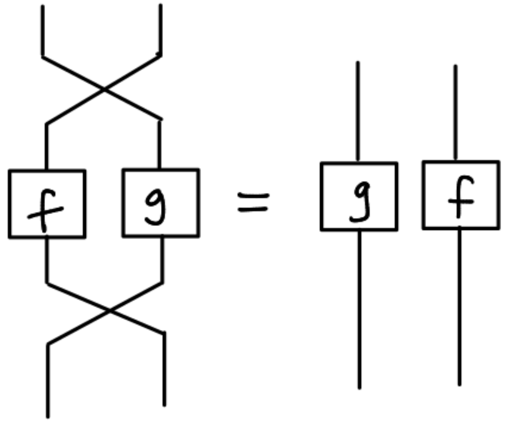
## The diagram description axiom:

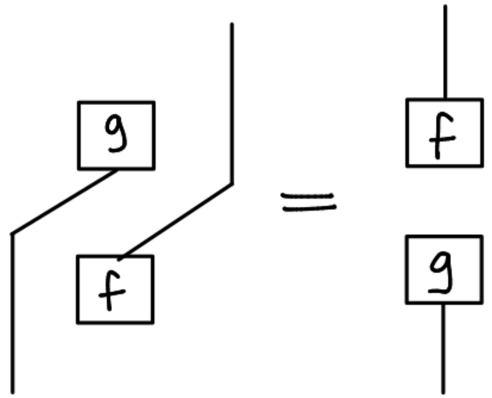
A complete description of a diagram consists of **what boxes it contains** and **how those boxes are connected**.

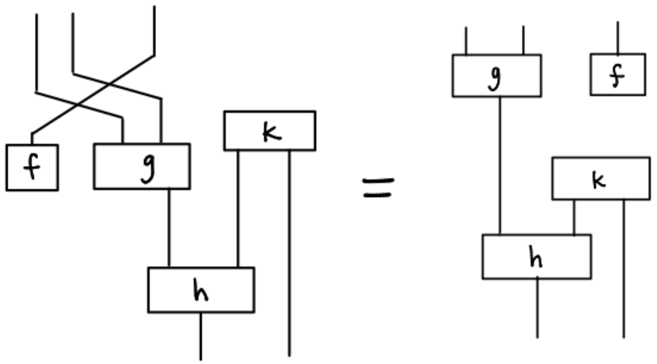
## Corollary (of the diagram description axiom)

If two diagrams can be deformed into each other without changing connections, then they are equal.



## Circuit Diagrams

- Circuit diagrams are diagrams that do **not** contain directed cycles of wires.
- An equivalent definition of a circuit diagram is:
  A diagram is a circuit if it can be constructed by composing boxes, including identities and swaps, by means of *parallel composition* and *sequential composition*.

# Process Theory

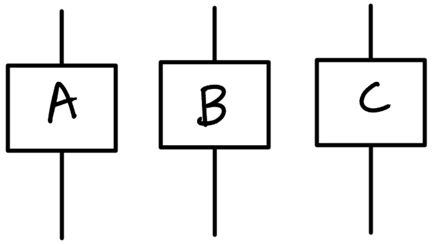
Usually, one is not interested in all possible processes, but rather, a certain class of related processes.

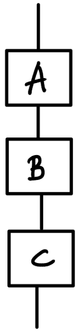
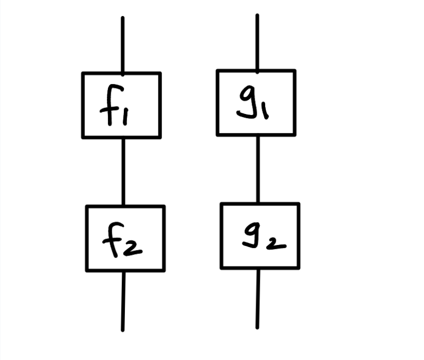
A process theory consists of:

1. A collection $T$ of system types represented by wires.
2. A collection $P$ of processes represented by boxes. For each process in $P$, the input types and the output types are taken from $T$.
    - Identities and swaps for all *system types*, and the empty process always belongs to $P$
3. All diagrams of processes in $P$ also belong to $P$.

## What properties are implicit for process theories?

- Symbolically, the same definition for a process theory defined over circuit diagrams can be expressed using a mathematical object called a *strict symmetric monoidal category.*
- The definition includes
    - an operator for parallel composition of processes $(\otimes)$
    - an operator for sequential composition of processes $(\circ)$

| Diagram | Property | Symbolic expression |
|---|---|---|
|  | $\otimes$ is associative and unital on processes | $A \otimes (B \otimes C) = (A \otimes B) \otimes C$<br>$A \otimes I = A = I \otimes A$ |

| Diagram | Property | Symbolic expression |
|---|---|---|
|  | $\circ$ is associative and unital on processes | $A \circ (B \circ C) = (A \circ B) \circ C$<br>$A \circ I_x = A = I_x \circ A$ |
|  | $\circ$ and $\otimes$ satisfy the interchange law | $(f_1 \otimes g_1) \circ (f_2 \otimes g_2)$<br>$= (f_1 \circ f_2) \otimes (g_1 \circ g_2)$ |

## Example of a process theory: *functions*

1. The collection $T$ consists of all sets.
2. The collection $P$ consists of all functions.
3. For $t_1, t_2 \in T$, $t_1 \otimes t_2 = t_1 \times t_2$ (Cartesian Product)
4. Let $p_1, p_2 \in P$ such that $p_1 : t_{1in} \to t_{1out}$ and $p_2 : t_{2in} \to t_{2out}$

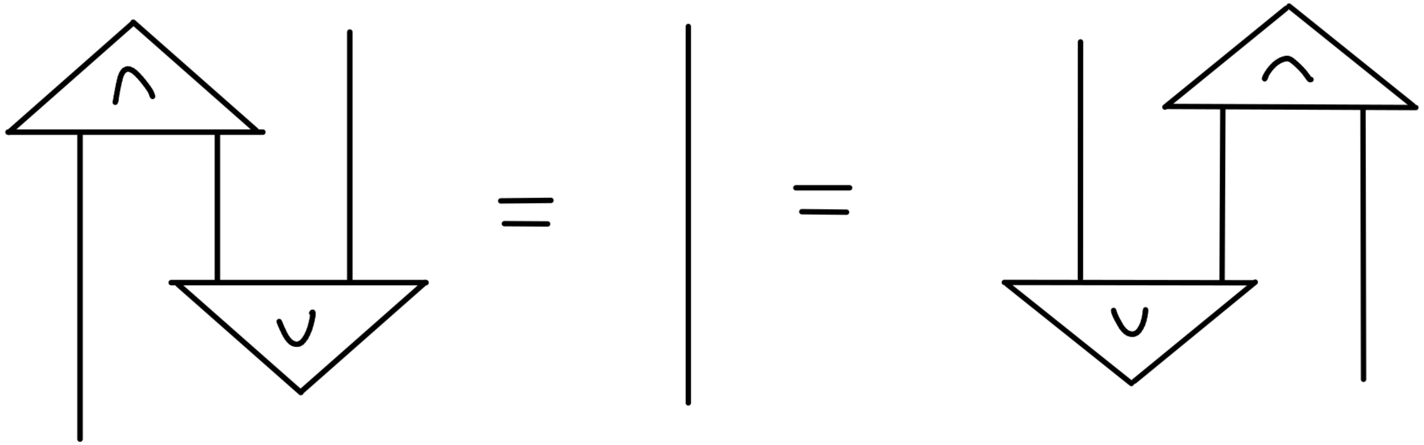
$$p_1 \otimes p_2 : t_{1in} \otimes t_{2in} \to t_{1out} \otimes t_{2out}$$
$$(x, y) \to (p_1(x), p_2(y))$$

5. If the types match, then we have: $p_2 \circ p_1(x) = p_2(p_1(x))$

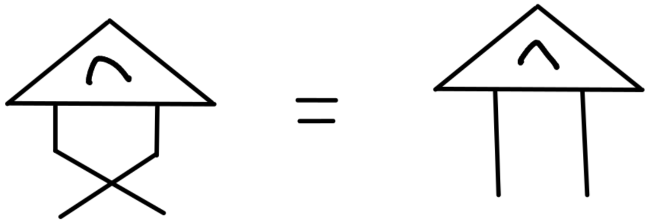
# Artefacts in a process theory

If a process theory defined on circuits possesses a state and an effect that satisfy the *Yanking equations* for every system type:

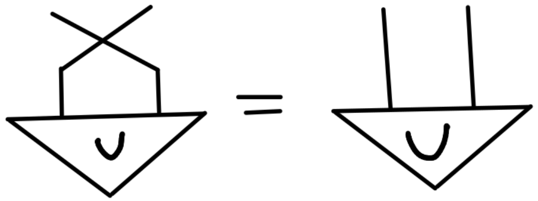
(I)



(II)

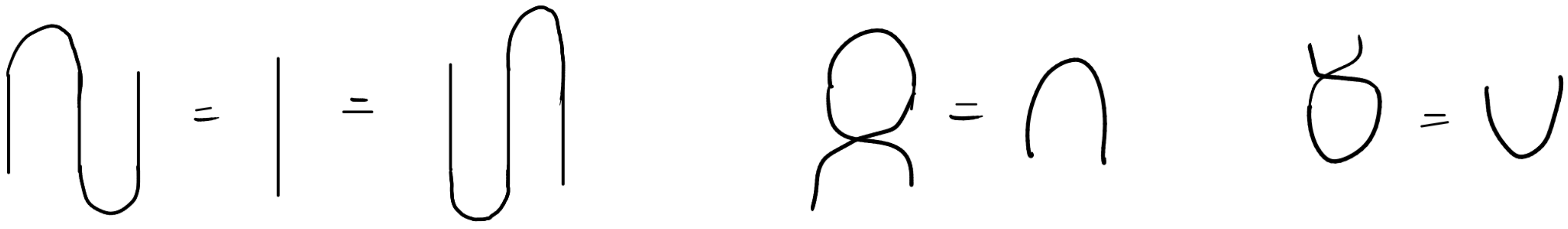


(III)



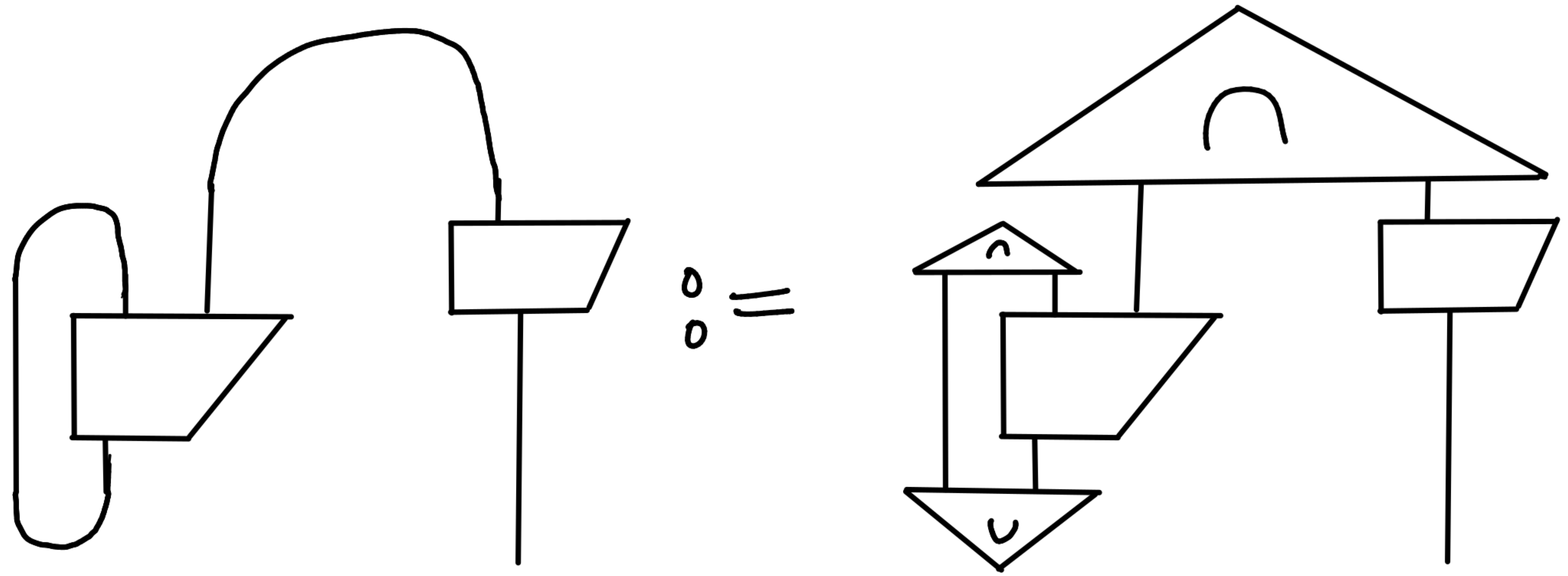
Then,

1. The state is known as the cup state and the effect is known as the cap effect.
2. The cups and caps can be replaced with wires.
3. The process theory is said to admit string diagrams.
4. The process theory has a process-state duality



The Yanking Equations

- String diagrams are diagrams in which an input can be connected to an input and an output can be connected to an output.

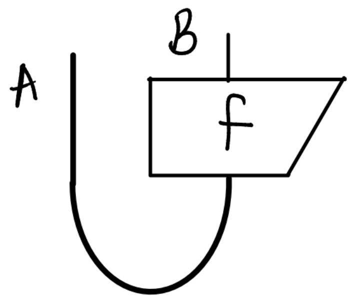


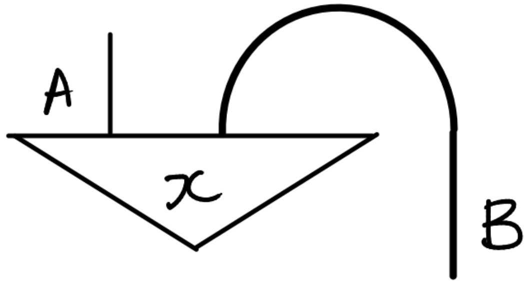
- We will also use deformed boxes to represent processes from now on, and exploit this breaking of symmetry.

## Process-state duality

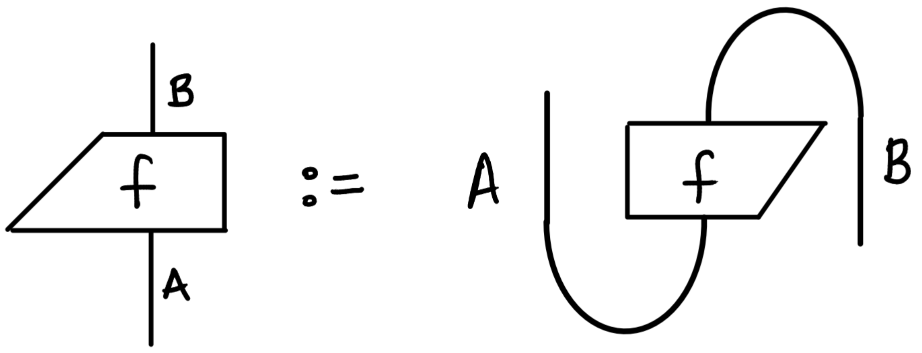
If a process theory admits string diagrams, there exists a correspondence between bipartite states and processes, as the cup and cap induce an invertible map between them.

Below is the state corresponding to process $f$, and process corresponding to state $x$.

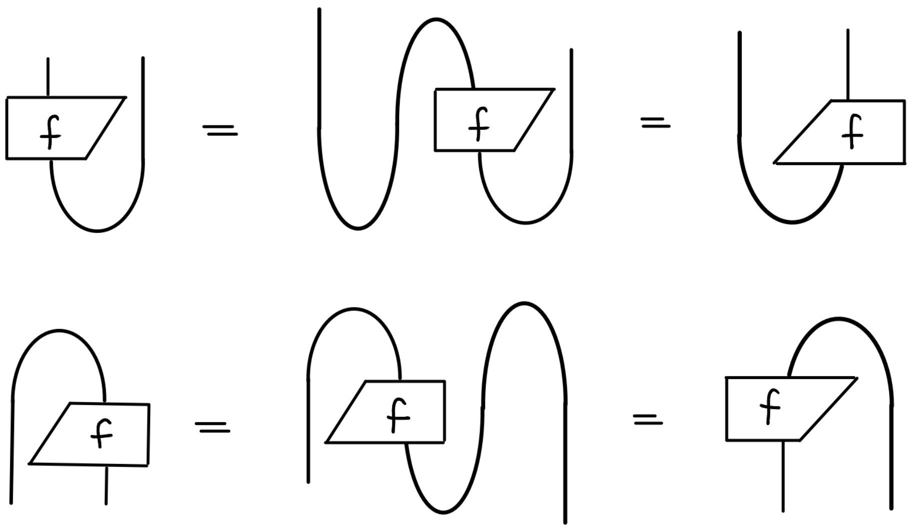


## Transpose

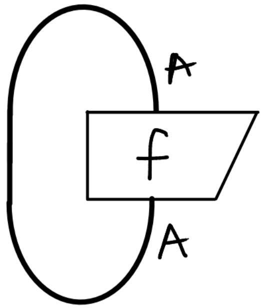



Because of this clever notation, we have:

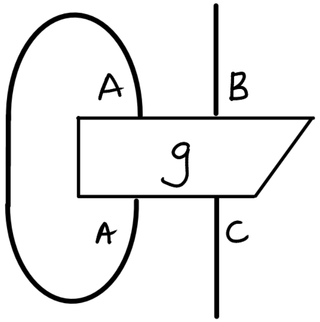


## Trace and Partial Trace

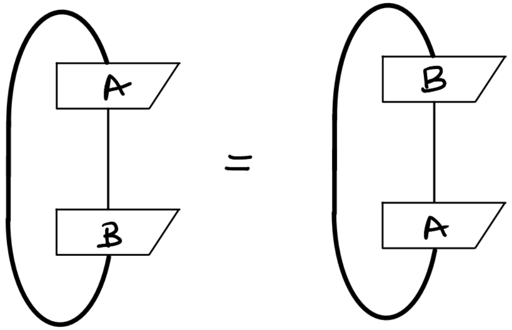
The trace of a process $f$ that has the same input and output type $A$ is given by:



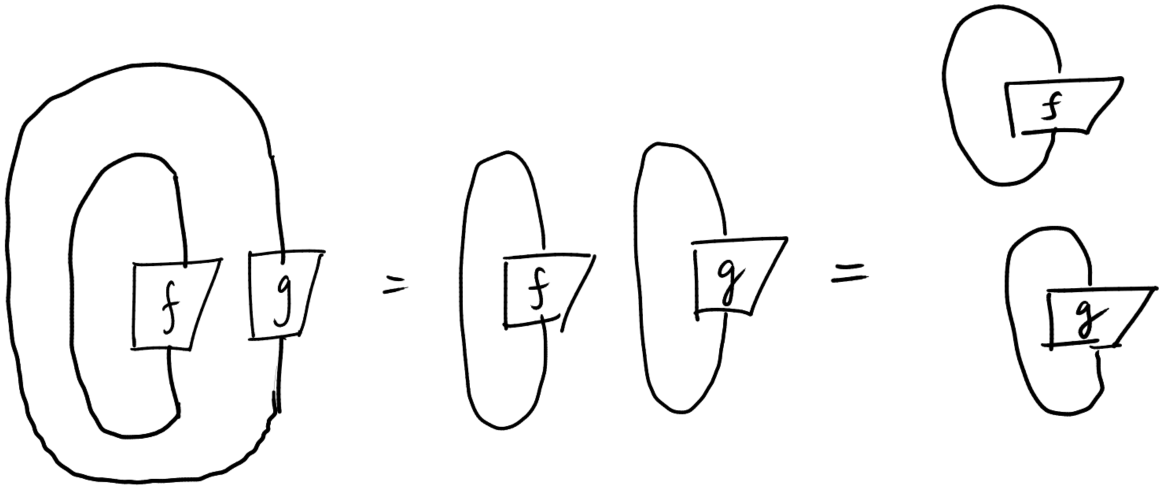

The partial trace of a process $g$ that has one same input and output type $A$ is given by:



$$\mathrm{Tr}(A \circ B) = \mathrm{Tr}(B \circ A)$$

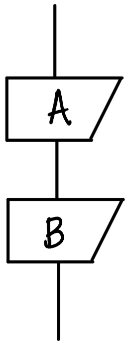
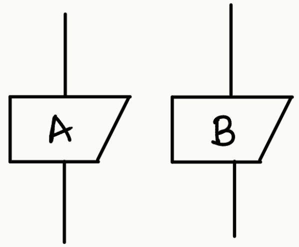


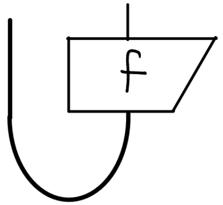
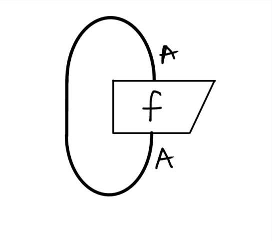
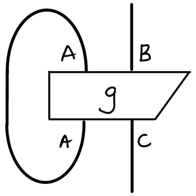

$$\mathrm{Tr}(f \otimes g) = \mathrm{Tr}(f) \circ \mathrm{Tr}(g)$$



# Linear maps

- *Linear maps* is a process theory that admits string diagrams, sums, has an orthonormal basis for each system type, and the "numbers" in the process theory are the complex numbers.
- It is possible to show a one-to-one correspondence between matrices of complex numbers and processes in the *linear maps* process theory
- States correspond to column matrices, effects correspond to row matrices and numbers correspond to $1 \times 1$ matrices.
- Hence, it is possible to verify whether any equation in matrix notation also holds in the *linear maps* process theory.

| Artefact | Matrix Notation | Diagrammatic notation |
|---|---|---|
| Matrix multiplication<br>Sequential composition | $AB$ |  |
| Tensor Product<br>Parallel composition | $A \otimes B$ |  |

| Artefact | Matrix Notation | Diagrammatic notation |
| --- | --- | --- |
| Vectorization | $\mathrm{vec}(f)$ |  |
| Trace | $\mathrm{Tr}(f)$ |  |
| Partial Trace | $\mathrm{Tr}_{\mathrm{A}}(g)$ |  |

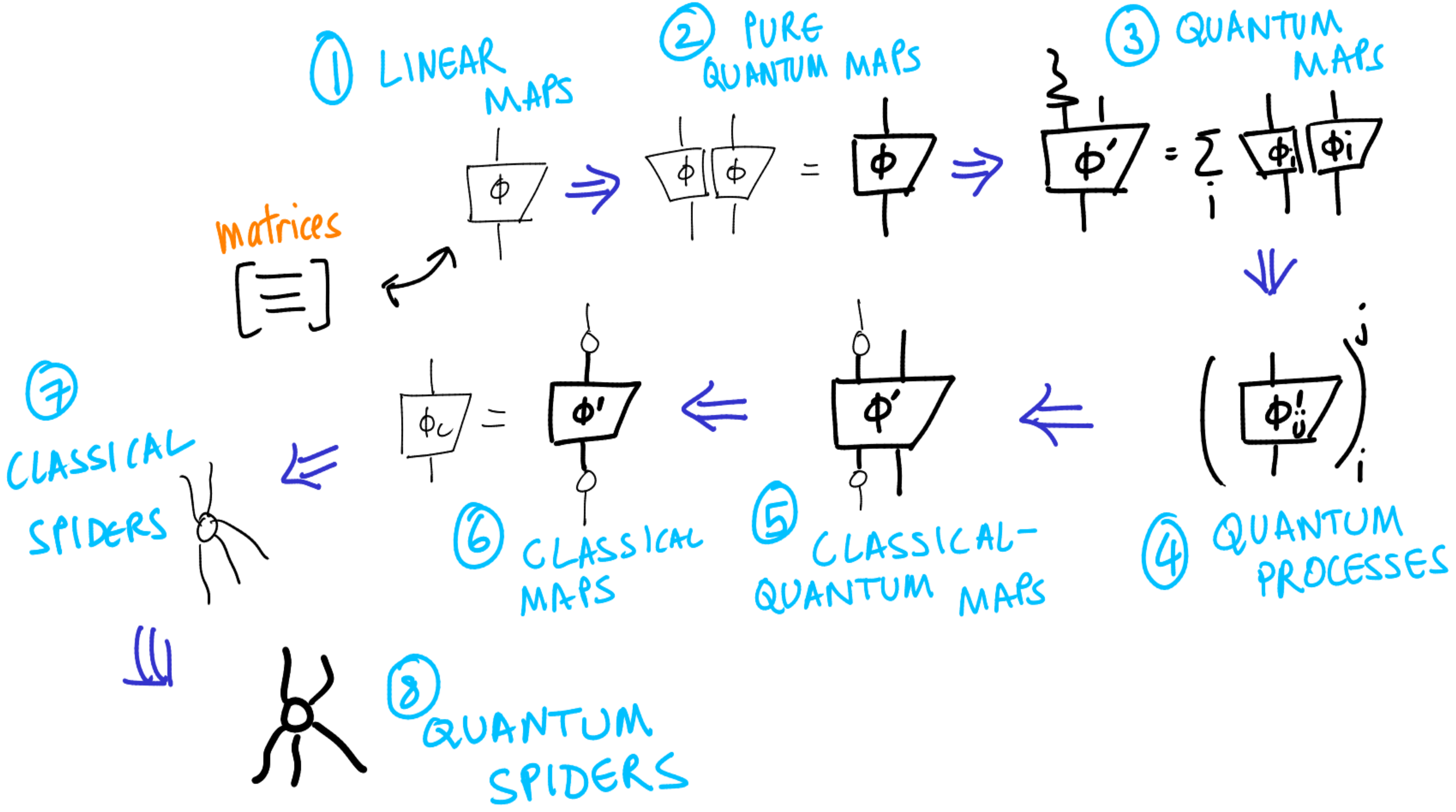
# Questions!

1. Prove $\mathrm{vec}(ABC) = (C^T \otimes A)\,\mathrm{vec}(B)$
2. Prove $\mathrm{Tr}\left(\rho_{AB}(I_A \otimes \sigma_B)\right) = \mathrm{Tr}\left(\mathrm{Tr}_A(\rho_{AB})\sigma_B\right)$

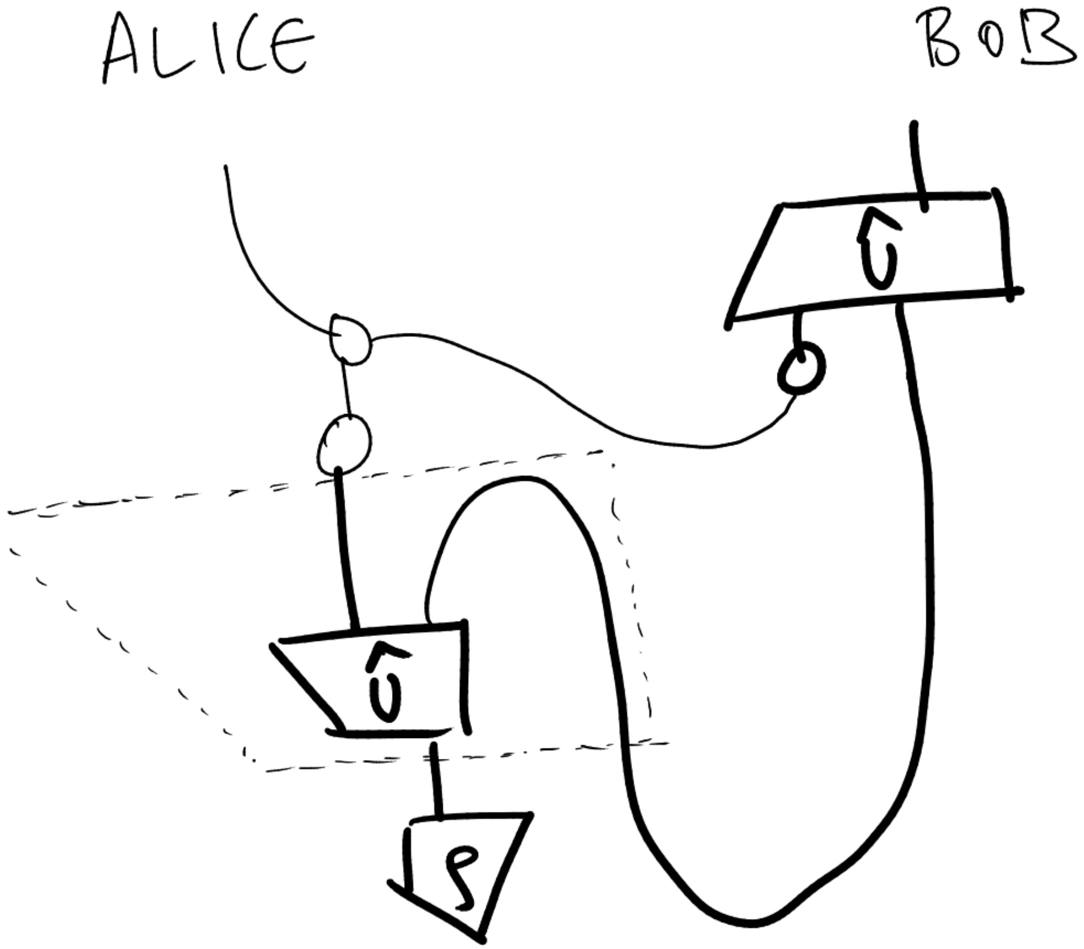
**Theorem:** $\mathrm{Tr}\left(\rho_{AB}(I_A \otimes \sigma_B)\right) = \mathrm{Tr}(\rho_B \sigma_B)$

**Algebraic Proof:**

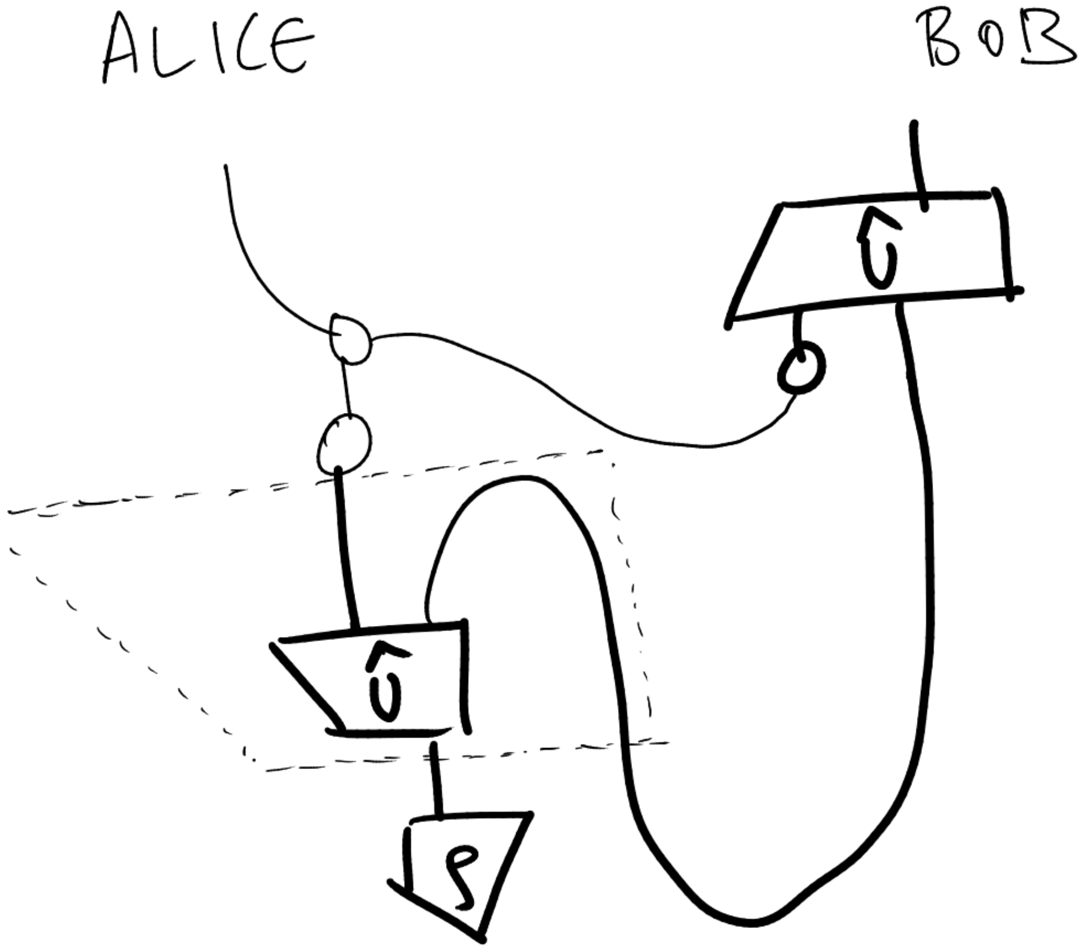
$$\begin{aligned}
\mathrm{Tr}\left(\rho_{AB}(I_A \otimes \sigma_B)\right) &= \mathrm{Tr}\left(\sum_{ijkl} p_{ijkl}\left(|i\rangle\langle j| \otimes |k\rangle\langle l|\right)(I_A \otimes \sigma_B)\right) \\
&= \sum_{ijkl} p_{ijkl}\,\mathrm{Tr}(|i\rangle\langle j|)\,\mathrm{Tr}(|k\rangle\langle l|\sigma_B) \\
&= \sum_{jkl} p_{jjkl}\,\mathrm{Tr}(|k\rangle\langle l|\sigma_B) \\
&= \mathrm{Tr}(\rho_B\sigma_B)
\end{aligned}$$



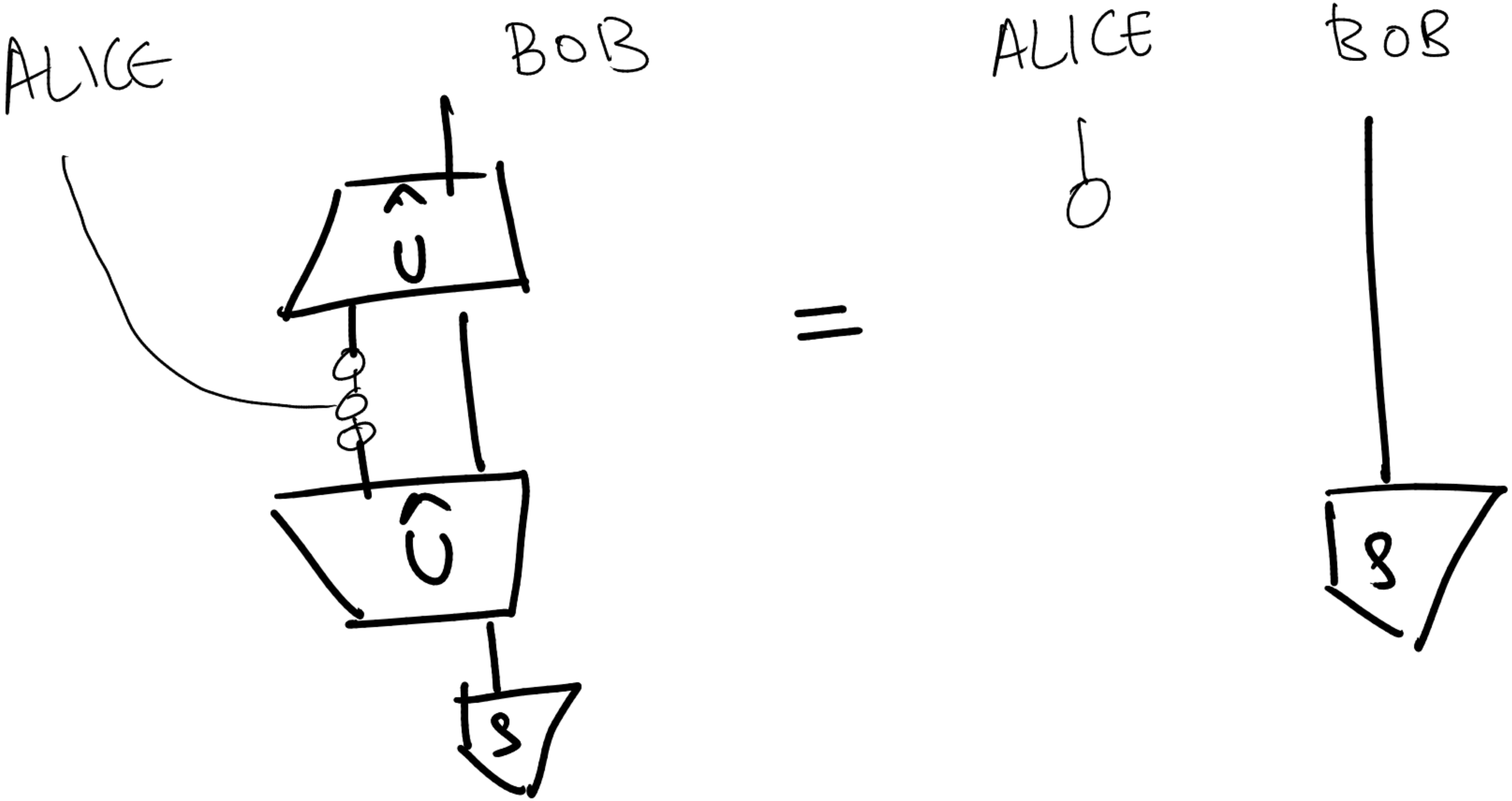
1 LINEAR MAPS
2 PURE QUANTUM MAPS
3 QUANTUM MAPS
matrices
4 QUANTUM PROCESSES
5 CLASSICAL-QUANTUM MAPS
6 CLASSICAL MAPS
7 CLASSICAL SPIDERS
8 QUANTUM SPIDERS


ALICE
BOB
Û
Û
ρ

# Teleportation



## continued ...



This talk is based on the first few chapters of the book *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*

# Thank you!