Bob:
- Ch. 01 Processes as diagrams
- Ch. 02 String diagrams
- Ch. 03 Hilbert space from diagrams
- Ch. 04 Quantum processes
- Ch. 05 Quantum measurement
- Ch. 06 Picturing classical processes

Aleks:
- Ch. 07 Picturing phases and complementarity
- Ch. 08 Quantum theory: the full picture
- Ch. 09 Quantum computing
- Ch. 10 Quantum foundations
— Ch. 1 – Processes as diagrams —

*Philosophy [i.e. physics] is written in this grand book—I mean the universe— which stands continually open to our gaze, but it cannot be understood unless one first learns to comprehend the language and interpret the characters in which it is written. It is written in the language of mathematics, and its characters are triangles, circles, and other geometrical figures, without which it is humanly impossible to understand a single word of it; without these, one is wandering around in a dark labyrinth.*


Here we introduce:

- process theories
- diagrammatic language
— Ch. 1 – Processes as diagrams —

– processes as boxes and systems as wires –
— Ch. 1 – Processes as diagrams —

— processes as boxes and systems as wires —

\[
\text{beam splitter} \\
\begin{array}{c}
\text{photons} \\
\text{photons} \\
\text{photons}
\end{array}
\]
— Ch. 1 – Processes as diagrams —

– processes as boxes and systems as wires –

\[
\text{quicksort} \quad \text{sorted lists} \quad \text{lists}
\]
— Ch. 1 – Processes as diagrams —

– processes as boxes and systems as wires –
— Ch. 1 – Processes as diagrams —

– processes as boxes and systems as wires –

[Diagram: A box labeled "baby" with inputs labeled "noise", "food", and outputs labeled "poo", "love"]
— Ch. 1 – Processes as diagrams —

– processes as boxes and systems as wires –

\[ x^2 + y \]
— Ch. 1 – Processes as diagrams —

– composing processes –
— Ch. 1 – Processes as diagrams —

— composing processes —

```
A    C     D
|    |    |
|    |    |
|    |    |
B    g    \nD
```

```
\textit{and}
```

```
D    A
```

```
— Ch. 1 – Processes as diagrams —

– composing processes –
— Ch. 1 – Processes as diagrams —

– composing processes –

— uk-plug —

plugstrip

— uk-plug —

— eu-plug —

adaptor

— uk-plug —

power drill

— eu-plug —
— Ch. 1 — Processes as diagrams —

— composing processes —

- power drill
  - eu-plug
- adaptor
  - uk-plug
  - uk-plug
- plugstrip
  - uk-plug

:=
— Ch. 1 – Processes as diagrams —

– process theories –
... consist of:

- set of systems \( S \)
- set of processes \( P \), with ins and outs in \( S \),
... consist of:

- set of systems $S$
- set of processes $P$, with ins and outs in $S$,

which are:

- closed under “plugging”.
... consist of:

- set of systems $S$
- set of processes $P$, with ins and outs in $S$,

which are:

- closed under “plugging”.

They tell us:

- how to interpret boxes and wires,
- and hence, when two diagrams are equal.
— Ch. 1 – Processes as diagrams —

— process theories —

\[
\begin{align*}
\text{quicksort} & ::= \\
& \begin{cases}
qs [] &= [] \\
qs (x :: xs) &= \\
& \quad qs [y \mid y <- xs; y < x] ++ [x] ++ \\
& \quad qs [y \mid y <- xs; y \geq x]
\end{cases}
\end{align*}
\]
Ch. 1 – Processes as diagrams —

– process theories –

\[
\text{quicksort} := \begin{cases}
\text{qs} \; [\; ] = \; [\; ] \\
\text{qs} \; (x :: xs) = \\
\quad \text{qs} \; [y \mid y \leftarrow xs; \; y < x] \; ++ \; [x] \; ++ \\
\quad \text{qs} \; [y \mid y \leftarrow xs; \; y \geq x]
\end{cases}
\]
Ch. 1 – Processes as diagrams

– process theories –

\[ \begin{align*}
  x^2 + y 
\end{align*} \]
Ch. 1 – Processes as diagrams –

– process theories –

\[ x^2 + y \quad := \quad \begin{array}{l}
\end{array} \]

\[ x^2 + y = x + y = -x \]
— Ch. 1 – Processes as diagrams —

— diagrams symbolically —
— Ch. 1 – Processes as diagrams —

– diagrams symbolically –

\[
\begin{align*}
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad //
— Ch. 1 – Processes as diagrams —

– diagrams symbolically –

\[ f^{A_1A_2} g^{B_1C_1} h^{D_1} \]
Thm. Diagrams $\equiv$ these symbolic expressions.
— Ch. 1 – Processes as diagrams —

– composing diagrams –
— Ch. 1 – Processes as diagrams —

– composing diagrams –

Two operations:

“$f \otimes g$” := “$f \text{ while } g$”
— Ch. 1 – Processes as diagrams —

— composing diagrams —

Two operations:

\[
\begin{align*}
\text{"f \circ g"} & : = \text{"f after g"} \quad (1)
\end{align*}
\]

\[
\text{Diagram:} \quad \circ \quad A \quad B
\]

\[
\begin{align*}
\text{Diagram:} & \quad g \\
\text{Diagram:} & \quad f
\end{align*}
\]
— Ch. 1 – Processes as diagrams —

— composing diagrams —

Two operations:

\[
\text{“} f \otimes g \text{” :} = \text{“} f \text{ while } g \text{”}
\]

\[
\text{“} f \circ g \text{” :} = \text{“} f \text{ after } g \text{”}
\]

These are:

• associative

• have as respective units:

  – ‘empty’-diagram
  – ‘wire’-diagram
— Ch. 1 – Processes as diagrams —

– circuits –
Defn. ... := can be build with $\otimes$ and $\circ$. 
Defn. ... := can be build with $\otimes$ and $\circ$.

Thm. Circuit $\iff$ no box ‘above’ itself.
Defn. ... := can be build with $\otimes$ and $\circ$.

Thm. Circuit $\iff$ no box ‘above’ itself.

Defn. ... := can be build with \( \otimes \) and \( \circ \).

Thm. Circuit \( \iff \) no box ‘above’ itself.


Not circuit:
— Ch. 1 – Processes as diagrams —

– why diagrams? –
Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?
Ch. 1 – Processes as diagrams

– why diagrams? –

Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since equations come for free!
— Ch. 1 – Processes as diagrams —

– why diagrams? –

Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since equations come for free!

\[(f \otimes g) \otimes h = \begin{array}{c} f \\ \hline \hline \end{array} \begin{array}{c} g \\ \hline \hline \end{array} \begin{array}{c} h \\ \hline \hline \end{array} = f \otimes (g \otimes h)\]
— Ch. 1 – Processes as diagrams —

– why diagrams? –

Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since equations come for free!

\[(f \otimes g) \otimes h = \begin{array}{c} f \\ \otimes \end{array} \begin{array}{c} g \\ \otimes \end{array} \begin{array}{c} h \\ \otimes \end{array} = f \otimes (g \otimes h)\]

\[f \otimes 1_I = \begin{array}{c} f \\ \otimes \end{array} = f\]
— Ch. 1 – Processes as diagrams —

– why diagrams? –

Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since equations come for free!

\[
\begin{align*}
& (g_1 \otimes g_2) \circ (f_1 \otimes f_2) \\
\end{align*}
\]
Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since all equations come for free!
Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since all equations come for free!
— Ch. 1 – Processes as diagrams —

– why diagrams? –

Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since all equations come for free!
— Ch. 1 – Processes as diagrams —

– why diagrams? –

Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since all equations come for free!
— Ch. 1 – Processes as diagrams —

– why diagrams? –

Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since all equations come for free!
Since ‘by definition’ circuits can be build by means of symbolic connectives, why bother with diagrams?

Since all equations come for free!
— Ch. 1 – Processes as diagrams —

– special processes/diagrams –
State :=

\[
\psi
\]
— Ch. 1 – Processes as diagrams —

– special processes/diagrams –

State :=

\[ \psi \]

Effect / Test :=

\[ \pi \]
— Ch. 1 – Processes as diagrams —

— special processes/diagrams —

State := 

\[ \psi \]

Effect/Test := 

\[ \pi \]

Number := 

\[ \lambda \]
Born rule :=
— Ch. 1 – Processes as diagrams —

– special processes/diagrams –

Dirac notation :=

\[
\begin{array}{cccc}
\downarrow & \circ & \rightarrow & \uparrow \\
\uparrow & \circ & \leftarrow & \downarrow \\
\uparrow & \circ & \leftarrow & \uparrow \\
\downarrow & \circ & \rightarrow & \downarrow \\
\end{array}
\]
— Ch. 1 – Processes as diagrams —

– special processes/diagrams –

Separable $\equiv$ disconnected $:=$

\[
\begin{array}{c}
  f \\
  = \\
  f_1 \quad f_2
\end{array}
\]
Separable $\equiv$ disconnected :=

\[
\begin{array}{c}
\psi \\
\hline \\
\hline \\
\hline \\
\hline \\
\hline
\end{array} = \begin{array}{c}
\psi_1 \\
\hline \\
\hline \\
\hline \\
\hline \\
\hline
\end{array} \quad \quad \begin{array}{c}
\psi_2 \\
\hline \\
\hline \\
\hline \\
\hline \\
\hline
\end{array} = \begin{array}{c}
\pi \\
\hline \\
\hline \\
\hline \\
\hline \\
\hline
\end{array}
\]
— Ch. 1 – Processes as diagrams —

– special processes/diagrams –

Non-separable := way more interesting!
When two systems, of which we know the states by their respective representatives, enter into temporary physical interaction due to known forces between them, and when after a time of mutual influence the systems separate again, then they can no longer be described in the same way as before, viz. by endowing each of them with a representative of its own. I would not call that one but rather the characteristic trait of quantum mechanics, the one that enforces its entire departure from classical lines of thought.

— Erwin Schrödinger, 1935.

Here we:

• introduce a wilder kind of diagram
• define quantum notions in great generality
• derive quantum phenomena in great generality
— Ch. 2 – String diagrams —

– process-state duality –

Exists state $\cup$ and effect $\cap$:
— Ch. 2 – String diagrams —

– process-state duality –

Exists state $\cup$ and effect $\cap$:

such that:

$(a)$

$(b)$
— Ch. 2 – String diagrams —

— process-state duality —

proof of duality:
— Ch. 2 – String diagrams —

— process-state duality —

proof of duality:
Change notation:

\[
\begin{align*}
\quad & := \\
\quad & := \\
\end{align*}
\]
Change notation:

\[ \quad := \quad \quad \quad := \quad \quad \quad \quad \quad := \quad \quad \quad \quad \quad := \quad \]

so that now:

\[ \quad = \quad \quad = \quad \quad = \quad \]
— Ch. 2 – String diagrams —

– definition –
Thm. TFAE:

- circuits with process-state duality and:

\[
\begin{align*}
\begin{array}{c}
\includegraphics{diagram.png}
\end{array}
\end{align*}
\]

= 

\[
\begin{align*}
\begin{array}{c}
\includegraphics{diagram.png}
\end{array}
\end{align*}
\]
Thm. TFAE:

- circuits with process-state duality and:

- diagrams with in-in and out-out connection:
— Ch. 2 – String diagrams —

– definition –

\[ f^{A_1A_2} g^{BD} h^{D}_{A_3} \]
— Ch. 2 – String diagrams —

– transpose –
... \( \overset{f}{=} \)
... :=

\[ f \]
Prop. The transpose is an involution:
Prop. Transpose of ‘cup’ is ‘cap’:
Clever new notation:
— Ch. 2 – String diagrams —

– transpose –

Clever new notation:

\[ f \quad ::= \quad f \]

⇒ just what happens when yanking hard!
Prop. Sliding:
--- Ch. 2 – String diagrams ---

– transpose –

**Prop.** Sliding:

\[
\begin{align*}
&= \quad \quad = \quad \quad = \\
\end{align*}
\]

**Pf.**

\[
\begin{align*}
&= \quad \quad = \\
\end{align*}
\]
Prop. Sliding:

\[
\begin{align*}
\text{[Diagram]} & \quad = \quad \text{[Diagram]} & \quad = \quad \text{[Diagram]}
\end{align*}
\]

... so this is a mathematical equation:
— Ch. 2 – String diagrams —

– trace –
\[ ... := \]

\[ \begin{tikzpicture}
  \node (f) at (0,0) {$f$};
  \node (A) at (2,0) {$A$};
  \draw[->] (f) to [out=90,in=180] (A);
\end{tikzpicture} \]
Partial ... :=

\[
\begin{array}{c}
\text{C} \\
\downarrow f \\
\text{B} \\
\end{array}
\]

\[\Rightarrow \]

\[\text{A}\]
Prop. Cyclicity:

\[
\begin{array}{c}
g \\
\hline
f \\
g \\
\end{array}
\quad = \quad
\begin{array}{c}
f \\
\hline
\end{array}
\]

\[
\begin{array}{c}
g \\
\hline
f \\
f \\
g \\
\end{array}
\]
Prop. Cyclicity:

\[
\begin{array}{c}
g \\
\text{f} \\
g
\end{array}
\quad = 
\begin{array}{c}
f \\
g \\
g
\end{array}
\]

Redundant but fun ‘ferris wheel’ proof:
— Ch. 2 – String diagrams —

— ‘quantum’-like features –

\[
\frac{\text{classical}}{\text{quantum}} = \begin{array}{c} \psi \end{array} = \begin{array}{c} \psi_1 \end{array} \begin{array}{c} \psi_2 \end{array} \neq \begin{array}{c} \psi_1 \end{array} \begin{array}{c} \psi_2 \end{array}
\]
Thm. All states separable $\Rightarrow$ rubbish theory.
**Thm.** All states separable $\Rightarrow$ rubbish theory.

**Lem.** All states separable $\Rightarrow$ wires separable.

**Pf.**
Perfect correlations:
Perfect correlations:
Logical reading:
— Ch. 2 – String diagrams —

– ‘quantum’-like features –

Operational reading:

\[ \begin{align*}
  & t_4 \\
  & \quad \vdash \\
  & t_3 \\
  & \quad \vdash \\
  & t_2 \\
  & \quad \vdash \\
  & t_1 \\
\end{align*} \quad = \quad \begin{align*}
  & t_4 \\
  & \quad \vdash \\
  & t_3 \\
  & \quad \vdash \\
  & t_2 \\
  & \quad \vdash \\
  & t_1 \\
\end{align*} \]
Realising time-reversal (and make NY times):

step 1:

\[ f \]

\[ g \]

step 2:

\[ corr \]

\[ sys \]

step 3:
Thm. No-cloning from assumptions:
Pf.
— Ch. 2 – String diagrams —

– ‘quantum’-like features –

Pf.
— Ch. 2 – String diagrams —

– ‘quantum’-like features –

Pf.
— Ch. 2 – String diagrams —

– ‘quantum’-like features –

Pf.
— Ch. 2 – String diagrams —

– ‘quantum’-like features –

Pf.
— Ch. 2 – String diagrams —

– ‘quantum’-like features –

\[ \text{l} \overset{\text{LHS}}{=} \quad \text{Diagram 1} \quad = \quad \text{Diagram 2} \quad = \quad \text{l} \overset{\text{RHS}}{=} \]

[Diagram 1 and Diagram 2 are complex string diagrams involving various lines and nodes, indicating some form of equality or transformation in string diagram theory.]
— Ch. 2 – String diagrams —

— ‘quantum’-like features —
— Ch. 2 – String diagrams —

— adjoint & conjugate —
A ‘ket’ sometimes wants to be ‘bra’:
— Ch. 2 – String diagrams —

– adjoint & conjugate –

Conjugate :=

\[
\begin{array}{c}
\text{Conjugate} :=
\end{array}
\]

![Diagram for Conjugate](image)

Adjoint :=

\[
\begin{array}{c}
\text{Adjoint} :=
\end{array}
\]

![Diagram for Adjoint](image)
Unitarity/isometry :=

\[
\begin{array}{c}
\begin{array}{c}
\hline
U \\
\hline
\end{array}
\end{array}
= \\
\begin{array}{c}
\begin{array}{c}
\hline
U \\
\hline
\end{array}
\end{array}
\]
Teleportation:
Entanglement swapping:
— Ch. 2 – String diagrams —
— designing teleportation —

Aleks

Bob

= 

Aleks

Bob
— Ch. 2 – String diagrams —

– designing teleportation –

[Diagram showing string diagrams for teleportation process]
— Ch. 2 – String diagrams —

– designing teleportation –

Bob’s problem now!
— Ch. 3 – Hilbert space from diagrams —

*I would like to make a confession which may seem immoral: I do not believe absolutely in Hilbert space any more.*


Here we introduce:

- ONBs, matrices and sums
- (multi-)linear maps & Hilbert space

and relate:

- string diagrams
- (multi-)linear maps & Hilbert space
Ch. 3 – Hilbert space from diagrams

ONB
A set:
\[ B = \left\{ 1, \ldots, n \right\} \]

is pre-basis if:
\[
\left( \forall i : \begin{array}{c}
\textbf{f} \\
_i
\end{array} = \begin{array}{c}
\textbf{g} \\
_i
\end{array} \right) \implies \begin{array}{c}
\textbf{f} \\
= \\
\textbf{g}
\end{array}
\]
Orthonormal :=

\[ \delta_{ij} \]
Orthonormal :=

\[
\begin{align*}
\overset{j}{\downarrow} & \quad = \quad \delta_{ij} \\
\overset{i}{\downarrow}
\end{align*}
\]

Canonical :=

\[
\begin{align*}
\overset{i}{\downarrow} & \quad := \quad \overset{i}{\downarrow} \\
\overset{i}{\downarrow} & \quad = \quad \overset{i}{\downarrow}
\end{align*}
\]
— Ch. 3 – Hilbert space from diagrams —

– matrix calculus –
Thm. We have:

\[
\forall i, j : \begin{align*}
\begin{array}{c}
\text{f} \\
\downarrow \quad \downarrow
\end{array} & =
\begin{array}{c}
\text{g} \\
\downarrow \quad \downarrow
\end{array}
\end{align*}
\]

so there is a matrix:

\[
\begin{pmatrix}
\text{f}_{11} & \text{f}_{12} & \cdots & \text{f}_{1n} \\
\text{f}_{21} & \text{f}_{22} & \cdots & \text{f}_{2n} \\
\cdots & \cdots & \cdots & \cdots \\
\text{f}_{m1} & \text{f}_{m2} & \cdots & \text{f}_{mn}
\end{pmatrix}
\]

with

\[
\text{f}_{ij} := \begin{align*}
\begin{array}{c}
\text{i} \\
\downarrow \quad \downarrow
\end{array} & =
\begin{array}{c}
\text{f} \\
\downarrow \quad \downarrow
\end{array} =
\begin{array}{c}
\text{j} \\
\downarrow \quad \downarrow
\end{array}
\end{align*}
\]
But one also may want to ‘glue’ things together:

\[ g = \sum_{ij} g_{ij} \]
--- Ch. 3 – Hilbert space from diagrams ---

– matrix calculus –

Sums := for \( \{ f_i \}_i \) of the same type there exists:

\[
\sum_{i=1}^{N} f_i
\]

which ‘moves around’:

\[
\sum_i \left( \begin{array}{c}
  g \\
  h_i \\
  g \\
\end{array} \right) f = \sum_i \left( \begin{array}{c}
  g \\
  h_i \\
  f \\
  g \\
\end{array} \right)
\]
In:

\[ \sum_j g_j = \sum_j \sum_i f_i = \sum_i f_i \sum_j = \sum_{ij} \]

the intuition is:

\[ \sum_i \]
In:

\[ \sum_j g_j \sum_i f_i = \sum_j f_i \sum_i g_j = \sum_i f_i \sum_j g_j = \sum i_j \]

the intuition is:

\[ \sum_i \]

but better (see later):

\[ \sum \]
— Ch. 3 — Hilbert space from diagrams —

— \textit{definition} —
Defn.

Linear maps := String diagrams s.t.:

•

•

•
Defn.

**Linear maps** := String diagrams s.t.:

- each system has ONB
Defn.

Linear maps := String diagrams s.t.:
- each system has ONB
- ∃ sums
Defn.

**Linear maps** := String diagrams s.t.:

- each system has ONB
- ∃ sums
- numbers are $\mathbb{C}$
Defn.

Linear maps := String diagrams s.t.:
  • each system has ONB
  • ∃ sums
  • numbers are \( \mathbb{C} \)

Hilbert space := states for a system with Born-rule.
THM. (Selinger, 2008)

An equation between string diagrams holds, if and only if it holds for Hilbert spaces and linear maps.
THM. (Selinger, 2008)

An equation between string diagrams holds, if and only if it holds for Hilbert spaces and linear maps.

I.e. defining Hilbert spaces and linear maps in this manner is a ‘conservative extension’ of string diagrams.
The art of progress is to preserve order amid change, and to preserve change amid order.


Here we introduce in terms of diagrams:

- pure quantum maps
- mixed/open quantum maps
- causality & Stinespring dilation
- general quantum processes done badly
— Ch. 4 – Quantum processes —

– doubling –
Goal 1:

test \{ \pi \} \quad \text{probability}

state \{ \psi \}
Goal 1:

\[
\begin{array}{c}
\text{test} \\
\{ \pi \}
\end{array}
\]

\[
\begin{array}{c}
\text{state} \\
\{ \psi \}
\end{array}
\]

\[
\text{probability}
\]

Goal 2:

\[
\psi \sim \lambda \psi
\]
Pure quantum state :=
Pure quantum effect :=

\[ \phi \]

\[ \phi \]

\[ \phi \]

\[ \phi \]
— Ch. 4 – Quantum processes —

– doubling –

⇒ genuine probabilities
--- Ch. 4 – Quantum processes ---

– doubling –

Pure quantum map :=

\[ \hat{f} := f f \]
Thm. We have:

\[ \hat{f} = \hat{g} \]

if and only if there exist \( \lambda \bar{\lambda} = \mu \bar{\mu} \):

\[ \lambda \begin{array}{c} f \end{array} = \mu \begin{array}{c} g \end{array} \]
--- Ch. 4 – Quantum processes ---

--- doubling ---

**Pf. Setting:**

\[
\lambda := \begin{array}{c}
\begin{array}{c}
\text{f} \\
\text{f}
\end{array}
\end{array} \quad \mu := \begin{array}{c}
\begin{array}{c}
\text{g} \\
\text{f}
\end{array}
\end{array}
\]

then:

\[
\lambda \tilde{\lambda} = \begin{array}{c}
\begin{array}{c}
\text{f} \\
\begin{array}{c}
\text{f} \\
\begin{array}{c}
\text{f} \\
\text{f}
\end{array}
\end{array}
\end{array}
\end{array} = \begin{array}{c}
\begin{array}{c}
\text{g} \\
\begin{array}{c}
\text{f} \\
\begin{array}{c}
\text{f} \\
\text{g}
\end{array}
\end{array}
\end{array}
\end{array} = \mu \tilde{\mu}
\]
— Ch. 4 – Quantum processes —

– doubling –

Pf. Setting:

\[
\lambda := \begin{array}{c}
\text{Diagram 1}
\end{array} \quad \mu := \begin{array}{c}
\text{Diagram 2}
\end{array}
\]

then:

\[
\lambda \begin{array}{c}
\text{Diagram 3}
\end{array} = \begin{array}{c}
\text{Diagram 4}
\end{array} = \begin{array}{c}
\text{Diagram 5}
\end{array} = \mu \begin{array}{c}
\text{Diagram 6}
\end{array}
\]
Ch. 4 – Quantum processes –

– open systems –
— Ch. 4 – Quantum processes —

– open systems –

Discarding :=

\begin{align*}
\begin{array}{c}
\text{Diagram 1} \\
\text{Diagram 2}
\end{array}
\end{align*}
Thm. Discarding is not a pure quantum map.
Discarding :=

\[ \begin{array}{c}
\begin{array}{c}
\text{\_}\text{\_}\text{\_}\\
\text{\text{\_\_\_}}
\end{array}
\end{array} \quad := \quad
\begin{array}{c}
\begin{array}{c}
\text{\_\_\_\_}\\
\text{\text{\_\_\_\_\_\_\_\_\_}}
\end{array}
\end{array} \]

**Thm.** Discarding is not a pure quantum map.

**Pf.** Something connected \( \neq \) something disconnected.
— Ch. 4 – Quantum processes —

– open systems –

Quantum maps := pure quantum maps + discarding
— Ch. 4 – Quantum processes —

– open systems –

Quantum maps := pure quantum maps + discarding

E.g. ‘maximally mixed state :=

\[
\frac{1}{D} \quad \quad \quad = \quad \quad \quad \frac{1}{D}
\]
Quantum maps := pure quantum maps + discarding

**Prop.** All quantum maps are of the form:
Quantum maps := pure quantum maps + discarding

**Prop.** All quantum maps are of the form:
Quantum maps := pure quantum maps + discarding

Prop. All quantum maps are of the form:
— Ch. 4 – Quantum processes —

– causality –
— Ch. 4 – Quantum processes —

– causality –

... of quantum maps:
— Ch. 4 – Quantum processes —

– causality –

**Prop.** For pure quantum maps:

\[ \text{causality} \iff \text{isometry} \]
— Ch. 4 – Quantum processes —

– *causality* –

**Prop.** For pure quantum maps:

\[
\text{causality} \iff \text{isometry}
\]

**Pf.**

\[
\begin{array}{c}
\begin{array}{ccc}
U & \quad & U \\
\hline
\end{array}
\end{array} = \\
\begin{array}{c}
\begin{array}{c}
\hline
\end{array}
\end{array}
\]
Prop. For general quantum maps:

causality $\iff$ of the form $\hat{U}$
Prop. For general quantum maps:

causality $\iff$ of the form $\hat{U}$

Pf. $\hat{U} = \hat{U}$
Prop. For general quantum maps:

causality $\iff$ of the form $\hat{U}$

Pf.

Cor. Stinespring dilation.
— Ch. 4 – Quantum processes —

– non-deterministic quantum processes –
such that

\[ \sum_{i} \Phi^i = \text{E.g. quantum measurements.} \]
The bureaucratic mentality is the only constant in the universe.

— Dr. McCoy, Star Trek IV: The Voyage Home, 2286.

Here we briefly address:

• Next-best-thing to observing
• Measurement-induced dynamics
• Measurement-only quantum computing
— Ch. 5 – Quantum measurement —

– is quantum measurement weird? –
— Ch. 5 – Quantum measurement —

—is quantum measurement weird?—

Thm. Observing is not a quantum process
— Ch. 5 – Quantum measurement —

—is quantum measurement weird? —

**Thm.** Observing is not a quantum process i.e. $\not \exists$:

$$
\left\{ \Phi^\phi \right\}_\phi \quad \text{with} \quad \Phi^\phi \quad = \quad \left\{ \begin{array}{ll} 1 & \text{iff } \psi = \phi \\ 0 & \text{iff } \psi \neq \phi \end{array} \right.
$$
**Thm.** Observing is not a quantum process i.e. $\#$: 

\[
\left\{ \Phi^{\phi} \right\}_{\phi} \quad \text{with} \quad \Phi^{\psi} = \begin{cases} 1 & \text{iff } \psi = \phi \\ 0 & \text{iff } \psi \neq \phi \end{cases}
\]

**Prop.** Condition can only hold for orthogonal states.
Thm. Observing is not a quantum process i.e. $\not\exists$:

$$\begin{cases}
\Phi_\phi \\
\Phi_\psi
\end{cases} \quad \text{with} \quad \begin{cases}
\Phi_\phi \\
\Phi_\psi
\end{cases} = \begin{cases}
1 \text{ iff } \psi = \phi \\
0 \text{ iff } \psi \neq \phi
\end{cases}$$

Prop. Condition can only hold for orthogonal states.

$\Rightarrow$ “measurement” is next-best-thing to observing
— Ch. 5 – Quantum measurement —

– is quantum measurement weird? –

Bohr-Heisenberg:

*any attempt to observe is bound to disturb*
— Ch. 5 – Quantum measurement —

— is quantum measurement weird? —

Bohr-Heisenberg:

any attempt to observe is bound to disturb

Newtonian equivalent:

locating a balloon by mechanical means
Heisenberg-Bohr: 

*any attempt to observe is bound to disturb*

Newtonian equivalent:

*locating a balloon by mechanical means*

Resulting diagnosis:

*we suffer from quantum-blindness*
— Ch. 5 – Quantum measurement —

– is quantum measurement weird? –

BUT, the stuff that people call quantum measurement turns out to be extremely useful nonetheless!
— Ch. 5 – Quantum measurement —

– what people call measurement –

ONB-measurement :=

\[
\{ \hat{\phi}^i \}_{i}
\]

\[
\{ \hat{\phi}^i \}_{i}
\]
— Ch. 5 – Quantum measurement —

– what people call measurement –

ONB-measurement :=

E.g. for \( \{ \beta_i \}_i \) Pauli-matrices:
Thm. All quantum maps arise from ONB-measurements.
**Thm.** All quantum maps arise from ONB-measurements.

**Pf.** There are ‘enough ONB’s’ such that:
— Ch. 5 – Quantum measurement —

– measurement-induced dynamics –
— Ch. 5 – Quantum measurement —

— measurement-induced dynamics —
— Ch. 5 – Quantum measurement —

– measurement-induced dynamics –

\[ t_0 \]

\[ \rho \]
— Ch. 5 – Quantum measurement —

– measurement-induced dynamics –
— Ch. 5 – Quantum measurement —

– measurement-only quantum computing –
— Ch. 5 – Quantum measurement —

– measurement-only quantum computing –
— Ch. 5 – Quantum measurement —  
— measurement-only quantum computing —
— Ch. 5 – Quantum measurement —

– measurement-only quantum computing –
— Ch. 5 – Quantum measurement —

— measurement-only quantum computing —

\[
\begin{array}{c}
\hat{\gamma}_i \\
\Phi \\
= \\
\Phi \\
\hat{\beta}_i
\end{array}
\]
Damn it! I knew she was a monster! John! Amy! Listen! Guard your buttholes.

— David Wong, This Book Is Full of Spiders, 2012.

Here we fully diagrammatically describe:

• all quantum processes
• special ones
• protocols

and introduce the humongously important notion of:

• spiders
— Ch. 6 – Picturing classical processes —

– classical vs. quantum wires –
They should meet:

\[
\text{quantum wires } \leftrightarrow \text{ classical wires}
\]
They should meet:

\[ \text{quantum wires} \leftrightarrow \text{classical wires} \]

but retain their distance:

\[ \text{quantum wires} \neq \text{classical wires} \]
— Ch. 6 – Picturing classical processes —

– classical vs. quantum wires –

They should meet:

quantum wires \leftrightarrow \text{classical wires}

but retain their distance:

quantum wires \neq \text{classical wires}

which can be realised via ‘un-doubling’:

\[
\begin{align*}
\text{classical wire} & = \text{normal (i.e. 1)} \\
\text{quantum wire} & = \text{boldface (i.e. 2)}
\end{align*}
\]
— Ch. 6 – Picturing classical processes —

– encoding classical data –
Classical data $\equiv$ ONB:
Classical data $\equiv$ ONB:

- $\tikz[baseline=-2pt]
\draw (-.3,.3) -- (0,0) -- (.3,.3) node[above] {$i$};$ := “providing classical value $i$”
— Ch. 6 – Picturing classical processes —

– encoding classical data –

Classical data $\equiv$ ONB:

- $\begin{array}{c}
\text{i} \\
\downarrow
\end{array}$ := “providing classical value $i$”

- $\begin{array}{c}
\text{i} \\
\uparrow
\end{array}$ := “testing for classical value $i$”
Classical data \( \equiv \) ONB:

- \( \begin{array}{c}
\triangleleft \\
i
\end{array} \) := “providing classical value \( i \)”

- \( \begin{array}{c}
\triangleright \\
i
\end{array} \) := “testing for classical value \( i \)”

Sanity check:

\[
\begin{array}{c}
\triangleright \\
j
\end{array} \\
\triangleleft \\
i
\nonumber = \delta_{i,j}
\]
Non-deterministic quantum process:
Non-deterministic quantum process:

\[
\begin{align*}
\{ \Phi^i \} & \sim \quad \{ \Phi_i \} \\
\{ \Psi^i \} & \sim \quad \{ \Psi_i \}
\end{align*}
\]

Process controlled by outcome:
Ch. 6 – Picturing classical processes —

— encoding classical data —
— Ch. 6 – Picturing classical processes —

– encoding classical data –
— Ch. 6 – Picturing classical processes —

– encoding classical data –
— Ch. 6 – Picturing classical processes —

– encoding classical data –

[Diagram of classical processes with symbols and labels]
— Ch. 6 – Picturing classical processes —

– classical data in diagrams –

Prop. Braces $\equiv$ sums
Prop. Braces $\equiv$ sums

Pf.

\[
\sum_i \Phi_i \equiv \sum_i \Phi_i \delta_{ij} = \Phi_j
\]
— Ch. 6 – Picturing classical processes —

– encoding classical data –

Non-deterministic quantum process:

\[
\{ \Phi_i \}_{i} \sim \sum_{i} \Phi_i
\]
Non-deterministic quantum process:

\[ \{ \Phi_i \} \sim \sum_i \Phi_i \]

Process controlled by outcome:

\[ \{ \Phi_j \} \sim \sum_j \Phi_j \]
— Ch. 6 – Picturing classical processes —

– encoding classical data –
— Ch. 6 – Picturing classical processes —

– encoding classical data –
— Ch. 6 – Picturing classical processes —

— classical-quantum maps —

... :=

\[
\sum_{ij} \Phi_{ij}
\]
Ch. 6 – Picturing classical processes

– classical-quantum maps –

Classical map :=

\[ \sum_{ij} \Phi_{ij} \]
Classical map examples:

- **Copy**
  
  ![Copy Diagram]

  \[ := \sum_i \ 	riangleleft \]

- **Delete**
  
  ![Delete Diagram]

  \[ := \sum_i \]

- **Match**
  
  ![Match Diagram]

  \[ := \sum_i \ 	riangleleft \]

- **Compare**
  
  ![Compare Diagram]

  \[ := \sum_i \triangleleft \triangleleft \triangleleft \]
The name explains the action:

\[
\text{\begin{figure}[h]
\begin{center}
\begin{tikzpicture}
  \node (i) at (0,0) {$i$};
  \node (j) at (0,-1.5) {$j$};
  \draw (i) -- (j);
\end{tikzpicture}
\end{center}
\end{figure}}
\]

\[
\sum_i \text{\begin{figure}[h]
\begin{center}
\begin{tikzpicture}
  \node (i) at (0,0) {$i$};
  \node (j) at (0,-1.5) {$j$};
  \draw (i) -- (j);
\end{tikzpicture}
\end{center}
\end{figure}} = \text{\begin{figure}[h]
\begin{center}
\begin{tikzpicture}
  \node (j) at (0,-1.5) {$j$};
\end{tikzpicture}
\end{center}
\end{figure}}
\]
The name explains the action:
Classical-quantum map examples:

\[ I := \sum_i i = \sum_i i \]

**encode**

\[ I := \sum_i i \]

**measure**
Thm. ... are always of the form:
Thm. ... are always of the form:

where $f$ is a quantum map.
Thm. ... are always of the form:
Thm. ... are always of the form:
— Ch. 6 – Picturing classical processes —

— classical maps —
—— Ch. 6 – Picturing classical processes ——

— classical maps —

**Thm.** ... are always of the form:
— Ch. 6 – Picturing classical processes —

– classical-quantum processes –
Thm. Causality:

\[
\Phi = 0
\]
— Ch. 6 – Picturing classical processes —

– classical-quantum processes –

Lem.
Thm. Causality:
Thm. Causality:
... :=

\[ \hat{f} \]

s.t.:

\[ \hat{f} = \text{Diagram} \]
— Ch. 6 – Picturing classical processes —

– teleportation diagrammatically –
Thm. Controlled isometry:
Thm. Controlled isometry:
— Ch. 6 – Picturing classical processes —

– teleportation diagrammatically –

Pf.
— Ch. 6 – Picturing classical processes —

— teleportation diagrammatically —
— Ch. 6 – Picturing classical processes —

— teleportation diagrammatically —
— Ch. 6 – Picturing classical processes —

— teleportation diagrammatically —

Aleks

Bob

\[ u \]

\[ u \]

\[ \psi \]
— Ch. 6 – Picturing classical processes —

— teleportation diagrammatically —

Aleks

Bob

ψ
— Ch. 6 – Picturing classical processes —

– dense coding –
— Ch. 6 – Picturing classical processes —

— dense coding —
— Ch. 6 – Picturing classical processes —

– dense coding –
— Ch. 6 – Picturing classical processes —

– Naimark dilation –
— Ch. 6 – Picturing classical processes —

– Naimark dilation –
— Ch. 6 – Picturing classical processes —

– Naimark dilation –
— Ch. 6 – Picturing classical processes —

– spiders –
— Ch. 6 – Picturing classical processes —

— spiders —

\[ \ldots := \sum_{i}^{m} \{ i \ldots i \} \]

\[ \{ n \} \]
— Ch. 6 – Picturing classical processes —

– spiders –

\[ \text{copy} := \sum_i \begin{array}{c} \text{triangle} \text{ down} \text{ triangle} \text{ down} \text{ triangle} \end{array} \]

\[ \text{match} := \sum_i \begin{array}{c} \text{triangle} \text{ down} \end{array} \]

\[ \text{delete} := \sum_i \begin{array}{c} \text{triangle} \end{array} \]

\[ \text{compare} := \sum_i \begin{array}{c} \text{triangle} \text{ down} \text{ triangle} \text{ down} \text{ triangle} \end{array} \]
— Ch. 6 – Picturing classical processes —

– spiders –
— Ch. 6 – Picturing classical processes —

– spiders –
— Ch. 6 – Picturing classical processes —

– spiders –
— Ch. 6 – Picturing classical processes —

– spiders –
Prop. Spiders obey:
Prop. Spiders obey:
For example:
... and in particular:
Thm. Spiders $\equiv$ ONBs
Thm. Spiders $\equiv$ ONBs

Pf. Consider copy spider:

so claim follows by only-orthogonals-are-clonable.
THM. (CPV) All families of linear maps: 

\[
\begin{align*}
&\{ \{ \cdots \cdots \} \} \\
&\{ \{ f_{nm} \} \}
\end{align*}
\]

which ‘behave’ like spiders, are spiders.
Ch. 6 – Picturing classical processes —

– spiders –

Classical spider :=

![Diagram of a classical spider]
Quantum spider :=
— Ch. 6 – Picturing classical processes —

– spiders –

Bastard spider :=

[Diagram of a complex spider-like structure]
Bastard spider :=

\[
\begin{array}{c}
\begin{align*}
\ldots & \quad = \\
\ldots & \\
\ldots & \\
\ldots & \\
\end{align*}
\end{array}
\]


Selinger, P. (2007) Dagger compact closed categories and completely positive maps. ENTCS.


Selinger, P. (2011) Finite dimensional Hilbert spaces are complete for dagger compact closed categories. ENTCS.

