Selling Category Theory to the Masses

Bob Coecke – Quantum Group - Computer Science - Oxford University


Lucas Dixon, R. Duncan, Aleks Kissinger, Alex Merry & Matvey Soloviev. sites.google.com/site/quantomatic/
. . . a tale of food, spiders and Google

Bob Coecke – Quantum Group - Computer Science - Oxford University

Lucas Dixon, R. Duncan, Aleks Kissinger, Alex Merry & Matvey Soloviev. sites.google.com/site/quantomatic/
Task: selling Category Theory to the masses!
Task: selling Category Theory to the masses!

. . . is it just tedious abstract nonsense?
Task: selling Category Theory to the masses!

... is it just tedious abstract nonsense?

No! Categories are everywhere!
1. Let $A$ be a raw potato.
1. Let $A$ be a raw potato. 

$A$ admits many *states* e.g. dirty, clean, skinned, ...
1. Let $A$ be a raw potato. $A$ admits many states e.g. dirty, clean, skinned, ...

2. We want to process $A$ into cooked potato $B$. $B$ admits many states e.g. boiled, fried, deep fried, baked with skin, baked without skin, ...
1. Let $A$ be a raw potato. $A$ admits many *states* e.g. dirty, clean, skinned, ...

2. We want to *process* $A$ into cooked potato $B$. $B$ admits many *states* e.g. boiled, fried, deep fried, baked with skin, baked without skin, ... Let

$$A \xrightarrow{f} B \quad A \xrightarrow{f'} B \quad A \xrightarrow{f''} B$$

be boiling, frying, baking.
1. Let $A$ be a raw potato. $A$ admits many states e.g. dirty, clean, skinned, ...

2. We want to process $A$ into cooked potato $B$. $B$ admits many states e.g. boiled, fried, deep fried, baked with skin, baked without skin, ... Let 

\[ A \xrightarrow{f} B \quad A \xrightarrow{f'} B \quad A \xrightarrow{f''} B \]

be boiling, frying, baking. States are processes 

$I := \text{unspecified} \xrightarrow{\psi} A.$
3. Let

\[ A \xrightarrow{g \circ f} C \]

be the *composite process* of first boiling \( A \xrightarrow{f} B \) and then salting \( B \xrightarrow{g} C \).
3. Let

\[ A \xrightarrow{g \circ f} C \]

be the **composite process** of first boiling \[ A \xrightarrow{f} B \] and then salting \[ B \xrightarrow{g} C \]. Let

\[ X \xrightarrow{1_X} X \]

be doing nothing. We have \[ 1_Y \circ \xi = \xi \circ 1_X = \xi \].
4. Let $A \otimes D$ be potato $A$ and carrot $D$
Let $A \otimes D$ be potato $A$ and carrot $D$ and let

$$A \otimes D \xrightarrow{f \otimes h} B \otimes E$$

be boiling potato while frying carrot.
4. Let $A \otimes D$ be potato $A$ and carrot $D$ and let

$$A \otimes D \xrightarrow{f \otimes h} B \otimes E$$

be boiling potato while frying carrot. Let

$$C \otimes F \xrightarrow{x} M$$

be mashing spice-cook-potato and spice-cook-carrot.
5. Total process:

\[
A \otimes D \xrightarrow{f \otimes h} B \otimes E \xrightarrow{g \otimes k} C \otimes F \xrightarrow{x} M = A \otimes D \xrightarrow{x \circ (g \otimes k) \circ (f \otimes h)} M.
\]
5. Total process:

\[ A \otimes D \xrightarrow{f \otimes h} B \otimes E \xrightarrow{g \otimes k} C \otimes F \xrightarrow{x} M = A \otimes D \xrightarrow{x \circ (g \otimes k) \circ (f \otimes h)} M. \]

6. Recipe = composition structure on processes.
5. Total *process*:

\[ A \otimes D \xrightarrow{f \otimes h} B \otimes E \xrightarrow{g \otimes k} C \otimes F \xrightarrow{x} M = A \otimes D \xrightarrow{x \circ (g \otimes k) \circ (f \otimes h)} M. \]

6. *Recipe* = *composition structure* on *processes*.

7. *Laws governing recipes*:

\[(1_B \otimes g) \circ (f \otimes 1_C) = (f \otimes 1_D) \circ (1_A \otimes g)\]
5. Total process:

\[ A \otimes D \xrightarrow{f \otimes h} B \otimes E \xrightarrow{g \otimes k} C \otimes F \xrightarrow{x} M = A \otimes D \xrightarrow{x \circ (g \otimes k) \circ (f \otimes h)} M. \]

6. Recipe = composition structure on processes.

7. Laws governing recipes:

\[(1_B \otimes g) \circ (f \otimes 1_C) = (f \otimes 1_D) \circ (1_A \otimes g)\]

i.e.

boil potato then fry carrot = fry carrot then boil potato
5. Total process:

\[ A \otimes D \xrightarrow{f \otimes h} B \otimes E \xrightarrow{g \otimes k} C \otimes F \xrightarrow{x} M = A \otimes D \xrightarrow{x \circ (g \otimes k) \circ (f \otimes h)} M. \]

6. **Recipe** = composition structure on processes.

7. Laws governing recipes:

\[(1_B \otimes g) \circ (f \otimes 1_C) = (f \otimes 1_D) \circ (1_A \otimes g)\]

i.e.

boil potato then fry carrot = fry carrot then boil potato

⇒ Symmetric Monoidal Category
Why does a tiger have stripes and a lion doesn’t?
Why does a tiger have stripes and a lion doesn’t?

prey \otimes\text{ predator} \otimes\text{ environment}

\downarrow \text{ hunt}

\text{ dead prey} \otimes\text{ eating predator}


wire and box language

Box := \[f\]

output wire(s)
input wire(s)
--- wire and box language ---

\[ \text{Box} := f \]

Interpretation: wire := system ; box := process
— wire and box language —

\[
Box ::= \begin{array}{c}
\text{output wire(s)} \\
\fbox{f} \\
\text{input wire(s)}
\end{array}
\]

**Interpretation:** wire := system ; box := process

one system: \hspace{1cm} n subsystems: \hspace{1cm} no system:

1 \hspace{1cm} \{ \cdots \} \hspace{1cm} 0
— wire and box games —
wire and box games

sequential or causal or connected composition:

\[ g \circ f \equiv \]

\[ g \]

\[ f \]
--- wire and box games ---

sequential or causal or connected composition:

\[ g \circ f \equiv \]

parallel or acausal or disconnected composition:

\[ f \otimes g \equiv \]
merely a new notation?
merely a new notation?

\[(g \circ f) \otimes (k \circ h) = (g \otimes k) \circ (f \otimes h)\]
merely a new notation?

\[(g \circ f) \otimes (k \circ h) = (g \otimes k) \circ (f \otimes h)\]

peel potato and then fry it, while, clean carrot and then boil it   =   peel potato while clean carrot, and then, fry potato while boil carrot
MINIMAL QUANTUM PROCESS LANGUAGE


[von Neumann 1932] Formalized quantum mechanics in “Mathematische Grundlagen der Quantenmechanik”
[von Neumann 1932] Formalized quantum mechanics in “Mathematische Grundlagen der Quantenmechanik”

[von Neumann to Birkhoff 1935] “I would like to make a confession which may seem immoral: I do not believe absolutely in Hilbert space no more.” (sic)
[von Neumann 1932] Formalized quantum mechanics in “Mathematische Grundlagen der Quantenmechanik”

[von Neumann to Birkhoff 1935] “I would like to make a confession which may seem immoral: I do not believe absolutely in Hilbert space no more.” (sic)

[von Neumann 1932] Formalized quantum mechanics in “Mathematische Grundlagen der Quantenmechanik”

[von Neumann to Birkhoff 1935] “I would like to make a confession which may seem immoral: I do not believe absolutely in Hilbert space no more.” (sic)


[1936 – 2000] many followed them, ...
[von Neumann 1932] Formalized quantum mechanics in “Mathematische Grundlagen der Quantenmechanik”

[von Neumann to Birkhoff 1935] “I would like to make a confession which may seem immoral: I do not believe absolutely in Hilbert space no more.” (sic)


[1936 – 2000] many followed them, ... and FAILED.
the mathematics of it
--- the mathematics of it ---

**Hilber space stuff**: continuum, field structure of complex numbers, vector space over it, inner-product, etc.
Hilber space stuff: continuum, field structure of complex numbers, vector space over it, inner-product, etc.

WHY?
— the mathematics of it —

Hilber space stuff: continuum, field structure of complex numbers, vector space over it, inner-product, etc.

WHY?

von Neumann: only used it since it was ‘available’.
the physics of it
von Neumann crafted Birkhoff-von Neumann Quantum ‘Logic’ to capture the concept of superposition.
von Neumann crafted Birkhoff-von Neumann Quantum ‘Logic’ to capture the concept of superposition.

Schrödinger (1935): the stuff which is the true soul of quantum theory is how quantum systems compose.
von Neumann crafted Birkhoff-von Neumann Quantum ‘Logic’ to capture the concept of superposition.

Schrödinger (1935): the stuff which is the true soul of quantum theory is how quantum systems compose.

Quantum Computer Scientists: Schrödinger is right!
the game plan
Task 0. Solve:

\[
\begin{align*}
tensor \text{ product structure} \\
\text{the other stuff} \\
\end{align*}
= ???
\]
Task 0. Solve:
\[
\begin{align*}
tensor \text{ product structure} & = \text{the other stuff} \\
& = ??? 
\end{align*}
\]
i.e. **axiomatize “⊗” without reference to spaces.**
Task 0. Solve:

\[
\frac{\text{tensor product structure}}{\text{the other stuff}} = ???
\]

i.e. axiomatize “⊗” without reference to spaces.

Task 1. Investigate which assumptions (i.e. which structure) on ⊗ is needed to deduce physical phenomena.
— the game plan —

Task 0. Solve:

\[
\begin{align*}
tensor \text{ product structure} & = \text{ the other stuff} \\
\end{align*}
\]

i.e. **axiomatize “⊗” without reference to spaces.**

Task 1. Investigate which assumptions (i.e. which structure) on \(⊗\) is needed to deduce **physical phenomena.**

Task 2. Investigate whether such an “interaction structure” appear elsewhere in **“our classical reality”**.
Outcome 1a:
Outcome 1a: “Sheer ratio of results to assumptions”

Outcome 1a: “Sheer ratio of results to assumptions” confirms that we are probing something very essential.

Outcome 1a: “Sheer ratio of results to assumptions” confirms that we are probing something very essential.

Outcome 1b: Exposing this structure has already helped to solve open problems elsewhere. (e.g. $2 \times$ ICALP’10)

Outcome 1a: “Sheer ratio of results to assumptions” confirms that we are probing something very essential.

Outcome 1b: Exposing this structure has already helped to solve open problems elsewhere. (e.g. 2× ICALP’10)

Outcome 1c: Framework is a simple intuitive (but rigorous) diagrammatic language,

Outcome 1a: “Sheer ratio of results to assumptions” confirms that we are probing something very essential.

Outcome 1b: Exposing this structure has already helped to solve open problems elsewhere. (e.g. 2× ICALP’10)

Outcome 1c: Framework is a simple intuitive (but rigorous) diagrammatic language, meanwhile adopted by others e.g. Lucien Hardy in arXiv:1005.5164:

“... we join the quantum picturalism revolution [1]”

Outcome 2a:

Behaviors of matter *(Abramsky-C; LiCS’04, quant-ph/0402130)*:

Meaning in language *(Clark-C-Sadrzadeh; Linguistic Analysis, arXiv:1003.4394)*:

Knowledge updating *(C-Spekkens; Synthese, arXiv:1102.2368)*:
BOXES AND WIRES II
quantitative metric

\[ f : A \rightarrow B \]
quantitative metric

\[ f^\dagger : B \rightarrow A \]
asserting (pure) entanglement

\[
\frac{\text{quantum}}{\text{classical}} = \frac{\text{\(\rightarrow\)}}{\text{\(\rightarrow\) \(\rightarrow\) \(\rightarrow\)}}
\]
asserting (pure) entanglement

\[ \frac{\text{quantum}}{\text{classical}} = \quad = \quad \]

⇒ introduce ‘parallel wire’ between systems:

subject to: only topology matters!
E.g.
Transpose:

\[ f \mapsto f^T = \begin{pmatrix} f & \vdots \\ \vdots & \ddots \\ \vdots & & f \end{pmatrix} \]

Conjugate:

\[ f \mapsto f^* = \begin{pmatrix} f^* & \vdots \\ \vdots & \ddots \\ \vdots & & f^* \end{pmatrix} \]
classical data flow?
classical data flow?

ALICE

BOB

⇒ quantum teleportation
symbolically: dagger compact categories

Thm. [Kelly-Laplaza ’80; Selinger ’05] An equational statement between expressions in dagger compact categorical language holds if and only if it is derivable in the graphical notation via homotopy.

Thm. [Hasegawa-Hofmann-Plotkin; Selinger ’08] An equational statement between expressions in dagger compact categorical language holds if and only if it is derivable in the dagger compact category of finite dimensional Hilbert spaces, linear maps, tensor product and adjoints.
Contest in problem solving between:

- Children using quantum picturalism
- Physics teachers using ordinary QM

I expect the children to win!

A SLIGHTLY DIFFERENT LANGUAGE FOR NATURAL LANGUAGE MEANING


the from-words-to-a-sentence process
Consider meanings of **words**, e.g. as vectors (cf. Google):
What is the meaning the *sentence* made up of these?

--- the from-words-to-a-sentence process ---
I.e. how do we/machines produce meanings of **sentences**?
I.e. how do we/machines produce meanings of sentences?

Information flow within a verb:

object → verb → subject
Information flow within a verb:

Again we have:

\[ \text{subject} \begin{array}{c} \longrightarrow \text{verb} \end{array} \text{object} \]
For noun type $n$, verb type is $^{-1}(n) \cdot s \cdot (n)^{-1}$, so:
For noun type $n$, verb type is $^{-1}(n) \cdot s \cdot (n)^{-1}$, so:

\[ n \cdot ^{-1}(n) \cdot s \cdot (n)^{-1} \cdot n = s \]
For noun type $n$, verb type is $^{-1}(n) \cdot s \cdot (n)^{-1}$, so:

$$n \cdot ^{-1}(n) \cdot s \cdot (n)^{-1} \cdot n = s$$

**Diagrammatic typing:**

```
  n                   n
   \             /     \           /   
   \          /       \         /     
   \        /   (n)^{-1}   \   /       
   \      /                \  /         
   \    /                    \ /          
   \   /                      \/           
   \ /                        s             
(n)                        (n)^{-1}       
```
For noun type $n$, verb type is $^{-1}(n) \cdot s \cdot (n)^{-1}$, so:

$$n \cdot ^{-1}(n) \cdot s \cdot (n)^{-1} \cdot n = s$$

Diagrammatic meaning:
Alice does not like Bob
Alice does not like Bob

grammare

meaning vectors of words
$\overrightarrow{Alice} \otimes \overrightarrow{does} \otimes \overrightarrow{not} \otimes \overrightarrow{like} \otimes \overrightarrow{Bob}$

meaning vectors of words

grammar
\(\overrightarrow{\text{Alice}} \otimes \overrightarrow{\text{does}} \otimes \overrightarrow{\text{not}} \otimes \overrightarrow{\text{like}} \otimes \overrightarrow{\text{Bob}}\)
$\overrightarrow{Alice} \otimes \overrightarrow{does} \otimes \overrightarrow{not} \otimes \overrightarrow{like} \otimes \overrightarrow{Bob}$

grammar

meaning vectors of words
Alice does not like Bob

Using:

\[ \text{like} = \bigcap \text{like} \quad \bigcup \text{like} = \text{like} \]
Alice \otimes \text{does} \otimes \text{not} \otimes \text{like} \otimes \overrightarrow{Bob}

= \overrightarrow{g} \left( \overrightarrow{f}(\overrightarrow{x}, \overrightarrow{y}) \right)
— *experiment: word disambiguation* —

E.g. what is “saw” in: “Alice saw Bob with a saw”.

<table>
<thead>
<tr>
<th>Model</th>
<th>High</th>
<th>Low</th>
<th>ρ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.47</td>
<td>0.44</td>
<td>0.16</td>
</tr>
<tr>
<td>Add</td>
<td>0.90</td>
<td>0.90</td>
<td>0.05</td>
</tr>
<tr>
<td>Multiply</td>
<td>0.67</td>
<td>0.59</td>
<td>0.17</td>
</tr>
<tr>
<td>Categorical (1)</td>
<td>0.73</td>
<td>0.72</td>
<td>0.21</td>
</tr>
<tr>
<td>Categorical (2)</td>
<td>0.34</td>
<td>0.26</td>
<td>0.28</td>
</tr>
<tr>
<td>UpperBound</td>
<td>4.80</td>
<td>2.49</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Edward Grefenstette & Mehrnoosh Sadrzadeh (2011) *Experimental support for a categorical compositional distributional model of meaning.* Accepted for: Empirical Methods in Natural Language Processing (EMNLP’11).
UNIVERSAL QUANTUM REASONING
New results on resource requirements, complexity of translations in MBQC (Duncan-Perdrix ICALP’10):

Example 18. The ubiquitous CNOT operation can be computed by the pattern

\[ P = X_3^4 Z_2^4 Z_1^2 M_0^3 M_0^2 E_1^3 E_2^3 E_3^4 N_3^3 N_4 \]

This yields the diagram,

\[ D_P = H H H \pi, \{2\} \pi, \{3\} \]

where each qubit is represented by a vertical "path" from top to bottom, with qubit 1 the leftmost, and qubit 4 is the rightmost.

By virtue of the soundness of \( R \) and Proposition 10, if \( D_P \) can be rewritten to a circuit-like diagram without any conditional operations, then the rewrite sequence constitutes a proof that the pattern computes the same operation as the derived circuit.

Example 19. Returning to the CNOT pattern of Example 18, there is a rewrite sequence, the key steps of which are shown below, which reduces the \( D_P \) to the unconditional circuit-like pattern for CNOT introduced in Example 7. This proves two things: firstly that \( P \) indeed computes the CNOT unitary, and that the pattern \( P \) is deterministic.

Similar stuff for TMBQC (Clare Horsman NJP’11):
automated theory exploration

Lucas Dixon (Google), Ross Duncan (Brussels), Aleks Kissinger, Alex Merry (Oxf) and Matvey Soloviev (Camb) — http://sites.google.com/site/quantomatic/
WIRES?
SPIDERS!
'spiders' = \{ \begin{array}{c}
m \\
n \end{array} \} \quad \text{such that, for } k > 0:

\begin{array}{c}
m + m' - k \\
n + n' - k \end{array} = \begin{array}{c}
m \\
n \end{array}

such that, for $k > 0$:
— spiders —

\[
\text{‘cups/caps’} = \left\{ \begin{array} {c}
m \\
n \end{array} \right\}
\]

such that, for \( k > 0 \):

\[
\begin{array} {c}
m + m' - k \\
n + n' - k
\end{array}
\]

\[
= \begin{array} {c}
m \\
n 
\end{array}
\]
complementary spiders
Thm.

--- complementary spiders ---
