picture logic for info-flow in language and physics
(incl. the Lambek vs. Lambek battle)

That flowin phyling — October 2010

Bob Coecke
Oxford University Computing Laboratory
Context:

Context:


Context:


Context:


This talk:

Survey of aspects of categories and graphical language:

• What is a monoidal category
• Connection to categories
• Known completeness results
This talk:
Survey of aspects of categories and graphical language:

- What is a monoidal category
- Connection to categories
- Known completeness results

Pictures of quantum state information flow

Pictures of linguistic meaning information flow
This talk:

Survey of aspects of categories and graphical language:
- What is a monoidal category
- Connection to categories
- Known completeness results

Pictures of **quantum state** information flow

Pictures of **linguistic meaning** information flow

Differences between these two

Flexibility of these two
WHY ‘MONOIDAL’ CATEGORIES?
BECAUSE THEY 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 $D$
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.
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 \textit{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 \textit{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. \textit{Recipe} = \textit{composition structure} on \textit{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. Law________________________:
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. Law governing recipes:

\[ (1_B \otimes g) \circ (f \otimes 1_C) = (f \otimes 1_D) \circ (1_A \otimes g) \]
5. Total \textit{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. \textit{Recipe} = \textit{composition structure} on \textit{processes}.

7. \textit{Law 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
Very successful in **proof theory** and **programming**:

<table>
<thead>
<tr>
<th>proof theory</th>
<th>programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositions</td>
<td>Data Types</td>
</tr>
<tr>
<td>Proofs</td>
<td>Programs</td>
</tr>
</tbody>
</table>

BLUE = systems
Red = processes
Very successful in **proof theory** and **programming**:

<table>
<thead>
<tr>
<th>proof theory</th>
<th>programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositions</td>
<td>Data Types</td>
</tr>
<tr>
<td>Proofs</td>
<td>Programs</td>
</tr>
</tbody>
</table>

BLUE = systems  
Red = processes

but also applies to:

<table>
<thead>
<tr>
<th>biology</th>
<th>physics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biological syst.</td>
<td>Physical syst.</td>
</tr>
<tr>
<td>Biological proc</td>
<td>Physical proc.</td>
</tr>
</tbody>
</table>
Very successful in **proof theory** and **programming**:

<table>
<thead>
<tr>
<th>proof theory</th>
<th>programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositions</td>
<td>Data Types</td>
</tr>
<tr>
<td>Proofs</td>
<td>Programs</td>
</tr>
</tbody>
</table>

BLUE = systems  
Red = processes

but also applies to:

<table>
<thead>
<tr>
<th>biology</th>
<th>physics</th>
<th>language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biological syst.</td>
<td>Physical syst.</td>
<td>language syst.</td>
</tr>
<tr>
<td>Biological proc</td>
<td>Physical proc.</td>
<td>language proc.</td>
</tr>
</tbody>
</table>
A MINIMAL LANGUAGE FOR QUANTUM REASONING


(physical) data in the language

Systems:

\[ A \quad B \quad C \]

Processes:

\[ A \xrightarrow{f} A \quad A \xrightarrow{g} B \quad B \xrightarrow{h} C \]

Compound systems:

\[ A \otimes B \quad I \quad A \otimes C \xrightarrow{f \otimes g} B \otimes D \]

Temporal composition:

\[ A \xrightarrow{h \circ g} C \quad := \quad A \xrightarrow{g} B \xrightarrow{h} C \quad A \xrightarrow{1_A} A \]

merely a new notation?

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

\[(g \circ f) \otimes (k \circ h) = (g \otimes k) \circ (f \otimes h)\]
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
\[
\begin{align*}
\psi &: I \to A \\
\pi &: A \to I \\
\pi \circ \psi &: I \to I
\end{align*}
\]
\[
\psi : I \rightarrow A \quad \pi : A \rightarrow I \quad \pi \circ \psi : I \rightarrow I
\]
\( \psi : I \rightarrow A \) \hspace{1cm} \( \pi : A \rightarrow I \) \hspace{1cm} \( \pi \circ \psi : I \rightarrow I \)
$\psi : I \to A$

$\pi : A \to I$

$\pi \circ \psi : I \to I$

--- graphical notation ---
\[ \psi : I \rightarrow A \quad \pi : A \rightarrow I \quad \pi \circ \psi : I \rightarrow I \]
$\psi : I \to A$

$\pi : A \to I$

$\pi \circ \psi : I \to I$
ψ : I → A

π : A → I

π ◦ ψ : I → I

--- graphical notation ---

\[
\begin{align*}
\psi &: I \to A \\
\pi &: A \to I \\
\pi \circ \psi &: I \to I
\end{align*}
\]
\[ \psi : I \rightarrow A \quad \pi : A \rightarrow I \quad \pi \circ \psi : I \rightarrow I \]
\[ \psi : I \rightarrow A \quad \pi : A \rightarrow I \quad \pi \circ \psi : I \rightarrow I \]
$\psi : I \rightarrow A$  \hspace{1cm} $\pi : A \rightarrow I$  \hspace{1cm} $\pi \circ \psi : I \rightarrow I$

--- graphical notation ---
adjoint

\[ f : A \rightarrow B \]
adjoint

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

\[
\frac{\text{quantum}}{\text{classical}} = \quad \neq \quad =
\]
quantum-like
quantum-like

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

\[
\begin{array}{c}
\text{A} \\
\text{A} \\
\end{array}
\quad \text{=}
\quad
\begin{array}{c}
\text{A} \\
\text{A} \\
\end{array}
\]
quantum-like
quantum-like
quantum-like

\[ \begin{align*}
A & \quad A \\
\downarrow & \quad \downarrow \\
\text{Red triangle} & = \\
\downarrow & \quad \downarrow \\
A & \quad A
\end{align*} \]
--- quantum-like ---

\[
\begin{align*}
\text{A} & \quad \text{A} \\
\text{U} & \\
\text{A} & \quad \text{A} \\
\text{A} & \quad \text{A}
\end{align*}
\]
quantum-like

\[ f = \uparrow \downarrow \]
In QM: cups = Bell-states, caps = Bell-effects, $\pi$-rotations = transpose
classical data flow?
classical data flow?
classical data flow?

\[ \text{ALICE} \xrightarrow{f} \text{BOB} \]

\[ \text{ALICE} \xrightarrow{f} \text{BOB} \]

\[ \xRightarrow{=} \]

\[ \Rightarrow \] quantum teleportation
Applying "decorated" normalization

\[ f = f \Rightarrow \text{Entanglement swapping} \]
classical data flow?

\[
f = f \rightarrow g
\]

\[
\Rightarrow \text{gate teleportation computation}
\]
dagger compact symmetric monoidal categories
Thm. [Selinger ’05] An equational statement between expressions in dagger compact symmetric monoidal categorical language holds if and only if it is derivable in the graphical notation via homotopy.
Thm. [Selinger ’05] An equational statement between expressions in dagger compact symmetric monoidal categorical language holds if and only if it is derivable in the graphical notation via homotopy.

Thm. [Selinger ’08] An equational statement between expressions in dagger compact symmetric monoidal categorical language holds if and only if it is derivable in the category of finite dimensional Hilbert spaces, linear maps, tensor product, and adjoints.
In words: Any equation involving:

- states, operations, effects
- unitarity, adjoints (e.g. self-adjoint), projections
- Bell-states/effects, transpose, conjugation
- inner-product, trace, Hilbert-Schmidt norm
- positivity, completely positive maps, ...

holds in quantum theory if and only if it can be derived in the graphical language via homotopy.
bases and phases
bases and phases

\[
\begin{align*}
\alpha & \quad \beta & \quad \gamma \\
& \quad H & \quad H \\
& \quad H & \quad H \\
& \quad H & \quad H
\end{align*}
\]
bases and phases
--- bases and phases ---

⇒ One-way quantum computing
A SLIGHTLY DIFFERENT LANGUAGE FOR NATURAL LANGUAGE MEANING

arXiv:1003.4394
Consider meaning triangle (e.g. vectors cf. Google) of words:

How do we/machines compute meaning of sentences?
Consider meaning triangle (e.g. vectors cf. Google) of words:

How do we/machines compute meaning of sentences?
Information flow within a verb:

Object \(\rightarrow\) Verb \(\rightarrow\) Subject

--- the from-words-to-a-sentence process ---

--- the from-words-to-a-sentence process ---
Information flow within a verb:

Again we have:
— going non-symmetric —

\[
I \xrightarrow{\eta^l} A \otimes A^l \quad A^l \otimes A \xrightarrow{\epsilon^l} I \quad I \xrightarrow{\eta^r} A^r \otimes A \quad A \otimes A^r \xrightarrow{\epsilon^r} I
\]
Vector spaces and linear maps (⊇ vectors) form CC

Meaning:
- vector spaces $V, W$
- linear maps $f: V \to W$
- tensor product $V \otimes W$ with unit $\mathbb{R}$
- $V^l = V^r = V$
- caps:
  $$\epsilon^l = \epsilon^r : V \otimes V \to \mathbb{R} :: \sum_{ij} c_{ij} s_i \otimes s_j \mapsto \sum_{ij} c_{ij} \langle s_i | s_j \rangle$$
- cups:
  $$\eta^l = \eta^r : \mathbb{R} \to V \otimes V :: 1 \mapsto \sum_{i} e_i \otimes e_i$$
Grammatical type calculus forms CC

Grammar = Pregroup:
- types $p, q$ and type reductions $p \leq q$
- concatenation $pq$ with unit is 1:
  \[ p \leq q \Rightarrow rp \leq rq, pr \leq qr \]
- left/right:
  \[ p \leq q \Rightarrow q^l \leq p^l, q^r \leq p^r \quad (pq)^l = q^lp^l \quad (pq)^r = q^rp^r \]
- caps and cups:
  \[ \epsilon^r = [pp^r \leq 1] \quad \epsilon^l = [p^l p \leq 1] \quad \eta^r = [1 \leq p^rp] \quad \eta^l = [1 \leq pp^l] \]
- yanking:
  \[ p^lp \leq 1 \leq pp^l \quad pp^r \leq 1 \leq p^rp \]
The categorical product of CC’s is a CC

- objects \((p, V)\) and morphisms \((p \leq q, f : V \to W)\)
- all operations are component-wise
The categorical product of CC’s is a CC

- objects \((p, V)\) and morphisms \((p \leq q, f : V \to W)\)
- all operations are component-wise

⇒ a CC that combines grammar and meaning

The grammar will control in which the meaning of words interact to make up the meaning of the sentence.

⇒ ‘grammatical’ quantum field theory
Alice $\otimes$ does $\otimes$ not $\otimes$ like $\otimes$ Bob

meaning vectors of words

grammar
Alice $\otimes$ does $\otimes$ not $\otimes$ like $\otimes$ Bob

meaning vectors of words

grammar
Alice does not like Bob

meaning vectors of words

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

meaning vectors of words

grammar
ANALOGIES & CONTRASTS
Alice hates Bob
meaning vectors of words
grammar

measurements

states
— analogy: “non-local” info-flows —

English (& French):

Hindi:

Persian:

Arabic (and Hebrew):

— methodological analogy —

**Physical theories**

- Interaction of quantum systems
- Interaction of words in sentences
- High-level interaction structures

**Linguistic theories**

- Language experiments: Querying texts, www and large data bases

**Physical experiments:**

- Probing systems with measurement devices
<table>
<thead>
<tr>
<th><strong>Quantum-flow</strong></th>
<th><strong>Meaning-flow</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>States</td>
<td>Meanings</td>
</tr>
<tr>
<td>Measurement patterns</td>
<td>Grammatical structure</td>
</tr>
<tr>
<td>Symmetric</td>
<td>Non-symmetric</td>
</tr>
<tr>
<td>Phases</td>
<td>Seemingly non counterpart</td>
</tr>
<tr>
<td>Bases</td>
<td>“that” (other logical stuff?)</td>
</tr>
<tr>
<td>Mixedness</td>
<td>Steve C. told me something</td>
</tr>
</tbody>
</table>
a bit on STRUCTURE REQUIREMENTS

(the Lambek vs. Lambek battle)
Does Alice hate associativity?
weak distributivity

Alice

Hate

Does

f

weak distributivity

f
Grammatical graphical components:

<table>
<thead>
<tr>
<th></th>
<th>unary - compact</th>
<th>binary - closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>type reduction</td>
<td>$a \circ a^\dagger \leq e$</td>
<td>$a \cdot (a \rightarrow c) \leq c$</td>
</tr>
<tr>
<td>type introduction</td>
<td>$e \leq a^\dagger \circ a$</td>
<td>$c \leq a \rightarrow (a \cdot c)$</td>
</tr>
</tbody>
</table>

All seem to do the job:

<table>
<thead>
<tr>
<th>intro</th>
<th>red</th>
<th>none</th>
<th>closed</th>
<th>compact</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td></td>
<td></td>
<td></td>
<td>AB-pom</td>
</tr>
<tr>
<td>closed</td>
<td></td>
<td>resid. pom</td>
<td></td>
<td>protogroup</td>
</tr>
<tr>
<td>compact</td>
<td></td>
<td></td>
<td></td>
<td>Grishin pom, pregroup</td>
</tr>
</tbody>
</table>

Inclusive non-associative variants it seems, ...
(limited) State of art:

Selinger’s Chapter in New Struc. Phys.:
  • Rigorous stuff in categories $\simeq$ graphical languages

Baez-Stay Chapter in New Struc. Phys.:
  • ‘Info-flow ideas’ on non-CC closed $\otimes$ cats
