String diagrams

a functorial semantics of proofs and programs

Paul-André Melliès

CNRS, Université Paris Denis Diderot

Hopf-in-Lux

Luxembourg 17 July 2009
Proof-knots

Aim: formulate an algebra of these logical knots
The purpose of semantics

A mathematical study of programming languages and compilation schemes.

Both functional and imperative languages, based on a kernel of $\lambda$-calculus:

- PCF
- $\lambda$-calcul
- higher order
- typing
- recursion
- Algol
- states
- ML
- exceptions
- references
- OCAML
- modules
- objects
- JAVA
- concurrency
- synchronization
- threads

Languages designed between academic and industrial circles, between semantics (denotational or operational) and implementations.
The purpose of semantics

A mathematical theory of programming languages from conception to compilation.

Advanced programming language
  ↓
  compilation
  ↓
Elementary commands of the machine

A prerequisite to the complete verification of a given implementation.
# The simply-typed $\lambda$-calculus

<table>
<thead>
<tr>
<th>Rule</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable</strong></td>
<td>$x : X \vdash x : X$</td>
</tr>
<tr>
<td><strong>Abstraction</strong></td>
<td>$\Gamma, x : A \vdash P : B$ \hspace{1cm} $\Gamma \vdash \lambda x. P : A \Rightarrow B$</td>
</tr>
<tr>
<td><strong>Application</strong></td>
<td>$\Gamma \vdash P : A \Rightarrow B$ \hspace{1cm} $\Delta \vdash Q : A$ \hspace{1cm} $\Gamma, \Delta \vdash PQ : B$</td>
</tr>
<tr>
<td><strong>Weakening</strong></td>
<td>$\Gamma \vdash P : B$ \hspace{1cm} $\Gamma, x : A \vdash P : B$</td>
</tr>
<tr>
<td><strong>Contraction</strong></td>
<td>$\Gamma, x : A, y : A \vdash P : B$ \hspace{1cm} $\Gamma, z : A \vdash P[x, y \leftarrow z] : B$</td>
</tr>
<tr>
<td><strong>Permutation</strong></td>
<td>$\Gamma, x : A, y : B, \Delta \vdash P : C$ \hspace{1cm} $\Gamma, y : B, x : A, \Delta \vdash P : C$</td>
</tr>
</tbody>
</table>
Key idea

Compilation itself is the ultimate semantics!

Advanced programming language
  ↓ semantics
  Elementary transition systems

There remains to describe the grammar of an idealized compilation.
Starting point: game semantics

Every proof of formula $A$ initiates a dialogue where

Proponent tries to convince Opponent

Opponent tries to refute Proponent

An interactive approach to logic and programming languages
The formal proof of the drinker’s formula

\[
\begin{align*}
\frac{A(x_0) \vdash A(x_0)}{A(x_0) \vdash A(x_0), \forall x. A(x)} & \quad \text{Axiom} \\
\frac{A(x_0) \vdash A(x_0), \forall x. A(x)}{\vdash A(x_0), A(x_0) \Rightarrow \forall x. A(x)} & \quad \text{Right Weakening} \\
\frac{\vdash A(x_0), \forall x. A(x) \Rightarrow \exists y. \{A(y) \Rightarrow \forall x. A(x)\}}{\vdash \exists y. \{A(y) \Rightarrow \forall x. A(x)\}} & \quad \text{Right } \exists \\
\frac{\vdash \forall x. A(x), \exists y. \{A(y) \Rightarrow \forall x. A(x)\}}{\vdash A(y_0) \Rightarrow \forall x. A(x), \exists y. \{A(y) \Rightarrow \forall x. A(x)\}} & \quad \text{Right } \forall \\
\frac{\vdash A(y_0) \Rightarrow \forall x. A(x), \exists y. \{A(y) \Rightarrow \forall x. A(x)\}}{\vdash \exists y. \{A(y) \Rightarrow \forall x. A(x)\}} & \quad \text{Left Weakening} \\
\frac{\vdash \exists y. \{A(y) \Rightarrow \forall x. A(x)\}, \exists y. \{A(y) \Rightarrow \forall x. A(x)\}}{\vdash \exists y. \{A(y) \Rightarrow \forall x. A(x)\}} & \quad \text{Right } \exists \\
\frac{\vdash \exists y. \{A(y) \Rightarrow \forall x. A(x)\}}{\vdash \exists y. \{A(y) \Rightarrow \forall x. A(x)\}} & \quad \text{Contraction}
\end{align*}
\]
Game semantics

Games typically defined as 2-player decision trees
Sequential game semantics

A proof $\pi$

alternating sequences of moves

A proof $\pi$

Game semantics: an *interleaving* semantics of proofs.
Sequential games

A sequential game \((M, P, \lambda)\) consists of

- \(M\) a set of moves,
- \(P \subseteq M^*\) a set of plays,
- \(\lambda : M \rightarrow \{-1, +1\}\) a polarity function on moves.

Alternatively, a sequential game is a polarized decision tree.
Strategies

A strategy $\sigma$ is a set of alternating plays of even-length

$$s = m_1 \cdots m_{2k}$$

starting by an Opponent move such that:

— $\sigma$ contains the empty play,

— $\sigma$ is closed by even-length prefix:

$$\forall s, \forall m, n \in M, \quad s \cdot m \cdot n \in \sigma \Rightarrow s \in \sigma$$

— $\sigma$ is deterministic:

$$\forall s \in \sigma, \forall m, n_1, n_2 \in M, \quad s \cdot m \cdot n_1 \in \sigma \text{ and } s \cdot m \cdot n_2 \in \sigma \Rightarrow n_1 = n_2.$$
An interleaving semantics

The boolean game $\mathcal{B}$:

![Diagram of the boolean game $\mathcal{B}$]

Player in red
Opponent in blue
Three algebraic operations on sequential games

- the negation \( \neg A \)
- the tensor \( A \otimes B \)
- the sum \( A \oplus B \)

The same algebraic structure as in linear algebra!
Negation

Proponent Program plays the game $A$

Opponent Environment plays the game $\neg A$

Negation permutes the rôles of Proponent and Opponent
Negation

Opponent
Environment

plays the game

\[ \neg A \]

Proponent
Program

plays the game

A

Negation permutes the rôles of Opponent and Proponent
Tensor product

Play the two games in parallel
Sum

Proponent selects one component
A category $\mathcal{G}$ of games and strategies

– its objects $A, B$ are the sequential games

– its morphisms

$$A \rightarrow B$$

are the strategies playing on

$$\rightarrow A \otimes B$$
A typical strategy
Composition of strategies

The composite strategy

\[ A \rightarrow C \]

of the two strategies

\[ A \xrightarrow{\sigma} B \quad \quad B \xrightarrow{\tau} C \]

is obtained by letting the strategies \( \sigma \) and \( \tau \) interact on the game \( B \).

Describes the evaluation of a program and its procedure
A typical composition
The identity strategy

The identity strategy

\[ A \rightarrow A \]

is defined as the copycat strategy on the game

\[ \neg A \otimes A \]

Clarifies the logical principle that: \( A \) or not \( A \)
An interleaving semantics

The tensor product of two boolean games $B_1$ et $B_2$:
A step towards true concurrency: bend the branches!
Asynchronous games: tile the diagram!
Asynchronous game semantics

A proof $\pi_1$

trajectories in asynchronous transition spaces

A proof $\pi_2$

Main result: innocent strategies are positional.
Illustration: the strategy (true $\otimes$ false)

Strategies seen as closure operators on complete lattices
Asynchronous games

There are two sequential implementations of the conjunction operator $\land$.

Left implementation

Right implementation
Asynchronous games

The game associated to $\mathbb{B} \otimes \mathbb{B} \rightarrow \mathbb{B}$ looks like a flower with eight petals

one for each terminal position $(input_L \otimes input_R) \rightarrow output$ of the game.
Asynchronous games

Left implementation

\[
\text{true}_R \\
\text{question}_R \\
\text{true}_L \\
\text{question}_L \\
\text{question}
\]
Asynchronous games

Right implementation

true

true_L

question_L

true_R

question_R

question
Asynchronous games

\[
\begin{align*}
\text{true} & \quad \text{true}_R & \quad \text{true}_L \\
\text{question}_R & \quad \sim & \quad \text{question}_L \\
\sim & \quad \sim & \quad \sim \\
\text{true}_L & \quad \sim & \quad \text{true}_R \\
\text{question}_L & \quad \sim & \quad \text{question}_R \\
\text{question} & & \\
\end{align*}
\]
Logical dualities in categories

From orders to categories
Duality in a boolean algebra

Negation defines a bijection

between the boolean algebra $B$ and its opposite boolean algebra $B^{op}$.
Duality in a category

Negation should define an equivalence

between a cartesian closed category $\mathcal{C}$ and its opposite category $\mathcal{C}^{\text{op}}$. 
However, in a cartesian closed category...

Suppose that the category has an initial object 0. Then,

Every morphism $f : A \rightarrow 0$ is an isomorphism.

Its inverse is the unique map $h$ making the diagram commute:
In a self-dual cartesian closed category...

\[
\text{Hom}( A, B ) \cong \text{Hom}( A \times 1, B ) \\
\cong \text{Hom}( 1, A \Rightarrow B ) \\
\cong \text{Hom}( \neg(A \Rightarrow B), \neg1 ) \\
\cong \text{Hom}( \neg(A \Rightarrow B), 0 )
\]

Hence, every such self-dual category \( \mathcal{C} \) is a preorder!
Duality in a heyting algebra

Every object \( \bot \) defines an **adjunction**

\[
\begin{array}{ccc}
H & \bot & H^{op} \\
\text{negation} & \text{negation} & \\
\end{array}
\]

between the heyting algebra \( H \) and its opposite algebra \( H^{op} \).

\[
a \leq_H \bot \circ b \iff b \leq_H a \circ \bot \iff a \circ \bot \leq_{H^{op}} b
\]
Duality in a dialogue category

Negation defines an adjunction

\[ A(a, \bot \rightarrow b) \cong A(b, a \rightarrow \bot) \cong A^{\text{op}}(a \rightarrow \bot, b) \]

between the dialogue category \( \mathcal{C} \) and its opposite category \( \mathcal{C}^{\text{op}} \).
String Diagrams

A notation by Roger Penrose (1970)
Monoidal Categories

A monoidal category is a category \( \mathcal{C} \) equipped with a functor:
\[
\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}
\]
an object:
\[
I
\]
and three natural transformations:
\[
\begin{align*}
(A \otimes B) \otimes C & \xrightarrow{\alpha} A \otimes (B \otimes C) \\
I \otimes A & \xrightarrow{\lambda} A \\
A \otimes I & \xrightarrow{\rho} A
\end{align*}
\]
satisfying a series of coherence properties.
String Diagrams

A morphism \( f : A \otimes B \otimes C \longrightarrow D \otimes E \) is depicted as:
Composition

The morphism $A \xrightarrow{f} B \xrightarrow{g} C$ is depicted as

$$g \circ f = \begin{array}{c} C \\ \downarrow \quad \downarrow \\ A \end{array} \quad = \quad \begin{array}{c} B \\ \downarrow \\ C \end{array}$$

Vertical composition
Tensor product

The morphism \((A \xrightarrow{f} B) \otimes (C \xrightarrow{g} D)\) is depicted as

\[
\begin{aligned}
B \otimes D & \quad = \quad B \otimes C \\
A \otimes C & \quad = \quad A \otimes D
\end{aligned}
\]

Horizontal tensor product
Example

\[ f \otimes \text{id}_D \]
**Example**

\[(f \otimes \text{id}_D) \circ (\text{id}_A \otimes g)\]
Example

\[(id_B \otimes g) \circ (f \otimes id_C)\]
Meaning preserved by deformation

\[(f \otimes id_D) \circ (id_A \otimes g) = (id_B \otimes g) \circ (f \otimes id_C)\]
Functorial boxes

\[ F : \mathcal{C} \rightarrow \mathcal{D} \]

A window on the blue \( \mathcal{C} \) inside the ambiant \( \mathcal{D} \).
Functorial equalities

\[
\begin{array}{c}
\begin{array}{ccc}
FC & C & FB \\
\circ g & \circ f & \circ f \\
B & A & FA \\
F & F & FA
\end{array}
& = & \\
\begin{array}{ccc}
FC & C & FB \\
\circ g & \circ f & \circ f \\
B & A & FA \\
F & F & FA
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{ccc}
FA & A & FA \\
\circ F & \circ f & \circ f \\
A & A & FA \\
F & F & FA
\end{array}
& = & \\
\begin{array}{ccc}
FA & A & FA \\
\circ F & \circ f & \circ f \\
A & A & FA \\
F & F & FA
\end{array}
\end{array}
\]
Lax monoidal functor

A lax monoidal functor is a functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ equipped with morphisms

$$m_{[A,B]} : FA \otimes FB \longrightarrow F(A \otimes B)$$

$$m_{[-]} : I \longrightarrow FI$$

satisfying a series of coherence relations.

A strong monoidal functor is lax monoidal with invertible coercions.
The purpose of coercions

\[ F(A_1 \otimes A_2 \otimes A_3) \]

\[ A_1 \otimes A_2 \otimes A_3 \]

\[ m[A_1, A_2, A_3] \]

\[ m[-] \]
Lax monoidal functor

A lax monoidal functor is a box with many inputs - one output.

\[ F(f) \circ m_{[A_1, \ldots, A_k]} : FA_1 \otimes \cdots \otimes FA_k \longrightarrow FB \]
Functorial equalities (on lax functors)
Strong monoidal functors

A strong monoidal functor is a box with many inputs - many outputs
Functorial equalities (on strong functors)
Functorial equalities (on strong functors)
Natural transformations

A natural transformation

\[ \theta : F \rightarrow G : \mathcal{C} \rightarrow \mathcal{D} \]

satisfies the pictorial equality:
Monoidal natural transformations

A **monoidal** natural transformation

\[ \theta : F \rightarrow G : \mathcal{C} \rightarrow \mathcal{D} \]

satisfies the pictorial equality:
The algebraic nature of knots

Functorial knot theory in ribbon categories
Braided categories

A monoidal category equipped with braid maps

\[ A \otimes B \xrightarrow{\gamma_{A,B}} B \otimes A \]

\[ B \otimes A \xrightarrow{\gamma_{A,B}^{-1}} A \otimes B \]
Coherence diagram for braids
Topological deformation in string diagrams
Coherence diagram for braids
Topological deformation in string diagrams
Duality

A duality $A ightarrow B$ is a pair of morphisms

$I \xrightarrow{\eta} A \otimes B$

$B \otimes A \xrightarrow{\epsilon} I$
Duality

satisfying the two “zig-zag” equalities:

\[
\begin{align*}
\begin{array}{c}
A \\
\end{array} & = \\
\begin{array}{c}
B \\
\end{array} \\
\begin{array}{c}
A \\
\end{array} & = \\
\begin{array}{c}
B \\
\end{array}
\end{align*}
\]

In that case, \( A \) is called a \textbf{left dual} of \( B \).
Ribbon category

A braided category in which every object $A$ has a right dual $A^*$, satisfying:
Knot invariants

Every ribbon category $\mathcal{D}$ induces a knot invariant

The free ribbon category is a category of framed tangles
Jones polynomial invariant

\[
\frac{2}{x^2} + \frac{1}{x^4} + \frac{y^2}{x^2}
\]

\[
2x^2 - x^4 + x^2y^2
\]

A compositional semantics of knots
The topological nature of negation

At the interface between topology, algebra and logic
Cartesian closed categories

A cartesian category $\mathcal{C}$ is closed when there exists a functor

$$\Rightarrow : \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathcal{C}$$

and a natural bijection

$$\varphi_{A,B,C} : \mathcal{C}(A \times B, C) \cong \mathcal{C}(A, B \Rightarrow C)$$
The free cartesian closed category

The objects of the category \( \text{free-ccc}(\mathcal{C}) \) are the formulas

\[
A, B ::= X \mid A \times B \mid A \Rightarrow B \mid 1
\]

where \( X \) is an object of the category \( \mathcal{C} \).

The morphisms are the simply-typed \( \lambda \)-terms, modulo \( \beta\eta \)-conversion.
**The simply-typed $\lambda$-calculus**

**Variable**

$$x : X \vdash x : X$$

**Abstraction**

$$\Gamma, x : A \vdash P : B$$

$$\Gamma \vdash \lambda x . P : A \Rightarrow B$$

**Application**

$$\Gamma \vdash P : A \Rightarrow B$$

$$\Delta \vdash Q : A$$

$$\Gamma, \Delta \vdash PQ : B$$

**Weakening**

$$\Gamma \vdash P : B$$

$$\Gamma, x : A \vdash P : B$$

**Contraction**

$$\Gamma, x : A, y : A \vdash P : B$$

$$\Gamma, z : A \vdash P[x, y \leftarrow z] : B$$

**Permutation**

$$\Gamma, x : A, y : B, \Delta \vdash P : C$$

$$\Gamma, y : B, x : A, \Delta \vdash P : C$$
Proof invariants

Every ccc $D$ induces a proof invariant $[-]$ modulo execution.

Hence the prejudice that proof theory is intrinsically syntactical...
A symmetric monoidal category $\mathcal{C}$ equipped with a functor

$$\neg : \mathcal{C}^{\text{op}} \rightarrow \mathcal{C}$$

and a natural bijection

$$\varphi_{A,B,C} : \mathcal{C}(A \otimes B, \neg C) \cong \mathcal{C}(A, \neg(\neg \neg) C)$$
The free dialogue category

The objects of the category \( \text{free-dialogue}(C) \) are dialogue games constructed by the grammar

\[
A, B ::= X \mid A \otimes B \mid \neg A \mid 1
\]

where \( X \) is an object of the category \( C \).

The morphisms are total and innocent strategies on dialogue games.

As we will see: proofs are 3-dimensional variants of knots...
A presentation of logic by generators and relations

Negation defines a pair of adjoint functors

\[
\begin{array}{c}
\mathcal{C} \\
\mathcal{C}^{\text{op}}
\end{array}
\]

\[
\begin{array}{c}
\dashv \\
\perp
\end{array}
\]

witnessed by the series of bijection:

\[
\mathcal{C}(A, \neg B) \cong \mathcal{C}(B, \neg A) \cong \mathcal{C}^{\text{op}}(\neg A, B)
\]
The 2-dimensional topology of adjunctions

The **unit** and **counit** of the adjunction $L \dashv R$ are depicted as

$$\eta : \text{Id} \to R \circ L \quad \quad \epsilon : L \circ R \to \text{Id}$$

Opponent move = functor $R$  Proponent move = functor $L$
A typical proof

Reveals the algebraic nature of game semantics
A purely diagrammatic cut elimination
The 2-dimensional dynamic of adjunction

Recovers the usual way to compose strategies in game semantics
Interesting fact

There are just as many canonical proofs

\[
\begin{align*}
\underbrace{\cdots}_{2p} & \quad \vdash \quad \underbrace{\cdots}_{2q} \\
\neg \cdots \neg A & \quad \vdash \quad \neg \cdots \neg A
\end{align*}
\]

as there are maps

\[
[p] \quad \rightarrow \quad [q]
\]

between the ordinals \([p] = \{0 < 1 < \cdots < p - 1\}\) and \([q]\).

This fragment of logic has the same combinatorics as simplices.
The two generators of a monad

Every increasing function is composite of faces and degeneracies:

\[ \eta : \begin{array}{c} 0 \\ \end{array} \rightarrow \begin{array}{c} 1 \\ \end{array} \]

\[ \mu : \begin{array}{c} 2 \\ \end{array} \rightarrow \begin{array}{c} 1 \\ \end{array} \]

Similarly, every proof is composite of the two generators:

\[ \eta : A \rightarrow \neg\neg A \]

\[ \mu : \neg\neg\neg\neg A \rightarrow \neg\neg A \]

The unit and multiplication of the double negation monad
The two generators in sequent calculus

$$
\frac{A \vdash A}{A, \neg A \vdash \neg \neg A} \quad 2
\frac{A \vdash A}{A, \neg A \vdash \neg \neg A} \quad 1
$$

$$
\frac{\neg A \vdash \neg A}{A, \neg A \vdash \neg \neg A} \quad 3
\frac{\neg \neg \neg \neg A, \neg A \vdash \neg \neg \neg \neg A}{\neg \neg \neg \neg A \vdash \neg \neg \neg \neg A} \quad 1
$$
The two generators in string diagrams

The **unit** and **multiplication** of the monad $R \circ L$ are depicted as

\[ \eta : Id \rightarrow R \circ L \quad \mu : R \circ L \circ R \circ L \rightarrow R \circ L \]
Tensor and negation

An atomist approach to proof theory
Four primitive components of logic

[1] the negation  ¬
[2] the linear conjunction  ⊗
[3] the repetition modality  !
[4] the existential quantification  ∃

Logic = Data Structure + Duality
Tensor vs. negation

A well-known fact: the continuation monad is strong

\[ (\neg\neg A) \otimes B \rightarrow \neg\neg (A \otimes B) \]

The starting point of the algebraic theory of side effects
Tensor vs. negation

Proofs are generated by a parametric strength

$$\kappa_X : \neg(X \otimes \neg A) \otimes B \to \neg(X \otimes \neg(A \otimes B))$$

which generalizes the usual notion of strong monad:

$$\kappa : \neg\neg A \otimes B \to \neg\neg(A \otimes B)$$
Proofs as 3-dimensional string diagrams

The left-to-right proof of the sequent

\[ \neg\neg A \otimes \neg\neg B \vdash \neg\neg (A \otimes B) \]

is depicted as
Tensor vs. negation: conjunctive strength

Linear distributivity in a continuation framework
Tensor vs. negation: disjunctive strength

Linear distributivity in a continuation framework
A factorization theorem

The four proofs $\eta, \epsilon, \kappa^{\otimes}$ and $\kappa^{\oplus}$ generate every proof of the logic. Moreover, every such proof

$$X \xrightarrow{\epsilon} \xrightarrow{\kappa^{\otimes}} \xrightarrow{\epsilon} \xrightarrow{\epsilon} \xrightarrow{\eta} \xrightarrow{\eta} \xrightarrow{\kappa^{\otimes}} \xrightarrow{\epsilon} \xrightarrow{\eta} \xrightarrow{\epsilon} \xrightarrow{\kappa^{\otimes}} \xrightarrow{\eta} \xrightarrow{\eta} Z$$

factors uniquely as

$$X \xrightarrow{\kappa^{\otimes}} \xrightarrow{\epsilon} \xrightarrow{\eta} \xrightarrow{\kappa^{\otimes}} Z$$

Corollary: two proofs are equal iff they are equal as strategies
Evaluation of a strategy against a counter-strategy

the formula $X$ is $\varnothing$-free and the formula $Z$ is $\varnothing$-free

hence, the formula $Y$ is both $\varnothing$-free and $\varnothing$-free

hence, the formula $Y$ it is a play!
Categorical combinatorics
(Russ Harmer, Martin Hyland, PAM)

Define a distributivity law

\[ ! \quad ? \quad \xrightarrow{\lambda} \quad ? \quad ! \]

between a monad \( ? \) and a comonad \( ! \) on a category of games.

The category of dialogue games and innocent strategies recovered by a bi-Kleisli construction

\[ ! \quad A \quad \longrightarrow \quad ? \quad B \]

Big step instead of small step
Multi-threaded strategies
(Samuel Mimram, PAM)

Additional hypothesis that negation defines a monoidal functor
Revisiting the negative translation

A rational reconstruction of linear logic
The algebraic point of view (in the style of Boole)

The negated elements of a Heyting algebra form a Boolean algebra.
The algebraic point of view (in the style of Frege)

A double negation monad is **commutative** iff it is **idempotent**. This amounts to the following diagrammatic equations:

In that case, the negated elements form a $\ast$-autonomous category.
Tensor logic

tensor logic = a logic of tensor and negation
= linear logic without $A \cong \neg\neg A$
= the very essence of polarized logic

Offers a synthesis of linear logic, games and continuations

Research program: recast every aspect of linear logic in this setting
A lax monoidal structure

Typically, the family of $n$-ary connectives

$$(A_1 \otimes \cdots \otimes A_n) := \neg (\neg A_1 \otimes \cdots \otimes \neg A_k)$$

$$:= R (LA_1 \otimes \cdots \otimes LA_k)$$

is still associative, but all together, and in the lax sense.
A general phenomenon of adjunctions

Given a 2-monad $T$ and an adjunction

\[
\begin{array}{c}
\mathcal{A} \\
\downarrow \mathbf{L} \\
\mathcal{B} \\
\uparrow \mathbf{R} \\
\mathcal{A}
\end{array}
\]

every lax $T$-algebraic structure

\[ T\mathcal{B} \xrightarrow{b} \mathcal{B} \]

induces a lax $T$-algebraic structure

\[ T\mathcal{A} \xrightarrow{TL} T\mathcal{B} \xrightarrow{b} \mathcal{B} \xrightarrow{R} \mathcal{A} \]
A general phenomenon of adjunctions

Consequently, every adjunction with a monoidal category $\mathcal{B}$ induces a lax action of $\mathcal{B}$ on the category $\mathcal{A}$

$\mathcal{B} \times \mathcal{A} \xrightarrow{\mathcal{B} \times L} \mathcal{B} \times \mathcal{B} \xrightarrow{\otimes_{\mathcal{B}}} \mathcal{B} \xrightarrow{R} \mathcal{A}$

Enscopes the double negation monad and the arrow.
Distributivity laws seen as lax bimodules

The distributivity law

\[ R(A \otimes LB) \otimes C \xrightarrow{\kappa} R(A \otimes L(B \otimes C)) \]

may be seen as a lax notion of bimodule:

\[(A \triangleright B) \triangleleft C \rightarrow A \triangleright (B \triangleleft C)\]

A useful extension of the notion of strong monad (case \(A = *\))
Complementary objects in dialogue categories

An involutive duality in continuations
Linearly distributive categories

A category \( \mathcal{C} \) equipped with

– a symmetric monoidal structure \( \otimes \) and true,

– a symmetric monoidal structure \( \otimes \) and false

with a distributivity law

\[
\delta^L_{A,B,C} : A \otimes (B \otimes C) \rightarrow (A \otimes B) \otimes C
\]

\[
\delta^R_{A,B,C} : (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)
\]
Pentagonal diagram

\[(A \otimes B) \otimes (C \otimes D) \]

\[((A \otimes B) \otimes C) \otimes D\]

\[(A \otimes (B \otimes C)) \otimes D\]

\[A \otimes ((B \otimes C) \otimes D)\]
Pentagonal diagram

\[
\begin{align*}
((A \otimes B) \otimes C) \otimes D & \xrightarrow{\alpha^\otimes \otimes D} (A \otimes (B \otimes (C \otimes D))) \\
& \xrightarrow{\delta^R} (A \otimes B) \otimes (C \otimes D) \\
& \xrightarrow{\alpha^\otimes} A \otimes (B \otimes (C \otimes D)) \\
& \xrightarrow{A \otimes \delta^R} A \otimes ((B \otimes C) \otimes D)
\end{align*}
\]
Pentagonal coherence diagram

\[
\begin{align*}
\delta^L & \quad (A \otimes B) \otimes (C \otimes D) \\
\alpha^\otimes & \quad \delta^L \\
\delta^L \otimes D & \quad A \otimes (B \otimes (C \otimes D)) \\
\delta^L & \quad A \otimes ((B \otimes C) \otimes D)
\end{align*}
\]
Pentagonal coherence diagram

\[
\begin{align*}
&((A \otimes B) \otimes C) \otimes D \\
&\quad \xrightarrow{\alpha \otimes D} \quad (A \otimes (B \otimes C)) \otimes D \\
&\quad \xrightarrow{\delta^L} \quad A \otimes ((B \otimes C) \otimes D) \\
&\quad \xrightarrow{\delta^L} \quad A \otimes (B \otimes (C \otimes D)) \\
&\quad \xrightarrow{\delta^L} \quad (A \otimes B) \otimes (C \otimes D)
\end{align*}
\]
Pentagonal diagram

\[
\begin{align*}
(A \otimes B) \otimes (C \otimes D) & \quad \delta^L \quad \delta^R \\
((A \otimes B) \otimes C) \otimes D & \quad \delta^R \otimes D \\
(A \otimes (B \otimes C)) \otimes D & \quad \alpha^\otimes \\
A \otimes ((B \otimes C) \otimes D) & \quad A \otimes \delta^L
\end{align*}
\]
Pentagonal diagram

\[
\begin{align*}
((A \otimes B) \otimes C) \otimes D & \quad \quad (A \otimes B) \otimes (C \otimes D) \quad \quad A \otimes (B \otimes (C \otimes D)) \\
\delta^L \otimes D & \quad \quad \delta^R & \quad \quad \delta^L
\end{align*}
\]

\[
\begin{align*}
(A \otimes (B \otimes C)) \otimes D & \quad \quad A \otimes ((B \otimes C) \otimes D) \\
\alpha & \quad \quad A \otimes \delta^R
\end{align*}
\]
Triangular diagrams

\[
\text{true } \otimes (A \otimes B) \xrightarrow{\delta^L} (\text{true } \otimes A) \otimes B
\]

\[
A \otimes B \xrightarrow{\lambda^\otimes} A \otimes B
\]

\[
A \otimes (B \otimes \text{true}) \xrightarrow{\delta^R} A \otimes B \otimes \rho^\otimes
\]

\[A \otimes B\]
Triangular diagrams

\[ A \otimes (B \otimes \text{false}) \xrightarrow{\delta^L} (A \otimes B) \otimes \text{false} \]
\[ A \otimes \rho^\otimes \quad \rho^\otimes \]
\[ A \otimes B \]

\[ (\text{false} \otimes A) \otimes B \xrightarrow{\delta^R} \text{false} \otimes (A \otimes B) \]
\[ \lambda^\otimes \otimes B \quad \lambda^\otimes \]
\[ A \otimes B \]
Right complementary

A right complementary of the object $A$ is an object $B$ together with

\begin{align*}
\text{ax} & : \text{true} \quad \rightarrow \quad A^* \odot A \\
\text{cut} & : \quad A \odot A^* \quad \rightarrow \quad \text{false}
\end{align*}
Complementary pair

making the diagrams

\[
\begin{array}{c}
A \odot \text{true} \xrightarrow{A \otimes \text{ax}} A \odot (A^* \odot A) \\
\downarrow \rho^\otimes \\
A \\
\downarrow \lambda^\otimes \\
false \odot A
\end{array}
\quad
\begin{array}{c}
true \odot A^* \xrightarrow{ax \otimes A^*} (A^* \odot A) \odot A^* \\
\downarrow \delta_L \\
(A \odot A^*) \odot A \\
\downarrow \text{cut} \otimes A \\
A
\end{array}
\quad
\begin{array}{c}
\text{true} \odot A^* \xrightarrow{ax \otimes A^*} (A^* \odot A) \odot A^* \\
\downarrow \delta_R \\
A^* \odot (A \odot A^*) \\
\downarrow A^* \odot \text{cut} \\
A^*
\end{array}
\quad
\begin{array}{c}
A^* \xleftarrow{\rho^\otimes} A^* \odot \text{false} \\
\downarrow \lambda^\otimes \\
false \odot A
\end{array}
\end{array}
\]

commute.
Fact

A $\ast$-autonomous category is the same thing as a linearly distributive category with a complementary object $A^\ast$ for every object $A$.

Established by Blute, Cockett and Seely
Linearly distributive adjunction

A specific notion of adjunction

between two monoidal categories \((\mathcal{A}, \otimes, \text{true})\) and \((\mathcal{B}, \otimes, \text{false})\).
Right complementary

A right complementary of the object $A$ is an object $B$ together with

$$ax : \text{true} \longrightarrow R(A^* \otimes LA)$$

$$cut : L(A \otimes RA^*) \longrightarrow \text{false}$$
Theorem

A dialogue category is the same thing as a linearly distributive adjunction with a complementary functor

\[(A \leftrightarrow A^*) : \mathcal{A} \rightarrow \mathcal{B}\]
An amusing illustration

Free monoids in games
Monoid

A monoid is a game $A$ equipped with two strategies

$$1 \xrightarrow{u} A \xleftarrow{m} A \otimes A$$

such that the diagrams

$$\begin{array}{c}
A \otimes A \otimes A \xrightarrow{m \otimes A} A \otimes A \\
A \otimes m \downarrow \quad m \downarrow \\
A \otimes A \xrightarrow{m} A
\end{array} \quad \begin{array}{c}
1 \otimes A \xrightarrow{u \otimes A} A \otimes A \xrightarrow{A \otimes u} A \otimes 1 \\
\lambda \downarrow \quad m \downarrow \quad \rho \downarrow \\
A \otimes A \xrightarrow{m} A \quad A \xrightarrow{A \otimes u} A
\end{array}$$

commute.
Free commutative monoids

Tensor algebra:

$$TA = \bigoplus_{n} A^\otimes n$$

Symmetric algebra:

$$SA = \bigoplus_{n} A^\otimes n / \sim$$

Games:

$$SA = \int^{n} A^\otimes n$$

Computed as an inductive limit in the category $\mathcal{G}$ of games
Computing the free commutative monoid

**Step 1.** Compute the quotient $A^n$

\[
\begin{array}{c}
A^{\otimes n} \xrightarrow{\text{symmetry}} A^{\otimes n} \xrightarrow{\text{symmetry}} \cdots \xrightarrow{\text{symmetry}} A^{\otimes n} \\
\end{array}
\]

\[
\begin{array}{c}
\xrightarrow{\text{quotient}} A^n
\end{array}
\]

**Step 2.** The free commutative monoid $SA$ is the inductive limit

\[
A^0 \longrightarrow A^1 \longrightarrow A^2 \cdots \longrightarrow A^n \longrightarrow A^{n+1} \cdots
\]

The game $SA$ is the game $A$ where Proponent may backtrack for ever.
Revisiting the drinker’s formula

The expressive power of resource modalities
Four primitive components of logic

1. the negation \( \neg \)
2. the linear conjunction \( \otimes \)
3. the repetition modality \( ! \)
4. existential quantification \( \exists \)

A theory of witnesses – what remains of intuitionism...
The classical formula translated as a game

The repetition modality $!$ gives an explicit account of the fact that Proponent is allowed to change her witness.
The classical formula translated as a game

The formula explicates the exact rules of the game (and space of logical interaction)
The formal proof in tensor logic

Axiom
\[
A(x_0) \vdash A(x_0)
\]

Dereliction
\[
!A(x_0) \vdash A(x_0)
\]

Left \vdash
\[
!A(x_0), \neg A(x_0) \vdash
\]

Weakening
\[

\]

Right \vdash
\[

\]

Left \vdash
\[

\]

Right \exists
\[

\]

Left \vdash
\[

\]

Left \exists
\[

\]

Right \exists
\[

\]

Dereliction
\[

\]

Weakening
\[

\]

Right \vdash
\[

\]

Left \exists
\[

\]

Contraction
\[

\]

Right \vdash
\[

\]

Right \vdash
\[

\]

Right \vdash
\[

\]

Right \vdash
\[

\]

Right \vdash
The intuitionistic formula translated as a game

This formula is not valid, because Proponent has to choose the witness \( y \) immediately – and cannot change its mind later.
Conclusion

Logic = Data Structure + Duality

This point of view is accessible thanks to algebra