Denotational semantics of linear logic

Lionel Vaux

I2M, Université d'Aix-Marseille, France

LL2016, Lyon
school: 7 and 8 November 2016
Denotational semantics

- Not about formulas (truth) but about the “meaning” of proofs
- Originated in the study of programming ($\lambda$-calculus)
Denotational semantics

- Not about formulas (truth) but about the “meaning” of proofs
- Originated in the study of programming (\(\lambda\)-calculus)

\[ A \leadsto [A] \]

<table>
<thead>
<tr>
<th>Logic</th>
<th>Computation</th>
<th>Denotational semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Type</td>
<td>Object (mathematical space)</td>
</tr>
</tbody>
</table>
Not about formulas (truth) but about the “meaning” of proofs

Originated in the study of programming ($\lambda$-calculus)

$$\pi \vdash \exists [\pi] \in [A]$$

<table>
<thead>
<tr>
<th>Logic</th>
<th>Computation</th>
<th>Denotational semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Type</td>
<td>Object (mathematical space)</td>
</tr>
<tr>
<td>Proof</td>
<td>Program</td>
<td>Point</td>
</tr>
</tbody>
</table>
Denotational semantics

- Not about formulas (truth) but about the “meaning” of proofs
- Originated in the study of programming (\(\lambda\)-calculus)

\[
\pi \\
\vdash \quad [\pi] : [\Gamma] \to [A]
\]

\[\Gamma \vdash A\]

<table>
<thead>
<tr>
<th>Logic</th>
<th>Computation</th>
<th>Denotational semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Type</td>
<td>Object (mathematical space)</td>
</tr>
<tr>
<td>Proof</td>
<td>Program</td>
<td>Point/Morphism (function)</td>
</tr>
</tbody>
</table>
Denotational semantics

- Not about formulas (truth) but about the “meaning” of proofs
- Originated in the study of programming (λ-calculus)

\[
\pi \\
\vdash \leadsto [\pi] : [\Gamma] \to [A] \times [B]
\]

\[\Gamma \vdash A \land B\]

<table>
<thead>
<tr>
<th>Logic</th>
<th>Computation</th>
<th>Denotational semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Type</td>
<td>Object (mathematical space)</td>
</tr>
<tr>
<td>Proof</td>
<td>Program</td>
<td>Point/Morphism (function)</td>
</tr>
<tr>
<td>Connective</td>
<td>Type constructor</td>
<td>Operation on spaces</td>
</tr>
</tbody>
</table>

L. Vaux (I2M)
Denotational semantics

- Not about formulas (truth) but about the “meaning” of proofs
- Originated in the study of programming ($\lambda$-calculus)

$$
\begin{align*}
\pi & \vdash A \\
\pi' & \vdash B \\
\vdash A \land B
\end{align*}

\implies \langle [\pi], [\pi'] \rangle : [\Gamma] \to [A] \times [B]
$$

<table>
<thead>
<tr>
<th>Logic</th>
<th>Computation</th>
<th>Denotational semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Type</td>
<td>Object (mathematical space)</td>
</tr>
<tr>
<td>Proof</td>
<td>Program</td>
<td>Point/Morphism (function)</td>
</tr>
<tr>
<td>Connective</td>
<td>Type constructor</td>
<td>Operation on spaces</td>
</tr>
<tr>
<td>Deduction rule</td>
<td>Term constructor</td>
<td>Operation on morphisms</td>
</tr>
</tbody>
</table>
Denotational semantics

- Not about formulas (truth) but about the “meaning” of proofs
- Originated in the study of programming (\(\lambda\)-calculus)

\[
\begin{array}{c}
\pi \\
\vdots \\
\Gamma \vdash A
\end{array}
\quad \quad \quad
\begin{array}{c}
\pi' \\
\vdots \\
A \vdash B
\end{array}
\quad \quad \quad
\Gamma \vdash B
\quad \text{cut}
\]

\[
\rightsquigarrow \quad \llbracket \pi' \rrbracket \circ \llbracket \pi \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket B \rrbracket
\]

<table>
<thead>
<tr>
<th>Logic</th>
<th>Computation</th>
<th>Denotational semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Type</td>
<td>Object (mathematical space)</td>
</tr>
<tr>
<td>Proof</td>
<td>Program</td>
<td>Point/Morphism (function)</td>
</tr>
<tr>
<td>Connective</td>
<td>Type constructor</td>
<td>Operation on spaces</td>
</tr>
<tr>
<td>Deduction rule</td>
<td>Term constructor</td>
<td>Operation on morphisms</td>
</tr>
<tr>
<td>Cut</td>
<td>Application</td>
<td>Composition</td>
</tr>
</tbody>
</table>
Denotational semantics

- Not about formulas (truth) but about the “meaning” of proofs
- Originated in the study of programming (\(\lambda\)-calculus)

\[
\begin{align*}
\pi & \quad \pi' \\
\Gamma \vdash A & \quad \Gamma \vdash A \\
\leadsto_{c.e.} & \quad \leadsto
\end{align*}
\]

\[
[\pi] = [\pi'] : [\Gamma] \to [A]
\]

<table>
<thead>
<tr>
<th>Logic</th>
<th>Computation</th>
<th>Denotational semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Type</td>
<td>Object (mathematical space)</td>
</tr>
<tr>
<td>Proof</td>
<td>Type constructor</td>
<td>Point/Morphism (function)</td>
</tr>
<tr>
<td>Connective</td>
<td>Term constructor</td>
<td>Operation on spaces</td>
</tr>
<tr>
<td>Deduction rule</td>
<td>Application</td>
<td>Operation on morphisms</td>
</tr>
<tr>
<td>Cut</td>
<td>Evaluation</td>
<td>Composition</td>
</tr>
<tr>
<td>Cut elimination</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Denotational semantics

- Not about formulas (truth) but about the “meaning” of proofs
- Originated in the study of programming ($\lambda$-calculus)

\[ \Gamma \vdash A \quad \Gamma \vdash A \]

\[ \pi \xrightarrow{c.e.} \pi' \quad \leadsto \quad [\pi] = [\pi'] : [\Gamma] \rightarrow [A] \]

<table>
<thead>
<tr>
<th>Logic</th>
<th>Computation</th>
<th>Denotational semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Type</td>
<td>Object (mathematical space)</td>
</tr>
<tr>
<td>Proof</td>
<td>Program</td>
<td>Point/Morphism (function)</td>
</tr>
<tr>
<td>Connective</td>
<td>Type constructor</td>
<td>Operation on spaces</td>
</tr>
<tr>
<td>Deduction rule</td>
<td>Term constructor</td>
<td>Operation on morphisms</td>
</tr>
<tr>
<td>Cut</td>
<td>Application</td>
<td>Composition</td>
</tr>
<tr>
<td>Cut elimination</td>
<td>Evaluation</td>
<td>=</td>
</tr>
</tbody>
</table>

- A notion of morphism = a category
Denotational semantics

- Not about formulas (truth) but about the “meaning” of proofs
- Originated in the study of programming (λ-calculus)

\[
\begin{align*}
\pi & : \Gamma \vdash A \quad \pi' & : \Gamma \vdash A \\
\Rightarrow & c.e. & \sim & \quad [\pi] = [\pi'] : [\Gamma] \to [A]
\end{align*}
\]

<table>
<thead>
<tr>
<th>Logic</th>
<th>Computation</th>
<th>Denotational semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula</td>
<td>Type</td>
<td>Object (mathematical space)</td>
</tr>
<tr>
<td>Proof</td>
<td>Program</td>
<td>Point/Morphism (function)</td>
</tr>
<tr>
<td>Connective</td>
<td>Type constructor</td>
<td>Operation on spaces</td>
</tr>
<tr>
<td>Deduction rule</td>
<td>Term constructor</td>
<td>Operation on morphisms</td>
</tr>
<tr>
<td>Cut</td>
<td>Application</td>
<td>Composition</td>
</tr>
<tr>
<td>Cut elimination</td>
<td>Evaluation</td>
<td>=</td>
</tr>
</tbody>
</table>

- a notion of morphism = a category
- a denotational model = a category with some additional structure
How linear logic got its name

Models of typed \( \lambda \)-calculus are cartesian closed categories (\( \to, \times \)).

Quantitative semantics, in a nutshell (Girard, early 80’s, before LL)

- alternative to Scott domains, motivated by non-determinism
- Analytic functions are defined by power series: there is a family of coefficients \( (\varphi_{\bar{a}, \beta})_{\bar{a} \in M_f(A), b \in B} \) such that
  \[
  \varphi(\alpha)_b = \sum_{\bar{a} \in M_f(A)} \varphi_{\bar{a}, b} \alpha_{\bar{a}}
  \]
  where \( \alpha^{[a_1, \ldots, a_n]} = \alpha_{a_1} \cdots \alpha_{a_n} \)
How linear logic got its name

Models of typed $\lambda$-calculus are cartesian closed categories ($\to$, $\times$).

Quantitative semantics, in a nutshell (Girard, early 80’s, before LL)

- alternative to Scott domains, motivated by non-determinism
- Analytic functions are defined by power series: there is a family of coefficients $(\varphi_{\alpha,\beta})_{\alpha \in \mathcal{M}_f(A), \beta \in B}$ such that
  \[
  \varphi(\alpha)_{\beta} = \sum_{\alpha \in \mathcal{M}_f(A)} \varphi_{\alpha,\beta} \alpha^\lambda 
  \]
  where $\alpha^{[a_1,\ldots,a_n]} = \alpha_{a_1} \cdots \alpha_{a_n}$

Linear decomposition of the function space

Setting $\alpha^!_\lambda = \alpha^\lambda$ we obtain $\alpha^! \in \mathcal{M}_f(A)$ and $\varphi(\alpha) = \varphi \cdot \alpha^!$ where $\cdot$ denotes matrix application:
\[
\text{An} \left( R^A, R^B \right) \cong \text{Lin} \left( \mathcal{M}_f(A), R^B \right)
\]
How linear logic got its name

Models of typed $\lambda$-calculus are cartesian closed categories ($\to$, $\times$).

Quantitative semantics, in a nutshell (Girard, early 80’s, before LL)

- alternative to Scott domains, motivated by non-determinism
- Analytic functions are defined by power series: there is a family of coefficients $(\varphi_{a,\beta})_{a \in M_f(A), b \in B}$ such that
  $$\varphi(\alpha)_b = \sum_{\bar{a} \in M_f(A)} \varphi_{\bar{a},b} \alpha^{\bar{a}}$$
  where
  $$\alpha^{[a_1, \ldots, a_n]} = \alpha_{a_1} \cdots \alpha_{a_n}$$

Linear decomposition of the function space

Setting $\alpha^!_a = \alpha^{\bar{a}}$ we obtain $\alpha^! \in R^{m_f(A)}$ and $\varphi(\alpha) = \varphi \cdot \alpha^!$ where $\cdot$ denotes matrix application:

$$\text{An} \left( R^A, R^B \right) \iff \text{Lin} \left( R^{m_f(A)}, R^B \right)$$

$$A \Rightarrow B \iff \;!A \rightarrow B$$
There is a catch with quantitative semantics: infinite sums might not converge.
There is a catch with quantitative semantics: infinite sums might not converge.

If we stick to the linear fragment, we can consider finite dimensional spaces.
There is a catch with quantitative semantics: infinite sums might not converge.
If we stick to the linear fragment, we can consider finite dimensional spaces.

A category of finite dimensional vector spaces: $\text{Mat}$

- objects = finite sets
- morphisms $\text{Mat}(A, B) = \text{matrices } R^{A \times B}$

- The finite set $A$ is the canonical base of $R^A$
- Matrices $R^{A \times B}$ represent linear maps from $R^A$ to $R^B$
Proofs as matrices

We interpret formulas $A$ as finite sets $\llbracket A \rrbracket$ and proofs $\pi : A \vdash B$ as matrices $\text{Mat}(\llbracket A \rrbracket, \llbracket B \rrbracket)$: for all $a \in \llbracket A \rrbracket$ and $b \in \llbracket B \rrbracket$, we have $\llbracket \pi \rrbracket_{a,b} \in R$. 
Proofs as matrices

We interpret formulas $A$ as finite sets $\llbracket A \rrbracket$ and proofs $\pi : A \vdash B$ as matrices $\text{Mat}(\llbracket A \rrbracket, \llbracket B \rrbracket)$: for all $a \in \llbracket A \rrbracket$ and $b \in \llbracket B \rrbracket$, we have $\llbracket \pi \rrbracket_{a,b} \in R$.

- The axiom is the identity matrix

$$\llbracket \begin{array}{c} \vdash A^\perp, A \\
\text{ax} \end{array} \rrbracket_{a,a'} = \delta_{a,a'}$$
Proofs as matrices

We interpret formulas $A$ as finite sets $[A]$ and proofs $\pi : A \vdash B$ as matrices $\text{Mat}([A], [B])$: for all $a \in [A]$ and $b \in [B]$, we have $[\pi]_{a,b} \in R$.

- The axiom is the identity matrix

$$\begin{bmatrix}
\vdash A \perp, A \\
\text{ax}
\end{bmatrix}_{a,a'} = \delta_{a,a'}$$

- Cut is composition, i.e. matrix product:

$$\begin{bmatrix}
\vdash A \perp, B & \vdash B \perp, C \\
\pi & \pi' \\
\vdash A \perp, C & \text{cut}
\end{bmatrix}_{a,c} = \sum_{b \in [B]} [\pi]_{a,b} [\pi']_{b,c}$$
The symmetric monoidal closed structure: $\otimes$, $\multimap$

Tensor product

Given matrices $\varphi \in \mathbb{R}^{A \times B}$ and $\varphi' \in \mathbb{R}^{A' \times B'}$, we can form their tensor product $\varphi \otimes \varphi' \in \mathbb{R}^{(A \times A') \times (B \times B')}$ setting $(\varphi \otimes \varphi')(a,a'),(b,b') = \varphi_a,b \varphi'_a,b'$

- Set $[A \otimes B] = [A] \times [B]$
- This is associative and commutative (up to isomorphism)
- There is a unit $1 = \{\ast\}$

Space of linear maps

$A \times B$ is also the “object of morphisms” from $A$ to $B$:

- Set $[A \multimap B] = [A] \times [B]$
- We have an adjunction: $\text{Mat}(A \otimes B, C) \cong \text{Mat}(A, B \multimap C)$
The $\ast$-autonomous structure: $A^\bot = A \multimap \bot$

**Dual space**

- The dual of $R^A$ is the set of linear forms on $R^A$: $\text{Lin}(R^A, R)$
The *-autonomous structure: \( A \perp = A \multimap \bot \)

### Dual space

- The dual of \( R^A \) is the set of linear forms on \( R^A \): \( \text{Lin}(R^A, R) \)
- Fix \( \llbracket \bot \rrbracket \) to be a singleton: \( \llbracket \bot \rrbracket = \{ * \} \) so that \( \text{Lin}(R^A, R) \cong \text{Mat}(A, \bot) \)
- By monoidal closedness, we can derive a canonical morphism
  \( \varphi \in \text{Mat}(A, (A \multimap \bot) \multimap \bot) \) from the identity on \( A \multimap \bot \):
  \[
  \varphi_{a,((a',*),*)} = \delta_{a,a'}
  \]
- \( \varphi \) is an isomorphism: this is *-autonomy
The $\ast$-autonomous structure: $A^\perp = A \multimap \bot$

Dual space

- The dual of $R^A$ is the set of linear forms on $R^A$: $\text{Lin}(R^A, R)$
- Fix $\lbrack \bot \rbrack$ to be a singleton: $\lbrack \bot \rbrack = \{\ast\}$ so that $\text{Lin}(R^A, R) \cong \text{Mat}(A, \bot)$
- By monoidal closedness, we can derive a canonical morphism $\varphi \in \text{Mat}(A, (A \multimap \bot) \multimap \bot)$ from the identity on $A \multimap \bot$:
  $\varphi_{a,((a',\ast),\ast)} = \delta_{a,a'}$
- $\varphi$ is an isomorphism: this is $\ast$-autonomy

- This isomorphism reflects the identity $A^{\perp\perp} = A$
- One recovers $A \multimap B$ as $(A^\perp \otimes B^\perp)^\perp$
- In fact here $R^{A \times \{\ast\}} \cong R^A$: we can set, more simply, $\lbrack A^\perp \rbrack = \lbrack A \rbrack$
- In particular $A \otimes B = A \multimap B$: the model is compact closed
## Interpretation of MLL

### General guidelines

- Interpret a proof of $A_1, \ldots, A_n \vdash B$ as a morphism:

$$A_1 \otimes \cdots \otimes A_n \rightarrow B$$

- By the $\ast$-autonomous structure, this is equivalent to a morphism

$$\mathbf{1} \rightarrow A_1^\perp \rightarrow \cdots \rightarrow A_n^\perp \rightarrow B$$

- This reflects the equiprovability of

$$\Gamma \vdash \Delta, \quad \vdash \Gamma^\perp, \Delta \quad \text{and} \quad \Gamma, \Delta^\perp \vdash$$

### In Mat

If $\pi$ is a proof of $\vdash A_1, \ldots, A_n$ then

$$[\pi] \in R[A_1] \times \cdots \times [A_n]$$
Interpretation of MLL

Rules

\[ \frac{\vdash A \perp, A}{\vdash A \perp, A} \text{ ax} \]
\[ \frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta} \quad \otimes \quad s,(a,b),t \]
\[ \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \multimap B} \quad \Rightarrow \quad s,(a,b) \]
\[ \frac{\vdash \Gamma, A \quad \vdash A \perp, \Delta}{\vdash \Gamma, \Delta} \quad \text{cut} \quad s,t \]
\[ = \delta_{a,a'} \]
\[ = \llbracket \pi \rrbracket_s a \llbracket \pi' \rrbracket_t b,t \]
\[ = \llbracket \pi \rrbracket_s a b \]
\[ = \sum_{a \in [A]} \llbracket \pi \rrbracket_s a \llbracket \pi' \rrbracket_a t \]
Vector spaces admit cartesian products: $R^A \times R^B \cong R^{A+B}$ where $A + B$ denotes the disjoint union ($\{1\} \times A) \cup (\{2\} \times B$

$[A \& B] = [A] + [B]$

This is the product type of programming languages: given two morphisms $\varphi \in \text{Mat}(C, A)$ and $\varphi \in \text{Mat}(C, B)$, one forms $\langle \varphi, \psi \rangle \in \text{Mat}(C, A + B)$, hence the rule

$$\frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} \text{ &}$$
Vector spaces admit cartesian products: \( R^A \times R^B \cong R^{A+B} \) where \( A + B \) denotes the disjoint union \((\{1\} \times A) \cup (\{2\} \times B)\)

\[
[A \& B] = [A] + [B]
\]

This is the product type of programming languages: given two morphisms \( \varphi \in \text{Mat}(C, A) \) and \( \varphi \in \text{Mat}(C, B) \), one forms \( \langle \varphi, \psi \rangle \in \text{Mat}(C, A + B) \), hence the rule

\[
\vdash \Gamma, A \quad \vdash \Gamma, B \\
\quad \quad \quad \vdash \Gamma, A \& B \\
\quad \quad \quad \&
\]

This is the additive conjunction
Cartesian product

- Vector spaces admit cartesian products: $R^A \times R^B \cong R^{A+B}$ where $A + B$ denotes the disjoint union $(\{1\} \times A) \cup (\{2\} \times B)$
- $[A \& B] = [A] + [B]$
- This is the product type of programming languages: given two morphisms $\varphi \in \text{Mat}(C, A)$ and $\varphi \in \text{Mat}(C, B)$, one forms $\langle \varphi, \psi \rangle \in \text{Mat}(C, A + B)$, hence the rule

$$\Gamma, A \vdash \Gamma, B \quad \Gamma, A \vdash \Gamma, B$$

This is the additive conjunction

(Note that $A \otimes B$ is not the type of pairs)
Additives

Direct sums

- By duality we obtain $A \oplus B = (A^\perp & B^\perp)^\perp$
- Here $\llbracket A \oplus B \rrbracket = \llbracket A & B \rrbracket$ (because of compact closedness)
- This is the sum type of programming languages: it comes with injections $\iota_l \in \text{Mat}(A, A + B)$ and $\iota_r \in \text{Mat}(B, A + B)$ hence the rules

\[
\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \quad \oplus_l \quad \text{and} \quad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B} \quad \oplus_r
\]

The additive disjunction
The **additive** disjunction

In fact we gave only half of the picture and we miss:

- projections for the products
- case definitions for sums

but they follow from pairs and injections by duality (see the cut elimination later)
Additives
Semantics

\[
\begin{array}{c}
\pi_1 & \pi_2 \\
\vdots & \vdots \\
\Gamma, A & \Gamma, B \\
\Gamma, A \& B \\
\end{array}
\] \\
\Rightarrow
\begin{array}{c}
\pi_1 [\pi_1]_{s,a} & \text{if } (i, c) = (1, a) \\
\pi_2 [\pi_2]_{s,b} & \text{if } (i, c) = (2, b) \\
\end{array}
\]

\[
\begin{array}{c}
\pi \\
\vdots \\
\Gamma, A \\
\Gamma, A \oplus B \\
\end{array}
\] \\
\Rightarrow
\begin{array}{c}
\pi [\pi]_{s,a} & \text{if } (i, c) = (1, a) \\
0 & \text{if } i = 2 \\
\end{array}
\]

\[
\begin{array}{c}
\pi \\
\vdots \\
\Gamma, B \\
\Gamma, A \oplus B \\
\end{array}
\] \\
\Rightarrow
\begin{array}{c}
0 & \text{if } 1 = 2 \\
\pi [\pi]_{s,b} & \text{if } (i, c) = (2, b) \\
\end{array}
\]
Additives
Cut elimination

\[
\begin{array}{c}
\pi_1 & \pi_2 \\
\vdots & \vdots \\
\vdash \Gamma, A & \vdash \Gamma, B \\
\hline 
&\& \\
&\& \vdash \Gamma, A \& B \\
&\& \vdash \Gamma, \Delta \\
\pi_1 & \pi_2 & \pi' \\
\vdots & \vdots & \vdots \\
\vdash \Gamma, A & \vdash \Gamma, B & \vdash \Delta, A_\perp & \vdash \Delta, B_\perp \\
& \hline & \& \& \\
& \& \& \vdash \Gamma, A_\perp \oplus B_\perp & \vdash \Gamma, \Delta \\
\end{array}
\]

\[\oplus_l \text{ cut} \]

\[
\begin{array}{c}
\pi_1 & \pi' \\
\vdots & \vdots \\
\vdash \Gamma, A & \vdash \Delta, A_\perp \\
\hline 
& \& \\
& \& \vdash \Gamma, \Delta \\
\end{array}
\]

\[\text{cut} \]

\[
\begin{array}{c}
\pi_2 & \pi' \\
\vdots & \vdots \\
\vdash \Gamma, B & \vdash \Delta, B_\perp \\
\hline 
& \& \\
& \& \vdash \Gamma, A_\perp \oplus B_\perp & \vdash \Gamma, \Delta \\
\end{array}
\]

\[\oplus_r \text{ cut} \]

\[
\begin{array}{c}
\pi_2 & \pi' \\
\vdots & \vdots \\
\vdash \Gamma, B & \vdash \Delta, B_\perp \\
\hline 
& \& \\
& \& \vdash \Gamma, \Delta \\
\end{array}
\]

\[\text{cut} \]

A good notion of proof nets with additives is difficult to design but additives live very naturally in the semantics (like units).
Additives
Cut elimination

\[ \frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \land B} \quad \frac{\Delta, A^\perp}{\Delta, A^\perp \oplus B^\perp} \quad \frac{}{\Gamma, \Delta} \]

\[ \frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \land B} \quad \frac{\Delta, B^\perp}{\Delta, A^\perp \oplus B^\perp} \quad \frac{}{\Gamma, \Delta} \]

- non-local!
- a good notion of proof nets with additives is difficult to design
- but additives live very naturally in the semantics (like units)
The relational model

Setting $R = \mathbb{B}$ (booleans), we recover the model presented yesterday:

- write $s \in [[\pi]]$ for $[[\pi]]_s = 1$
- a matrix $\varphi \in \mathbb{B}^{A \times B}$ is a relation $\varphi \subseteq A \times B$
- we no longer care about finiteness, since the composition of relations $\varphi \subseteq A \times B$ and $\psi \subseteq B \times C$ is always defined:
  
  $$ (a, c) \in \psi \varphi \iff \text{there exists } b \in B \text{ such that } (a, b) \in \varphi \text{ and } (b, c) \in \psi $$

- write $\text{Rel}$ for the category of sets and relations

Recall from yesterday that the relational model can be described by a kind of type system:

$$ \begin{array}{l}
\vdash A_1, \ldots, A_n \\
\pi \vdash a_1, \ldots, a_n \\
\vdash A_1, \ldots, A_n
\end{array} \quad \text{is derivable} \quad \iff \quad (a_1, \ldots, a_n) \in \left[ \begin{array}{l}
\vdash A_1, \ldots, A_n \\
\pi \\
\vdash A_1, \ldots, A_n
\end{array} \right] $$
The relational model
Rules for MLL

\[
[A \otimes B] = [A \otimes B] = [A] \times [B]
\]

\[
\begin{array}{c}
\frac{\vdash \Gamma, A^\bot}{\vdash \Gamma, A} & \frac{\vdash \Gamma, \Delta}{\vdash \Gamma, \Delta} \\
\text{ax} & \text{cut}
\end{array}
\]

\[
\frac{\vdash \Gamma, A \otimes B, \Delta}{\vdash \Gamma, A \otimes B} & \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \otimes B}
\]

\[
\frac{\vdash \Gamma, A^\bot}{\vdash \Gamma, A^\bot} & \frac{\vdash \Gamma, A^\bot}{\vdash \Gamma, A^\bot}
\]

\[
\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A, B} & \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A, B}
\]

\[
\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A, B}
\]
The relational model

Rules for MLL MALL

\[ [A \otimes B] = [A \And B] = [A] \times [B] \]

\[ \frac{\vdash \Gamma, A \Downarrow \quad \vdash \Gamma, A, \Delta \quad \text{cut}}{\vdash \Gamma, A \Downarrow, A} \]

\[ \frac{s \quad a \quad t}{\vdash \Gamma, A \Downarrow, A} \]

\[ \vdash \Gamma, A \Downarrow, A \]

\[ \frac{s \quad a \quad b \\ t}{\vdash \Gamma, A, B} \quad \vdash \Gamma, A \And B \]

\[ \vdash \Gamma, A \And B \]

\[ [A \& B] = [A \oplus B] = [A] + [B] \]

\[ \frac{s \quad a \quad b}{\vdash \Gamma, A \& B} \quad \vdash \Gamma, A \& B \]

\[ \vdash \Gamma, A \& B \]

\[ \frac{s \quad a}{\vdash \Gamma, 1, A} \quad \vdash \Gamma, 1, A \]

\[ \vdash \Gamma, A \& B \]

\[ \frac{s \quad (1,a)}{\vdash \Gamma, A \& B} \quad \vdash \Gamma, A \& B \]

\[ \vdash \Gamma, A \& B \]

\[ \frac{s \quad (2,b)}{\vdash \Gamma, 2, B} \quad \vdash \Gamma, 2, B \]

\[ \vdash \Gamma, A \& B \]

\[ \frac{s \quad (2,b)}{\vdash \Gamma, A \& B} \quad \vdash \Gamma, A \& B \]

\[ \vdash \Gamma, A \& B \]

\[ \frac{s \quad (1,a)}{\vdash \Gamma, A \oplus B} \quad \vdash \Gamma, A \oplus B \]

\[ \vdash \Gamma, A \oplus B \]

\[ \frac{s \quad (1,a)}{\vdash \Gamma, A \oplus B} \quad \vdash \Gamma, A \oplus B \]

\[ \vdash \Gamma, A \oplus B \]

\[ \frac{s \quad (2,b)}{\vdash \Gamma, A \oplus B} \quad \vdash \Gamma, A \oplus B \]

\[ \vdash \Gamma, A \oplus B \]
Exponentials in $\text{Rel}$

Ingredients

Recall from quantitative semantics: $[!A] = \mathcal{M}_f ([A])$.
Exponentials in $\text{Rel}$

Ingredients

Recall from quantitative semantics: $\llbracket !A \rrbracket = \mathcal{M}_f (\llbracket A \rrbracket)$.

- This defines an functor in $\text{Rel}$: if $\varphi \in \text{Rel}(A, B)$ then
  $$!\varphi = \{(a_1, \ldots, a_n, b_1, \ldots, b_n) ; (a_i, b_i) \in \varphi \text{ for all } i \} \in \text{Rel}(!A, !B)$$
Recall from quantitative semantics: $\llbracket !A \rrbracket = \mathcal{M}_f (\llbracket A \rrbracket)$.

- This defines an functor in $\text{Rel}$: if $\varphi \in \text{Rel}(A, B)$ then
  \[
  !\varphi = \{ ([a_1, \ldots, a_n], [b_1, \ldots, b_n]) ; (a_i, b_i) \in \varphi \text{ for all } i \} \in \text{Rel}(!A, !B)
  \]
- This has the structure of a comonad with
  \[
  \text{der} = \{ ([a], a) ; a \in A \} \in \text{Rel}(!A, A)
  \]
  and
  \[
  \text{digg} = \{ (\sum \bar{a}_1 + \ldots + \bar{a}_n, [\bar{a}_1, \ldots, \bar{a}_n]) ; \bar{a}_1, \ldots, \bar{a}_n \in !A \} \in \text{Rel}(!A, !!A)
  \]
Exponentials in \(\text{Rel}\)

**Ingredients**

Recall from quantitative semantics: \(\mathbb{M}_f([A]) = \mathbb{M}_f([A])\).

- This defines an functor in \(\text{Rel}\): if \(\varphi \in \text{Rel}(A, B)\) then
  \[
  !\varphi = \{([a_1, \ldots, a_n], [b_1, \ldots, b_n]); (a_i, b_i) \in \varphi \text{ for all } i\} \in \text{Rel}(!A, !B)
  \]

- This has the structure of a comonad with

  \[
  \text{der} = \{([a], a); a \in A\} \in \text{Rel}(!A, A)
  \]

  and

  \[
  \text{digg} = \{(\bar{a}_1 + \ldots + \bar{a}_n, [\bar{a}_1, \ldots, \bar{a}_n]); \bar{a}_1, \ldots, \bar{a}_n \in !A\} \in \text{Rel}(!A, !!A)
  \]

- There is an isomorphism \((A & B) \cong !A \otimes !B\):

  \[
  \{(([1, a_1], \ldots, (1, a_n), (2, b_1), \ldots, (2, b_n]), ([a_1, \ldots, a_n], [b_1, \ldots, b_p]))\}
  \]
Exponentials in Rel
Ingredients

Recall from quantitative semantics: \([!A] = \mathcal{M}_f ([A])\).

▶ This defines an functor in Rel: if \(\varphi \in \text{Rel}(A, B)\) then

\[ !\varphi = \{([a_1, \ldots, a_n], [b_1, \ldots, b_n]); \ (a_i, b_i) \in \varphi \text{ for all } i \} \in \text{Rel}(!A, !B) \]

▶ This has the structure of a comonad with

\[ \text{der} = \{([a]; a \in A) \} \in \text{Rel}(!A, A) \]

and

\[ \text{digg} = \{(\bar{a}_1 + \ldots + \bar{a}_n, [\bar{a}_1, \ldots, \bar{a}_n]); \ \bar{a}_1, \ldots, \bar{a}_n \in !A \} \in \text{Rel}(!A, !!A) \]

▶ There is an isomorphism \(!{(A \& B)} \cong !A \otimes !B:\)

\[ \{([(1, a_1), \ldots, (1, a_n), (2, b_1), \ldots, (2, b_n)],[a_1, \ldots, a_n], [b_1, \ldots, b_p]))\} \]

This (plus a bunch of other conditions to make everything work together) makes \(\text{Rel}\) a model of LL.
Exponentials in Rel Interpretation

Structural rules: resource management

\[
\begin{align*}
\frac{}{\vdash \Gamma, A} & \quad ?d \\
\frac{s \quad a}{s\quad [a]} & \quad \vdash \Gamma, ?A \\
\frac{s}{s} & \quad \vdash \Gamma, ?A \\
\frac{s \quad \bar{a} \quad \bar{a}'}{s \quad \bar{a}+\bar{a}'} & \quad \vdash \Gamma, ?A
\end{align*}
\]
Structural rules: resource management

\[
\frac{s \quad a}{\Gamma, A} \quad d
\]
\[
\frac{s}{\Gamma, A} \quad w
\]
\[
\frac{s \quad \bar{a} \quad \bar{a}'}{\Gamma, A, A} \quad c
\]

Promotion: resource factory

\[
\frac{\bar{s}_i \quad a_i}{\omega \Gamma, A} \quad (\text{for } 1 \leq i \leq n)
\]
\[
\frac{\sum_{i=1}^{n} \bar{s}_i \quad [a_1, \ldots, a_n]}{\omega \Gamma, !A}
\]

(Quantitative semantics relies on quite complicated formulas for the coefficients.)
Example

\[
\begin{align*}
\quad & ax \\
\quad & \vdash A^\perp, A \\
\quad & \vdash A^\perp \oplus B^\perp, A \\
\quad & ?d \\
\quad & \vdash ?(A^\perp \oplus B^\perp), A \\
\quad & \vdash ?(A^\perp \oplus B^\perp), ?d \\
\quad & \vdash ?(A^\perp \oplus B^\perp), A^\perp, A!
\end{align*}
\]

\[
\begin{align*}
\quad & ax \\
\quad & \vdash B^\perp, B \\
\quad & \vdash A^\perp \oplus B^\perp, B \\
\quad & ?d \\
\quad & \vdash ?(A^\perp \oplus B^\perp), B \\
\quad & \vdash ?(A^\perp \oplus B^\perp), ?d \\
\quad & \vdash ?(A^\perp \oplus B^\perp), B^\perp, !B
\end{align*}
\]

\[
\begin{align*}
\quad & ?(A^\perp \oplus B^\perp), ?(A^\perp \oplus B^\perp), !A \otimes !B \\
\quad & \vdash ?(A^\perp \oplus B^\perp), ?c \\
\quad & \vdash ?(A^\perp \oplus B^\perp), !A \otimes !B
\end{align*}
\]
Example

\[
\frac{\text{ax}}{\vdash A \perp, A} \quad \oplus_l
\]
\[
\frac{\vdash A \perp \oplus B \perp, A}{\vdash ?(A \perp \oplus B \perp), A} \quad ?d
\]
\[
\frac{\vdash ?(A \perp \oplus B \perp), !A}{\vdash ?(A \perp \oplus B \perp), !A \otimes !B} \quad \otimes
\]
\[
\frac{\vdash ?(A \perp \oplus B \perp), ?(A \perp \oplus B \perp), !A \otimes !B}{\vdash ?(A \perp \oplus B \perp), !A \otimes !B} \quad ?c
\]
\[
\frac{\text{ax}}{\vdash B \perp, B} \quad \oplus_r
\]
\[
\frac{\vdash A \perp \oplus B \perp, B}{\vdash ?(A \perp \oplus B \perp), B} \quad ?d
\]
\[
\frac{\vdash ?(A \perp \oplus B \perp), !B}{\vdash ?(A \perp \oplus B \perp), !A \otimes !B} \quad \otimes
\]
\[
\frac{\vdash ?(A \perp \oplus B \perp), ?(A \perp \oplus B \perp), !A \otimes !B}{\vdash ?(A \perp \oplus B \perp), !A \otimes !B} \quad ?c
\]
Example

\[
\begin{align*}
\frac{a}{\vdash A \perp, A} & \quad \text{ax} \\
\frac{}{\vdash A \perp \oplus B \perp, A} & \quad \oplus I \\
\frac{}{\vdash ?(A \perp \oplus B \perp), A} & \quad ?d \\
\frac{}{\vdash ?(A \perp \oplus B \perp), \quad !A} & \quad ! \\
\frac{}{\vdash ?(A \perp \oplus B \perp), \quad ?(A \perp \oplus B \perp), \quad !A \otimes !B} & \quad ?c \\
\frac{}{\vdash ?(A \perp \oplus B \perp), \quad !A \otimes !B} & \quad \otimes
\end{align*}
\]

\[
\begin{align*}
\frac{b}{\vdash B \perp, B} & \quad \text{ax} \\
\frac{}{\vdash A \perp \oplus B \perp, B} & \quad \oplus r \\
\frac{}{\vdash ?(A \perp \oplus B \perp), B} & \quad ?d \\
\frac{}{\vdash ?(A \perp \oplus B \perp), \quad !B} & \quad ! \\
\frac{}{\vdash ?(A \perp \oplus B \perp), \quad ?(A \perp \oplus B \perp), \quad !A \otimes !B} & \quad ?c \\
\frac{}{\vdash ?(A \perp \oplus B \perp), \quad !A \otimes !B} & \quad \otimes
\end{align*}
\]
Example

\[
\begin{align*}
\text{ax} & \quad \frac{a}{\vdash A, A} \\
& \quad \frac{a}{\vdash A \otimes B, A} \quad \oplus_I \\
& \quad \frac{[(1, a)]}{\vdash ?(A \otimes B), A} \quad ?d \\
& \quad \vdash ?(A \otimes B), A \\
& \quad \vdash ?(A \otimes B), !A \\
& \quad \vdash ?(A \otimes B), ?(A \otimes B), !A \otimes !B \\
& \quad \vdash ?(A \otimes B), !A \otimes !B
\end{align*}
\]

\[
\begin{align*}
\text{ax} & \quad \frac{b}{\vdash B, B} \\
& \quad \frac{(2, b)}{\vdash A \otimes B, B} \quad \oplus_r \\
& \quad \frac{[(2, b)]}{\vdash ?(A \otimes B), B} \quad ?d \\
& \quad \vdash ?(A \otimes B), B \\
& \quad \vdash ?(A \otimes B), !B \\
& \quad \vdash ?(A \otimes B), ?(A \otimes B), !A \otimes !B \\
& \quad \vdash ?(A \otimes B), !A \otimes !B
\end{align*}
\]
Example

\[
\begin{array}{ll}
\text{ax} & \frac{a_i}{A \perp, A} \\
\frac{a_i}{A \perp \oplus B \perp, A} & \frac{a_i}{[(1,a_i)]} \\
\frac{[(1,a_i),\ldots,(1,a_n)] [a_1,\ldots,a_n]}{\vdash ?(A \perp \oplus B \perp), A} & \frac{[(2,b_i)] b_i}{\vdash ?(A \perp \oplus B \perp), B} \\
\end{array}
\]

\[
\begin{array}{ll}
\text{ax} & \frac{b_i}{B \perp, B} \\
\frac{b_i}{A \perp \oplus B \perp, B} & \frac{b_i}{[(2,b_i)]} \\
\frac{[(2,b_2),\ldots,(2,b_p)] [b_2,\ldots,b_p]}{\vdash ?(A \perp \oplus B \perp), B} & \frac{[(1,a_i),\ldots,(1,a_n)] a_i}{\vdash ?(A \perp \oplus B \perp), A} \\
\end{array}
\]

\[
\begin{array}{ll}
\vdash ?(A \perp \oplus B \perp), ?(A \perp \oplus B \perp), !A \otimes !B & \vdash ?(A \perp \oplus B \perp), ?(A \perp \oplus B \perp), !A \otimes !B \\
\end{array}
\]

\[
\begin{array}{ll}
\vdash ?(A \perp \oplus B \perp), ?(A \perp \oplus B \perp), !A \otimes !B & \vdash ?(A \perp \oplus B \perp), !A \otimes !B \\
\end{array}
\]
Example

\[
\begin{align*}
\text{ax} & \quad \vdash A \bot, A \\
\text{ax} & \quad \vdash B \bot, B \\
\oplus_l & \quad \vdash A \bot \oplus B \bot, A \\
\oplus_r & \quad \vdash B \bot \oplus A \bot, B \\
\text{?d} & \quad \vdash \equiv{(1,a_i)} A \\
\text{?d} & \quad \vdash \equiv{(2,b_i)} B \\
\text{?c} & \quad \vdash \equiv{(1,a_i)} \equiv{(2,b_i)} A \otimes !B \\
\end{align*}
\]
Example

\[
\begin{align*}
\frac{a_i}{\vdash A^\bot, A} & \quad \text{ax} \\
\frac{(1,a_i)}{} & \quad \vdash A^\bot \oplus B^\bot, A \\
\frac{\vdash A^\bot \oplus B^\bot, A}{[(1,a_i)]} & \quad \vdash ?(A^\bot \oplus B^\bot), A \\
\frac{[a_1,\ldots,a_n]}{[1,a_1),\ldots,(1,a_n)]} & \quad \vdash ?(A^\bot \oplus B^\bot), !A
\end{align*}
\]

\[
\begin{align*}
\frac{b_i}{\vdash B^\bot, B} & \quad \text{ax} \\
\frac{(2,b_i)}{} & \quad \vdash A^\bot \oplus B^\bot, B \\
\frac{\vdash A^\bot \oplus B^\bot, B}{[(2,b_i)]} & \quad \vdash ?(A^\bot \oplus B^\bot), !B \\
\frac{[b_2,\ldots,b_p]}{[2,b_2),\ldots,(2,b_p)]} & \quad \vdash ?(A^\bot \oplus B^\bot), !B
\end{align*}
\]

\[
\begin{align*}
\frac{[a_1,\ldots,a_n],[b_2,\ldots,b_p]}{[(1,a_1),\ldots,(1,a_n)],[(2,b_2),\ldots,(2,b_p)]} & \quad \vdash ?(A^\bot \oplus B^\bot), ?(A^\bot \oplus B^\bot), !A \otimes !B \\
\frac{[(1,a_1),\ldots,(1,a_n),\ldots,(1,a_n)],[(2,b_2),\ldots,(2,b_p)]}{[a_1,\ldots,a_n],[b_2,\ldots,b_p]} & \quad \vdash ?(A^\bot \oplus B^\bot), !A \otimes !B
\end{align*}
\]
The relational model is highly non uniform: the union of arbitrary relations is still a relation. This is good for capturing non-deterministic superpositions of proofs. But it comes with many degeneracies: most notably $A^\perp = A$; and it does not see vicious cycles in proof structures.
The relational model is highly non uniform: the union of arbitrary relations is still a relation.

This is good for capturing non-deterministic superpositions of proofs.

But it comes with many degeneracies: most notably $A^\perp = A$; and it does not see vicious cycles in proof structures.

**Coherence spaces**

A coherence space $A$ is the data of a set $|A|$ (the web of $A$) and of a reflexive and symmetric relation $\bowtie_A$ on $|A|$.

The intuitive meaning of $a \bowtie_A a'$ is that $a$ and $a'$ can appear simultaneously in the semantics of the same proof of $A$: we will interpret proofs by cliques.

$$\mathcal{C}(A) = \{ \alpha \subseteq |A| ; \ a \bowtie_A a', \ for \ all \ a, a' \in \alpha \}$$
Negation reverses the coherence (but we still need a symmetric relation). Define three new relations on $|A|$: 

- $a \perp_A$ iff $a \not\in_A a'$; 
- $a \bowtie_A$ iff $a = a'$ or $a \perp_A a'$; 
- $a \asymp_A$ iff $a \bowtie_A a'$ and $a \neq a'$.

**Dual** 

Define $A^\perp$ by $|A^\perp| = |A|$ and $\bowtie_{A^\perp} = \bowtie_A$. 
Multiplicatives

It is natural to set:

\[ |A \otimes B| = |A| \times |B| \]

\[ (a, b) \lhd _{A \otimes B} (a', b') \iff a \lhd _A a' \text{ and } b \lhd _B b' \]

We deduce \( \lhd _\otimes \) by de Morgan rules:

\[ (a, b) \lhd _{A \lhd \otimes B} (a', b') \iff a \lhd _A a' \text{ or } b \lhd _B b' \]

Linear implication \( A \twoheadrightarrow B = (A \otimes B^\perp)^\perp \) follows:

\[ (a, b) \lhd _{A \twoheadrightarrow B} (a', b') \iff a \lhd _A a' \text{ entails } b \lhd _B b' \]

It should be clear that if \( \varphi \in \mathcal{C}(A \twoheadrightarrow B) \) and \( \psi \in \mathcal{C}(B \twoheadrightarrow C) \), then \( \psi \varphi \in \mathcal{C}(A \twoheadrightarrow C) \) (where composition is that of relations).
A $\ast$-autonomous category of coherence spaces

Coherence spaces and coherent relations

Write $\text{Coh}$ for the category of coherence spaces, with morphisms $\text{Coh}(A, B) = \mathcal{C}(A \rightarrow B)$.

- Setting $|\perp| = \{\ast\}$ equipped with the only possible coherence relation it is clear that $A \rightarrow \perp \cong A^\perp$.
- We no longer have $A \cong A^\perp$.
- The morphisms giving the $\ast$-autonomous structure are exactly the same as in $\text{Rel}$.
A $\ast$-autonomous category of coherence spaces

Coherence spaces and coherent relations

Write $\text{Coh}$ for the category of coherence spaces, with morphisms $\text{Coh}(A, B) = \mathcal{C}(A \rightarrow B)$.

- Setting $\perp = \{\ast\}$ equipped with the only possible coherence relation it is clear that $A \rightarrow \perp \simeq A^\perp$.
- We no longer have $A \simeq A^\perp$.
- The morphisms giving the $\ast$-autonomous structure are exactly the same as in $\text{Rel}$.

Lemma

If we fix an interpretation $[X]_{\text{Coh}}$ of atoms as coherence spaces and consider the relational model where $[X]_{\text{Rel}} = |[X]_{\text{Coh}}|$, then $[\pi]_{\text{Rel}} \in \mathcal{C}(|[\Gamma]_{\text{Coh}}|)$ for all MLL proof $\pi$ of $\Gamma$. 
The previous result extends to additives: set $|A \& B| = |A \oplus B| = |A| + |B|$ and

\[
(i, c) \bowtie_{A \& B} (i', c') \iff \begin{cases} 
i = i' = 1 \text{ and } c \bowtie_A c' \\
or \quad i = i' = 2 \text{ and } c \bowtie_B c' \\
or \quad i \neq i'
\end{cases}
\]

\[
(i, c) \bowtie_{A \oplus B} (i', c') \iff \begin{cases} 
i = i' = 1 \text{ and } c \bowtie_A c' \\
or \quad i = i' = 2 \text{ and } c \bowtie_B c'
\end{cases}
\]

Now $A \& B \not\equiv A \oplus B$ and cliques correspond with our intuition:

- a clique in $A \& B$ corresponds with a pair of cliques in $A$ and $B$ respectively;
- a clique in $A \oplus B$ corresponds with either a clique in $A$ or a clique in $B$. 
Experiments in MLL proof nets

- We can play the same game as with Rel:

\[
\begin{align*}
    & \text{ax} & \text{cut} \\
    & a & a \\
    & A \perp & A \\
    & a & a \\
    & A \otimes B & B \\
    & a & b \\
    & (a,b) & (a,b) \\
    & A \otimes B & A \otimes B \\
    & a & b \\
    & A \multimap B & B \\
    & a & a \\
    & A^\perp & A
\end{align*}
\]

In fact one can show that (for a well chosen interpretation of atoms) a cut-free MLL structure is correct iff its relational semantics is clique.
Experiments in MLL proof nets

- We can play the same game as with Rel:

- But now we can detect vicious cycles:

\[(a, a) \bowtie_{A \bot \otimes A} (a', a')\] as soon as \(a \neq a' \in |A|\)
We can play the same game as with $\text{Rel}$:

But now we can detect vicious cycles:

$(a, a) \bowtie_{A_\perp \otimes A} (a', a')$ as soon as $a \neq a' \in |A|$

In fact one can show that (for a well chosen interpretation of atoms) a cut-free MLL structure is correct iff its relational semantics is clique.
Exponentials

Multiset based

- A natural choice is $|!A| = \mathcal{M}_f(|A|)$.
- Set $[a_1, \ldots, a_n] \llhd !A \llhd [a'_1, \ldots, a'_p]$ iff $a_i \llhd_A a'_j$ for all $i$ and $j$.
- But this is not reflexive!
Exponentials
Multiset based

- A natural choice is $|!A| = M_f(|A|)$.
- Set $[a_1, \ldots, a_n] \subseteq! A [a'_1, \ldots, a'_p]$ iff $a_i \subseteq_A a'_j$ for all $i$ and $j$.
- But this is not reflexive!

Multicliques

- We set $|!A|$ to be the set of all finite multicliques (finite multisets whose support is a clique).
- Then $\bar{a} \subseteq \bar{a'}$ iff $\bar{a} + \bar{a'} \in |!A|$. 

Multicliques
A natural choice is $|!A| = \mathcal{M}_f (|A|)$.

Set $[a_1, \ldots, a_n] \supset !A \ [a'_1, \ldots, a'_p]$ iff $a_i \supset_A a'_j$ for all $i$ and $j$.

But this is not reflexive!

### Multicliques

We set $|!A|$ to be the set of all finite multicliques (finite multisets whose support is a clique).

Then $\bar{a} \supset \bar{a}'$ iff $\bar{a} + \bar{a}' \in |!A|$.

$|?A|$ is the set of multicliques in $A^\perp$.

$\supset ?A$ is a strange beast.
The system of annotations now comes with side conditions:

\[
\begin{align*}
&\Gamma, A \vdash s \frac{a}{\Gamma, ?A} \quad \text{(d)}
\qquad &\Gamma \vdash s \frac{\cdot}{\Gamma, ?A} \quad \text{(w)}
\qquad &\Gamma, ?A, ?A \vdash s \frac{\overline{a} + \overline{a}'}{\overline{a} + \overline{a}' \in ?A} \quad \text{(c)}
\end{align*}
\]

\[
\begin{align*}
&\vdash ?\Gamma, A \quad \text{(for } 1 \leq i \leq n) \quad \sum_{i=1}^{n} \overline{s}_i \in ?\Gamma \\
&\vdash ?\Gamma, [a_1, \ldots, a_n] \quad \sum_{i=1}^{n} \overline{s}_i \in ?\Gamma \\
&\vdash ?\Gamma, !A
\end{align*}
\]
Coherence spaces were first introduced as a refinement of qualitative domains: particular sets of subsets.

**Cliques**

- We set $|!A|$ to be the set of all finite cliques of $A$.
- Then $\tilde{a} \subseteq \tilde{a}'$ iff $\tilde{a} \cup \tilde{a}' \in |!A|$.

- $?A$ is a still strange beast.
- And we still need annotations.
- This model is less fine but it has a close relationship with the notion of stable functions which was crucial in the invention of coherence spaces.
Orthogonality

Coherence spaces (at least for MLL) are an instance of a general construction:

- The core of the interpretation is the \(*\)-autonomous structure: composition is the interaction between a proof \(\vdash \Gamma, A\) and a counterproof \(\vdash A^\perp, \Delta\).
- We can refine a preexisting model by selecting morphisms that interact “nicely”.
Observe that if $\alpha \in \mathcal{C}(A)$ and $\alpha' \in \mathcal{C}(A^\perp)$ then $\alpha \cap \alpha'$ has at most one element.
Observe that if $\alpha \in \mathcal{C}(A)$ and $\alpha' \in \mathcal{C}(A^\perp)$ then $\alpha \cap \alpha'$ has at most one element.

Let $A$ be a set: for all $\alpha, \alpha' \subseteq A$ write $\alpha \perp \alpha'$ iff $\#(\alpha \cap \alpha') \leq 1$.

If $\mathcal{A} \subseteq \mathcal{P}(A)$, let $\mathcal{A}^\perp = \{\alpha' \subseteq A; \alpha \perp \alpha', \text{ for all } \alpha \in \mathcal{A}\}$. 
Observe that if $\alpha \in C(A)$ and $\alpha' \in C(A^\perp)$ then $\alpha \cap \alpha'$ has at most one element.

Let $A$ be a set: for all $\alpha, \alpha' \subseteq A$ write $\alpha \perp \alpha'$ iff $\#(\alpha \cap \alpha') \leq 1$.

If $\mathcal{U} \subseteq \mathcal{P}(A)$, let $\mathcal{U}^\perp = \{\alpha' \subseteq A; \ \alpha \perp \alpha', \ \text{for all} \ \alpha \in \mathcal{U}\}$.

Observe that $\alpha' \in \mathcal{U}^\perp$ iff for all $a'_1, a'_2 \in \alpha'$, $\{a'_1, a'_2\} \in \mathcal{U}^\perp$.

In other words, the sets $\mathcal{U}^\perp$ are exactly the sets of cliques of coherence spaces with web $A$. 

Observe that if $\alpha \in \mathcal{C}(A)$ and $\alpha' \in \mathcal{C}(A^\perp)$ then $\alpha \cap \alpha'$ has at most one element.

Let $A$ be a set: for all $\alpha, \alpha' \subseteq A$ write $\alpha \perp \alpha'$ iff $\#(\alpha \cap \alpha') \leq 1$.

If $\mathcal{A} \subseteq \mathcal{P}(A)$, let $\mathcal{A}^\perp = \{\alpha' \subseteq A; \alpha \perp \alpha', \text{ for all } \alpha \in \mathcal{A}\}$.

Observe that $\alpha' \in \mathcal{A}^\perp$ iff for all $a_1', a_2' \in \alpha'$, $\{a_1', a_2'\} \in \mathcal{A}^\perp$.

In other words, the sets $\mathcal{A}^\perp$ are exactly the sets of cliques of coherence spaces with web $A$.

### Characterization of morphisms

A relation $\varphi \subseteq |A| \times |B|$ is coherent iff for all $\alpha \in \mathcal{C}(A)$ and all $\beta' \in \mathcal{C}(B)^\perp$, $\varphi \cdot \alpha \perp \beta'$ and $\varphi^\perp \cdot \beta' \perp \alpha$ (where $\varphi^\perp$ is the reverse relation).
Observe that if $\alpha \in \mathcal{C}(A)$ and $\alpha' \in \mathcal{C}(A^\perp)$ then $\alpha \cap \alpha'$ has at most one element.

Let $A$ be a set: for all $\alpha, \alpha' \subseteq A$ write $\alpha \perp \alpha'$ iff $\#(\alpha \cap \alpha') \leq 1$.

If $\mathcal{A} \subseteq \mathcal{P}(A)$, let $\mathcal{A}^\perp = \{\alpha' \subseteq A; \alpha \perp \alpha', \text{ for all } \alpha \in \mathcal{A}\}$.

Observe that $\alpha' \in \mathcal{A}^\perp$ iff for all $a_1', a_2' \in \alpha'$, $\{a_1', a_2'\} \in \mathcal{A}^\perp$.

In other words, the sets $\mathcal{A}^\perp$ are exactly the sets of cliques of coherence spaces with web $A$.

**Characterization of morphisms**

A relation $\varphi \subseteq |A| \times |B|$ is coherent iff for all $\alpha \in \mathcal{C}(A)$ and all $\beta' \in \mathcal{C}(B)^\perp$, $\varphi \cdot \alpha \perp \beta'$ and $\varphi^\perp \cdot \beta' \perp \alpha$ (where $\varphi^\perp$ is the reverse relation).

The coherence space model is thus derived from the relational model by orthogonality.

This kind of construction generalizes to other notions of orthogonality, and to a generic categorical framework.
Perspectives

- Systematic categorical descriptions of models of LL exist
  - I was just lazy (see, e.g., Melliès, Panoramas et Synthèses 2009)
  - one always recover a model of λ-calculus (co-Kleisli: !A →o B)

- Characterization of computational properties
  - normalizability
  - bounds on the number of cut elimination step
  - connexion with interaction types
  - rely on models of pure λ-calculus: solve o = !o →o o

- Quantitative semantics
  - can be described explicitly in a standard setting (Ehrhard’s finiteness spaces)
  - differential linear logic
  - a guide for introducing and studying new models (see Michele Pagani’s talk during the workshop)

- Duality given by interaction
  - at the core of geometry of interaction (operational-ish semantics)
  - the basis of game semantics, a multi-purpose family of models

- ...
Have a nice LL2016

Tutorial time!