On the Lambek Calculus with an Exchange Modality

Jiaming Jiang[1], **Harley Eades III**[2], Valeria de Paiva[3]

[1]North Carolina State University; [2]Augusta University; [3]Nuance Communications

Linearity and Non-Linearity

- Girard bridged linearity with non-linearity via $!A$.
- This modality isolates the structural rules:

$$\frac{\Gamma_1, \Gamma_2 \vdash B}{\Gamma_1, !A, \Gamma_2} \textsc{weak} \qquad \frac{\Gamma_1, !A, !A, \Gamma_2 \vdash B}{\Gamma_1, !A, \Gamma_2} \textsc{contract}$$

- Linear Logic = linearity + of-course

Linearity and Non-Linearity

Linear Logic takes for granted the structural rule:

$$\frac{\Gamma_1, A, B, \Gamma_2 \vdash C}{\Gamma_1, B, A, \Gamma_2 \vdash C}\ \mathrm{Ex}$$

Lambek Calculus

- Lambek invented what we call the Lambek Calculus to give a mathematical semantics to sentence structure.

- Lambek Calculus = linearity - exchange
  - Non-commutative tensor: $A \triangleright B$
  - Non-commutative implications: $[[A < -B]]$ and $[[A- > B]]$

- No modalities

- Applications

Lambek Calculus

Question posed by computational linguists:

Can a we add a modality to the Lambek Calculus that does for exchange what of-course does for weakening and contraction?

Motivation

In process calculi, to model sequential composition of processes:

$A \otimes B$

- Commutative tensor product
- Processes A and B run in parallel

$A \triangleright B$

- Non-commutative tensor product
- Process A runs first, then process B

Basic Approach

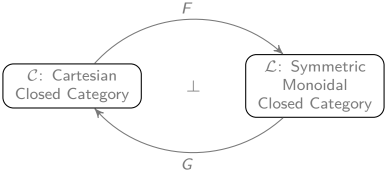
Abstract Benton's Linear/Non-Linear (LNL) model:

- Remove the exchange structural rule: implicit in $\Phi, \Psi; \Gamma, \Delta$
- Two logics:
  - Intuitionistic linear logic
  - Lambek Calculus

Linear/Non-Linear Model

A symmetric monoidal adjunction $F \dashv G$:

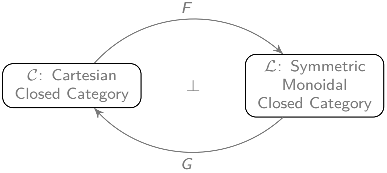
- Counit: $\varepsilon : FG \to id_{\mathcal{C}}$
- Unit: $\eta : id_{\mathcal{L}} \to GF$



Linear/Non-Linear Model

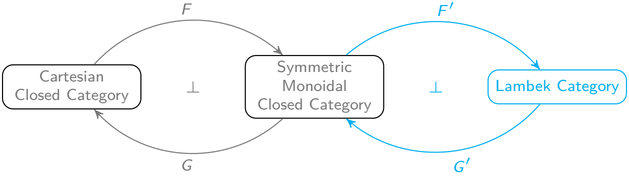
A symmetric monoidal adjunction $F \dashv G$:

- Counit: $\varepsilon : FG \to id_{\mathcal{C}}$
- Unit: $\eta : id_{\mathcal{L}} \to GF$



- Monad $(GF, \eta, \mu = G\varepsilon_F)$ on the CCC: strong and commutative
- Comonad $(FG, \varepsilon, \delta = F\eta_G)$ on the SMCC: symmetric monoidal
- Of-course modality: $! = FG$

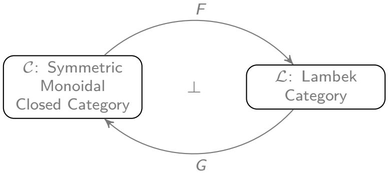
Commutative/Non-Commutative (CNC) Model



Commutative/Non-Commutative (CNC) Model

A monoidal adjunction $F \dashv G$:

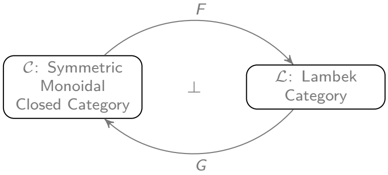
- Counit: $\varepsilon : FG \to id_{\mathcal{C}}$
- Unit: $\eta : id_{\mathcal{L}} \to GF$



Commutative/Non-Commutative (CNC) Model

A monoidal adjunction $F \dashv G$:

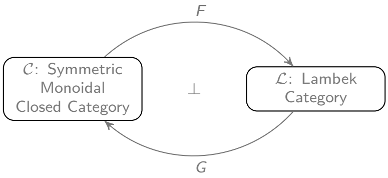
- Counit: $\varepsilon : FG \to id_{\mathcal{C}}$
- Unit: $\eta : id_{\mathcal{L}} \to GF$



- Monad $(GF, \eta, \mu = G\varepsilon_F)$ on the SMCC: strong but non-commutative
- Comonad $(FG, \varepsilon, \delta = F\eta_G)$ on the Lambek category: monoidal
- Exchange: a natural transformation $\mathrm{ex}^{FG} : A \triangleright B \to B \triangleright A$ in the co-Eilenberg-Moore category $\mathcal{L}^{FG}$ of the comonad
  $\Rightarrow$: $\mathcal{L}^{FG}$ is symmetric monoidal

CNC Logic

- Left: intuitionistic linear logic
- Right: mixed commutative/non-commutative Lambek calculus



CNC Logic: Notation

Intuitionistic Linear Logic	**Lambek Calculus**
$\mathcal{C}$-Types: W, X, Y, Z	$\mathcal{L}$-Types: A, B, C, D
$\mathcal{C}$-Terms: t	$\mathcal{L}$-Terms: s
$\mathcal{C}$-Contexts: Φ, Ψ	$\mathcal{L}$-Contexts: Γ, Δ
$\mathcal{C}$-Typing Judgment: $\Phi, \Psi \vdash_{\mathcal{C}} t : X$	$\mathcal{L}$-Typing Judgment: $\Gamma; \Delta \vdash_{\mathcal{L}} s : A$

CNC Logic: Example Typing Rules

Exchange rules:

$$\frac{\Phi, x : X, y : Y, \Psi \vdash_{\mathcal{C}} t : Z}{\Phi, z : Y, w : X, \Psi \vdash_{\mathcal{C}} \mathsf{ex}\ w, z\ \mathsf{with}\ x, y\ \mathsf{in}\ t : Z}\ \mathcal{C}\text{-ex}$$

$$\frac{\Gamma; x : X; y : Y; \Delta \vdash_{\mathcal{L}} s : A}{\Gamma; z : Y; w : X; \Delta \vdash_{\mathcal{L}} \mathsf{ex}\ w, z\ \mathsf{with}\ x, y\ \mathsf{in}\ s : A}\ \mathcal{L}\text{-ex}$$

CNC Logic: Example Typing Rules

Functor rules for G:

$$\frac{\Phi \vdash_{\mathcal{L}} s : A}{\Phi \vdash_{\mathcal{C}} \mathsf{G}s : \mathsf{G}A} \; \mathcal{C}\text{-}\mathsf{G}_I$$

$$\frac{\Phi \vdash_{\mathcal{C}} t : \mathsf{G}A}{\Phi \vdash_{\mathcal{L}} \mathsf{derelict}\; t : A} \; \mathcal{C}\text{-}\mathsf{G}_E$$

CNC Logic: Example Typing Rules

Functor rules for G:

$$\frac{\Phi \vdash_{\mathcal{L}} s : A}{\Phi \vdash_{\mathcal{C}} \mathsf{G}s : \mathsf{G}A}\ \mathcal{C}\text{-}\mathsf{G}_I$$

$$\frac{\Phi \vdash_{\mathcal{C}} t : \mathsf{G}A}{\Phi \vdash_{\mathcal{L}} \mathsf{derelict}\ t : A}\ \mathcal{C}\text{-}\mathsf{G}_E$$

Functor rules for F:

$$\frac{\Phi \vdash_{\mathcal{C}} t : X}{\Phi \vdash_{\mathcal{L}} \mathsf{F}t : \mathsf{F}X}\ \mathcal{L}\text{-}\mathsf{F}_I$$

$$\frac{\Gamma \vdash_{\mathcal{L}} s_1 : \mathsf{F}X \qquad \Delta_1; x : X; \Delta_2 \vdash_{\mathcal{L}} s_2 : A}{\Delta_1; \Gamma; \Delta_2 \vdash_{\mathcal{L}} \mathsf{let}\ s_1 : \mathsf{F}X\ \mathsf{be}\ \mathsf{F}x\ \mathsf{in}\ s_2 : A}\ \mathcal{L}\text{-}\mathsf{F}_E$$

CNC Logic: Other Results

- β-reductions: one step β-reduction rules
- Commuting conversions
- Cut elimination
- Equivalence between sequent calculus and natural deduction
- Strong normalization via a translation to LNL logic
- A concrete model in dialectica categories

Conclusion

- Commutative/Non-commutative Logic:
  - Left: intuitionistic linear logic
  - Right: Lambek calculus
- Categorical model: a monoidal adjunction
  - Left: symmetric monoidal closed category
  - Right: Lambek category

Exchange Natural Transformation

$\mathrm{ex}^{FG} : A \triangleright B \to B \triangleright A$ in the co-Eilenberg-Moore category $\mathcal{L}^{FG}$ of the comonad on the Lambek category