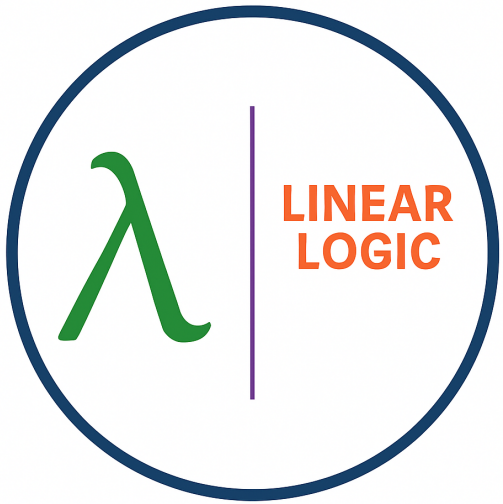
LAMBDA CALCULI THROUGH THE LENS OF LINEAR LOGIC

Delia KESNER
Email : kesner@irif.fr
URL : www.irif.fr/~kesner

Université Paris Cité and CNRS

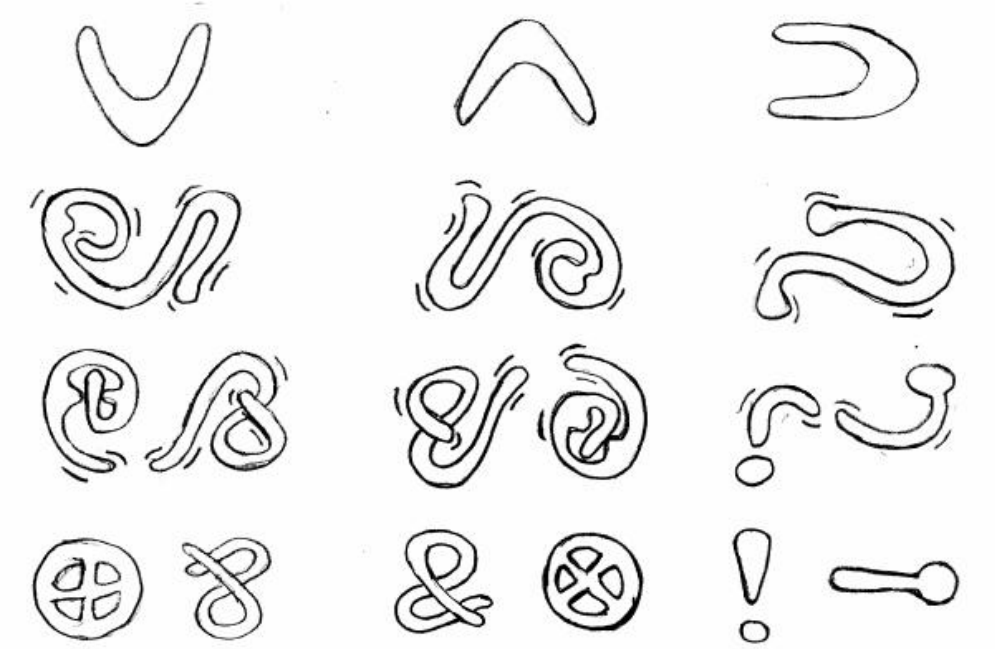


Why Lambda Calculus?

- **A foundational model of computation**
  - Origin of functional programming languages (e.g. Haskell, OCaml, Scala)
  - Central to the theory of programming languages and proof theory
- **Minimal syntax, maximal expressiveness**
  - Captures computation via substitution and β-reduction
  - Basis for type systems, evaluation strategies, and semantics
- **Still a fertile ground for research**
  - New hints about call-by-name, call-by-value and call-by-need operational semantics.
  - Interaction with category theory, game semantics, rewriting systems, etc.

LINEAR LOGIC

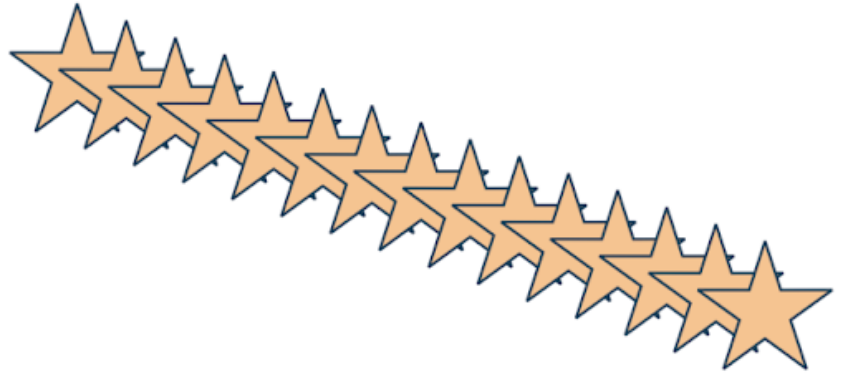
Jean-Yves Girard



Resource aware logic

No Weakening (No Erasure)

No Contraction (No Duplication)



Two Exponential Modalities:

Why not	Of course/Bang
$?$	$!$

Intuition: formulas must be marked with exponentials to be erased/weakened or duplicated/contracted.

Multiplicative Exponential Linear Logic (MELL)

Why Linear Logic?

- **Refines intuitionistic and classical logic**
  - Controls duplication and erasure
  - Naturally captures resource-sensitive computation
- **Unlocks new insights into computation**
  - Proof-nets: a graphical syntax
  - Call-by-name vs call-by-value as logical phenomena
  - Implicit complexity and cost models
- **Provides a fruitful perspective for revisiting old ideas**
  - Offers a fine-grained look at evaluation and typing
  - Inspires new calculi and type systems

Goals of the Course

- **Explore lambda calculi variants inspired by linear logic**
  - Syntax, semantics, and operational properties
  - Fine-grained (implementation) calculi
  - Subsuming frameworks (capturing different models of computation)
- **Understand the structure of computation through linear-logic types**
  - Intersection/Quantitative types
- **Provide resource aware tools for modern theoretical research**
  - Observational equivalence
  - Inhabitation
  - Genericity

Course Outline

- Day 1: Linear Logic Proof-Nets
- Day 2: A Lambda-Calculus Inspired from Linear Logic Proof-Nets
- Day 3: Intersection/Quantitative Types
- Day 4: A Subsuming Framework Inspired from Linear Logic
- Day 5: Observational Equivalence By Means of Intersection Types