Towards a term syntax for L-nets

Pablo Donato

Université Paris Diderot

April 1, 2020

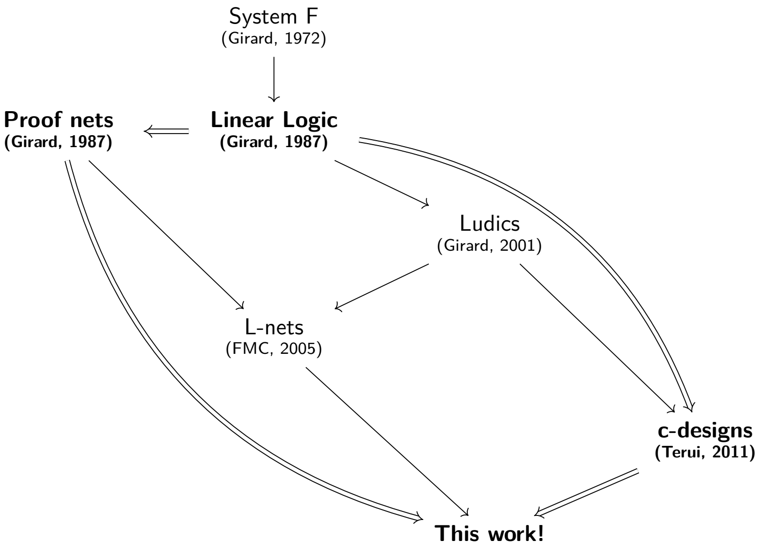
Encadré par Alexis Saurin

Curry-Howard isomorphism

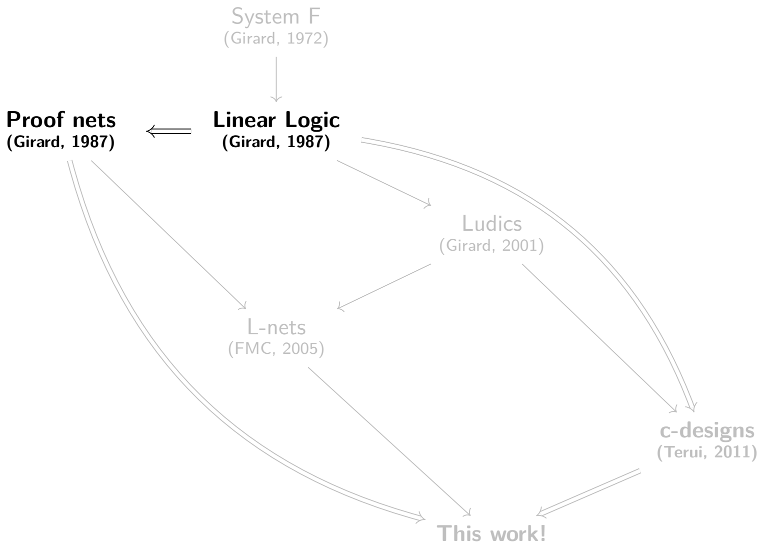
proof	$\Longleftrightarrow$	**program**
formula	$\Longleftrightarrow$	**type**
normalization/cut elimination	$\Longleftrightarrow$	**evaluation/execution**

- Initially discovered for **minimal intuitionistic** natural deduction and **simply-typed λ-calculus** (Howard, 1969)
- Extended to **2nd-order intuitionistic** natural deduction and **polymorphic λ-calculus/System F** (Girard-Reynold, 1972)

Motivations



Multiplicative linear logic



Linear logic

$$\begin{array}{rll} A ::= & X \mid X^\perp \mid & \text{(Atoms)} \\ & \mathbf{1} \mid \perp \mid A \otimes A \mid A \parr A \mid & \text{(Multiplicatives)} \\ & \mathbf{0} \mid \top \mid A \oplus A \mid A \,\&\, A \mid & \text{(Additives)} \\ & !A \mid ?A & \text{(Exponentials)} \end{array}$$

- Fine management of formulae as **resources**
- Can encode both **classical** and **intuitionistic** logic
- Exhibits features of both worlds:
  - **Disjunction property** for $\oplus$
  - **Involutive negation** $\cdot^\perp$:

$$A = A^{\perp\perp}$$

Sequents

- Formulae are generated by the following **polarized** grammar:

$$
\begin{aligned}
P &::= X \mid N \otimes N \mid {\downarrow} N & \textbf{(positive)} \\
N &::= X^\perp \mid P \parr P \mid {\uparrow} P & \textbf{(negative)}
\end{aligned}
$$

- Negation defined by **De Morgan** equations on dual connectives:

$$
\begin{array}{lll}
(X)^\perp = X^\perp & (N_1 \otimes N_2)^\perp = N_1^\perp \parr N_2^\perp & ({\downarrow} N)^\perp = {\uparrow} N^\perp \\
(X^\perp)^\perp = X & (P_1 \parr P_2)^\perp = P_1^\perp \otimes P_2^\perp & ({\uparrow} P)^\perp = {\downarrow} P^\perp
\end{array}
$$

- A **focalized sequent** is a multiset of formulae of the form:

$$\vdash A, P_1, \ldots, P_n \quad \text{also written} \quad \vdash A, \Gamma$$

Proofs

- A **proof** of a sequent is just a **derivation tree**
- Leaves are **axiom** rules:

$$\dfrac{}{\vdash P^{\perp}, P}\ \text{ax}$$

- Nodes are either **cut** rules or **logical** rules:

$$\dfrac{\vdash P^{\perp}, \Gamma \quad \vdash P, \Delta}{\vdash \Gamma, \Delta}\ \text{cut}$$

Logical rules

They define how to prove **compound** formulae:

$$\frac{\vdash N_1, \Gamma \quad \vdash N_2, \Delta}{\vdash N_1 \otimes N_2, \Gamma, \Delta} \otimes \qquad \frac{\vdash P_1, P_2, \Gamma}{\vdash P_1 \parr P_2, \Gamma} \parr$$

$$\frac{\vdash N, \Gamma}{\vdash {\downarrow} N, \Gamma} \downarrow \qquad \frac{\vdash P, \Gamma}{\vdash {\uparrow} P, \Gamma} \uparrow$$

Cut elimination

- A **rewriting** procedure that *eliminates* cut rules from proofs
- Through Curry-Howard, can be seen as an **evaluation strategy**
- Defined as the iteration of **cut reduction** steps:

$$\dfrac{\dfrac{}{\vdash P^{\perp}, P}\,\mathrm{ax} \qquad \begin{array}{c}\vdots\ \pi \\ \vdash P, \Gamma\end{array}}{\vdash P, \Gamma}\,\mathrm{cut} \quad \rightsquigarrow \quad \begin{array}{c}\vdots\ \pi \\ \vdash P, \Gamma\end{array}$$

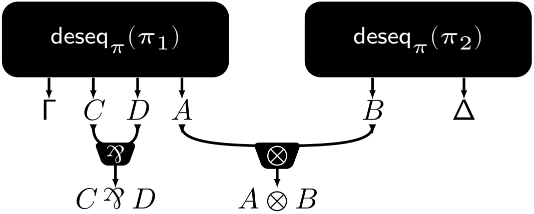
Permutation of rules

- Cut elimination **terminates**
- But it is not **confluent**
- This gives rise to artificial **orderings** on rules:

$$\dfrac{\dfrac{\dfrac{\vdots\ \pi_1}{\vdash A, C, D, \Gamma} \quad \dfrac{\vdots\ \pi_2}{\vdash B, \Delta}}{\vdash A \otimes B, C, D, \Gamma, \Delta}\ \otimes}{\vdash A \otimes B, C \parr D, \Gamma, \Delta}\ \parr \qquad \dfrac{\dfrac{\dfrac{\vdots\ \pi_1}{\vdash A, C, D, \Gamma}}{\vdash A, C \parr D, \Gamma}\ \parr \quad \dfrac{\vdots\ \pi_2}{\vdash B, \Delta}}{\vdash A \otimes B, C \parr D, \Gamma, \Delta}\ \otimes$$

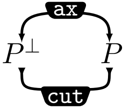
Proof structures

- To *equate* proofs which only differ by permutation of rules, Girard devised a new syntax of **proof structures**, which are **graphs** rather than **trees**
- The **desequentialization** function deseq_π maps sequent calculus proofs to their equivalent proof structures
- The two preceding examples collapse to:



Proof nets

- **Proof nets** are the image of desequentialization deseq_π
- Equivalently, they are the **sequentializable** proof structures
- **Cut elimination** on proof nets **terminates** and is **confluent**
- **Not all** proof structures are sequentializable:



Contradicts termination of cut elimination!

Correctness criterions

A way of checking the **correctness** of a proof structure without having to guess one if its underlying sequent calculus proofs!

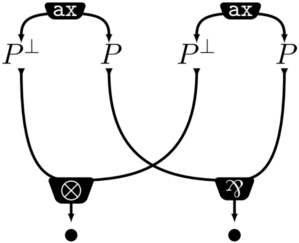
Theorem (Sequentialization)

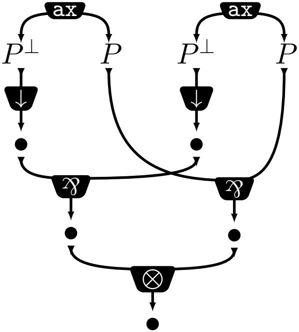
A proof structure $\mathfrak{S}$ *is a proof net* iff *it satisfies (all) correctness criterions.*

The DR criterion

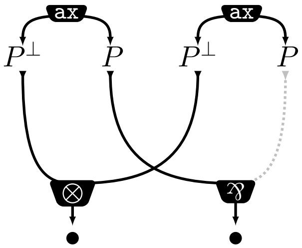
- A **switching graph** of $\mathfrak{S}$ is $\mathfrak{S}$ where every $\parr$-node has one of its premisses *erased*
- $\mathfrak{S}$ is $\mathbf{DR}$**-correct** if all its switching graphs are **connected** and **acyclic**

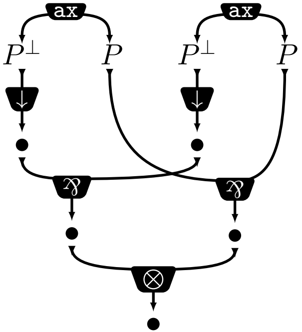
Examples


ax
ax
$P^\perp$
P
$P^\perp$
P
$\otimes$
$⅋$

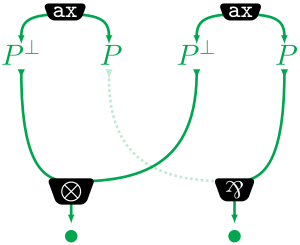

ax
ax
$P^\perp$
P
$P^\perp$
P
$\downarrow$
$\downarrow$
$⅋$
$⅋$
$\otimes$

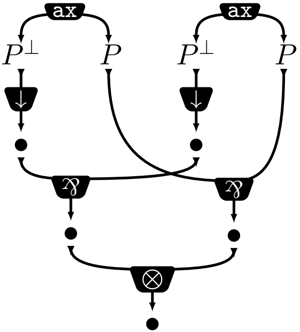
Examples


ax
ax
$P^\perp$
P
$P^\perp$
P
$\otimes$
⅋

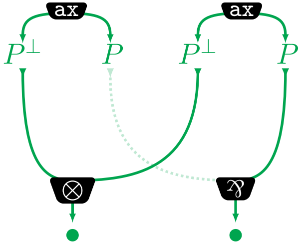

ax
ax
$P^\perp$
P
$P^\perp$
P
$\downarrow$
$\downarrow$
⅋
⅋
$\otimes$

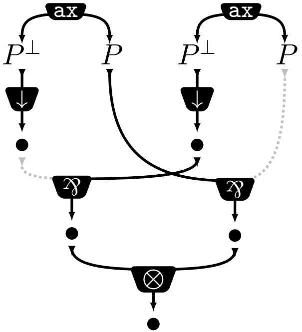
Examples


ax
ax
$P^\perp$
P
$P^\perp$
P
$\otimes$
⅋

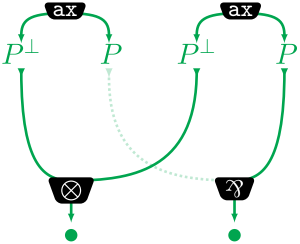

ax
ax
$P^\perp$
P
$P^\perp$
P
$\downarrow$
$\downarrow$
⅋
⅋
$\otimes$

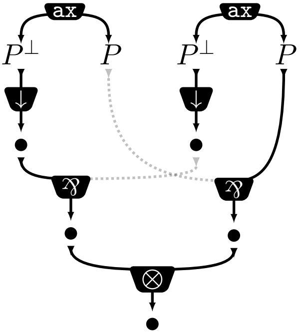
Examples



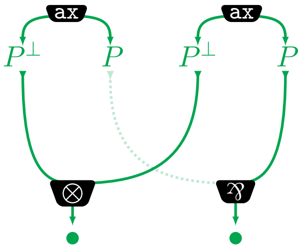


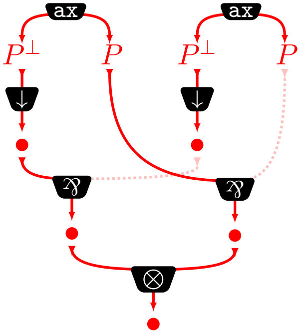
Examples


ax
ax
$P^\perp$
P
$P^\perp$
P
$\otimes$
$⅋$

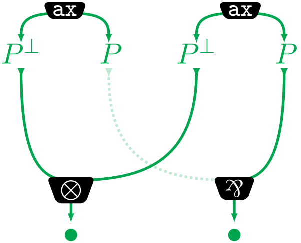

ax
ax
$P^\perp$
P
$P^\perp$
P
$\downarrow$
$\downarrow$
$⅋$
$⅋$
$\otimes$

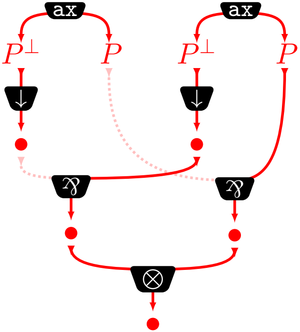
Examples



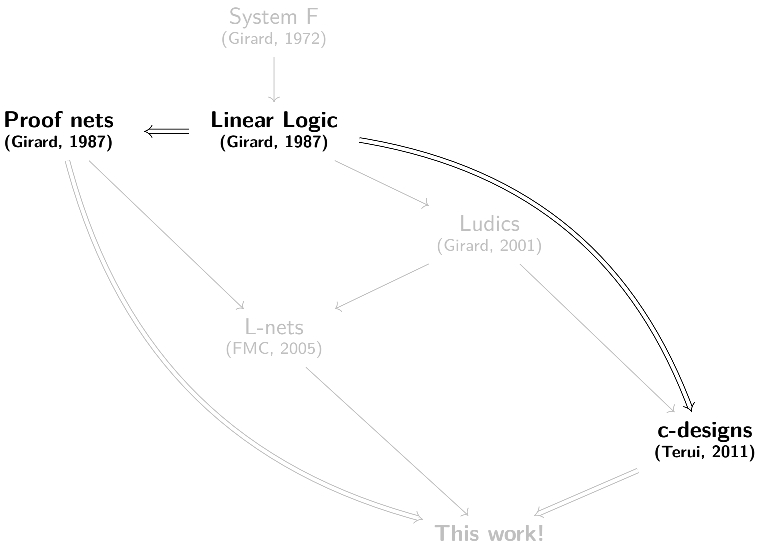


Examples


ax
ax
$P^\perp$
P
$P^\perp$
P
$\otimes$
⅋


ax
ax
$P^\perp$
P
$P^\perp$
P
$\downarrow$
$\downarrow$
⅋
⅋
$\otimes$

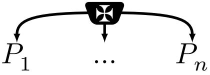
Towards ludics



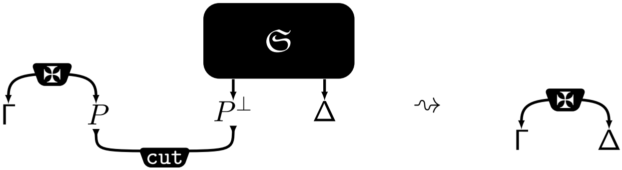
CP criterion

- **Idea:** $\mathfrak{S}$ proves A if it wins against every **counterproof**, that is every proof of $A^\perp$
- **Problem:** Having proofs of A *and* $A^\perp$ would make our logic **inconsistent**!
- Well, if we can still characterize correct proofs... Let's try a new kind of axiom, the **daimon** ✠:

$$\dfrac{}{\vdash P_1, \dots, P_n}\ \maltese$$



Cut elimination



- A proof that uses $\maltese$ is called a **paraproof**
- $\maltese$ **absorbs** $\mathfrak{S}$ instead of keeping it like ax
- $\mathfrak{S} \perp \mathfrak{S}'$ if cutting their dual conclusions evaluates to $\maltese$

Correctness

CP criterion

$\mathfrak{S}$ is CP-correct if for every **counterproof** $\mathfrak{S}'$, $\mathfrak{S} \perp \mathfrak{S}'$.

DR criterion

$\mathfrak{S}$ is DR-correct if for every **switching** $\mathfrak{S}'$, $\mathfrak{S}'$ connected and acyclic.

In fact, DR is just a **static** reformulation of CP:

- **Termination** of interaction is replaced by **acyclicity**
- **Uniqueness** of result (one ✠) is replaced by **connectedness**

Multiplicative c-designs

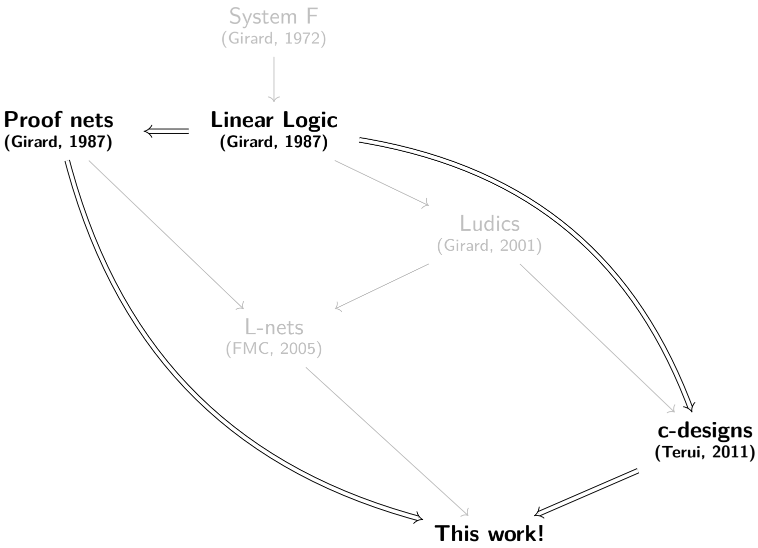
$$
\begin{aligned}
P &::= \maltese(\vec{x}) \mid \Omega \mid N_0 \parallel \otimes(N_1, N_2) \mid N_0 \parallel \downarrow(N_1) \\
N &::= x \mid \parr(x_1, x_2).P \mid \uparrow(x).P
\end{aligned}
$$

- **Specialization** of Terui's c-designs, introduced as a **term syntax** for exporting **ludics** to computability/complexity theory:

$$\text{M recognizes } \mathcal{L} \quad \Leftrightarrow \quad \mathfrak{D}_{\mathsf{M}} \perp \mathfrak{D}_{\mathcal{L}}$$

- Used here to encode sequent calculus paraproofs
- **Co-inductive** definition $\Rightarrow$ might be **infinite**!

Desequentialization of terms



Paranets

- A term syntax for **paraproof structures**
- Inspired by the **differential interaction nets** of D. Mazza (2016)
- A **cell** is an expression of one of the following forms:

  daimon: $\maltese(\vec{x})$, $\mathrm{gc}(\vec{x};\vec{y})$ **multiplicative cells:** $\otimes(x;y,z)$, $\parr(x;y,z)$

  box: $\mathrm{box}(\vec{x};\vec{x}')$ **shift cells:** $\downarrow(x;y)$, $\uparrow(x;y)$

- **Ports** x, y, z are **free** (resp. **bound**) when they occur *once* (resp. *twice*)
- A **paranet** is a multiset of cells and **wires** $x \leftrightarrow y$, with multiset union denoted by $|$

cell	$\Longleftrightarrow$	**node**
bound port/wire	$\Longleftrightarrow$	**edge**
free port	$\Longleftrightarrow$	**conclusion**

Desequentialization

This is a **two-step** procedure:

$$\mathcal{D} \xrightarrow{\quad \mathsf{B} \quad} \mathcal{B} \xrightarrow{\quad \mathsf{deseq}_{\mu} \quad} \mathcal{N}$$

1. **Traduction** B from multiplicative c-designs to paranets
2. **Removal of boxes** through rewriting steps deseq_{μ}. Two variants:
   - deseq_{μ}^{n} ("big-step"): remove entire boxes
   - deseq_{μ}^{1} ("small-step"): remove wire by wire

Correction

$$\begin{tikzcd}
\mathsf{MLL}\maltese \arrow[rr, "\mathsf{deseq}_\pi"] \arrow[d, "\mathsf{D}"] & & \mathsf{PN}_{\mathsf{MLL}\maltese} \arrow[d, "\mathsf{N}"] \\
\mathcal{D} \arrow[r, "\mathsf{B}"] \arrow[d, "\rightsquigarrow_{\mathcal{D}}"] & \mathcal{B} \arrow[r, "\mathsf{deseq}_\mu"] \arrow[d, "\rightsquigarrow_{\mathcal{B}}"] & \mathcal{N} \arrow[d, "\rightsquigarrow_{\mathcal{B}}"] \\
\mathcal{D} \arrow[r, "\mathsf{B}"] & \mathcal{B} \arrow[r, "\mathsf{deseq}_\mu"] & \mathcal{N}
\end{tikzcd}$$

- **Static** correction (top): desequentialization of terms simulates desequentialization of proofs
- **Dynamic** correction (bottom): desequentialization of terms commutes with cut elimination on terms

Conclusion

What we have done

- Introduced and related sequential, parallel and interactive proof systems for multiplicative linear logic
- Designed and introduced term syntaxes for those systems
- Related the sequential and parallel syntaxes through desequentialization

Future work

- Proving that our desequentialization is correct
- Importing results such as correctness criterions in our syntax
- Extending our syntax to the additive fragment of linear logic, and abstracting away from connectives: this would lead us to L-nets

Bibliography