Remarks on proof nets as combinatorial maps

Lê Thành Dũng (Tito) Nguyễn (CNRS & Aix-Marseille Univ.)

DIAgrams in LOgic and COmputation @ FLoC 2026, July 19 (Marseille $\xrightarrow{\text{online}}$ Lisboa)

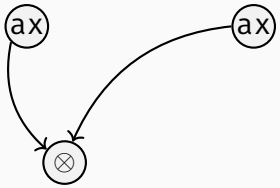
The simplest setting: MLL proof nets

Multiplicative Linear Logic (MLL): $A, B ::= X \mid X^{\perp} \mid A \otimes B \mid A \parr B$

A *proof net* is a sort of graph made of ax, $\otimes$ and $\parr$ links which represents a proof

- i.e. translated from a sequent calculus proof
- Equivalently, set of proof nets inductively generated

$$\dfrac{\dfrac{}{\vdash A, A^{\perp}}\,\mathsf{ax} \qquad \dfrac{}{\vdash B, B^{\perp}}\,\mathsf{ax}}{\vdash A \otimes B, A^{\perp}, B^{\perp}}\,\otimes$$



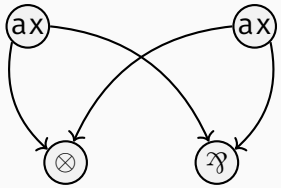
The simplest setting: MLL proof nets

Multiplicative Linear Logic (MLL): $A, B ::= X \mid X^\perp \mid A \otimes B \mid A \parr B$

A *proof net* is a sort of graph made of ax, $\otimes$ and $\parr$ links which represents a proof

- i.e. translated from a sequent calculus proof
- Equivalently, set of proof nets inductively generated

$$\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,\mathsf{ax} \qquad \dfrac{}{\vdash B, B^\perp}\,\mathsf{ax}}{\vdash A \otimes B, A^\perp, B^\perp}\,\otimes}{\vdash A \otimes B, A^\perp \parr B^\perp}\,\parr$$



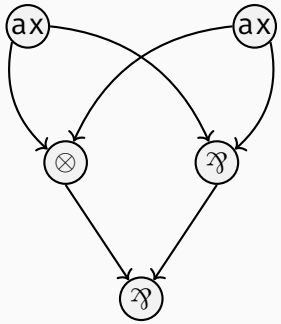
The simplest setting: MLL proof nets

Multiplicative Linear Logic (MLL): $A, B ::= X \mid X^\perp \mid A \otimes B \mid A \parr B$

A *proof net* is a sort of graph made of ax, $\otimes$ and $\parr$ links which represents a proof

- i.e. translated from a sequent calculus proof
- Equivalently, set of proof nets inductively generated

$$\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,\mathsf{ax} \qquad \dfrac{}{\vdash B, B^\perp}\,\mathsf{ax}}{\vdash A \otimes B, A^\perp, B^\perp}\,\otimes}{\vdash A \otimes B, A^\perp \parr B^\perp}\,\parr}{\vdash (A \otimes B) \parr (A^\perp \parr B^\perp)}\,\parr$$



Proof nets vs proof structures

Proof structures: graphs made of `ax`-links, $\otimes$-links and $\parr$-links

- Proof structures $\supsetneq$ proof nets!
  Some are not images of any sequent calculus proof

Problem (Correctness)

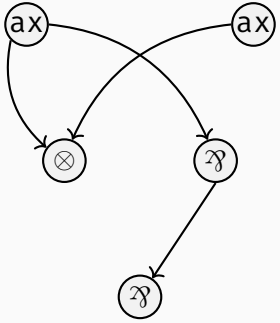
Given a proof structure, decide whether it is a proof net.

Related to *correctness criteria*: non-inductive combinatorial characterizations of proof nets among proof structures

A correctness criterion for MLL

Most common criterion: Danos & Regnier 1989, *The structure of multiplicatives*

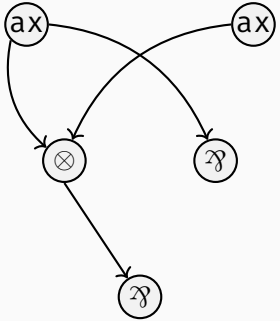
- Delete 1 of the 2 premises of each $\parr$-link;
  do you always get an (undirected) *tree*, i.e. an *acyclic* and *connected* graph?
- If so, then you've got an MLL proof net



A correctness criterion for MLL

Most common criterion: Danos & Regnier 1989, *The structure of multiplicatives*

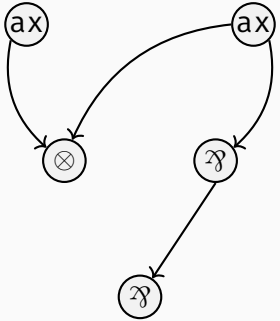
- Delete 1 of the 2 premises of each $\parr$-link;
  do you always get an (undirected) *tree*, i.e. an *acyclic* and *connected* graph?
- If so, then you've got an MLL proof net



A correctness criterion for MLL

Most common criterion: Danos & Regnier 1989, *The structure of multiplicatives*

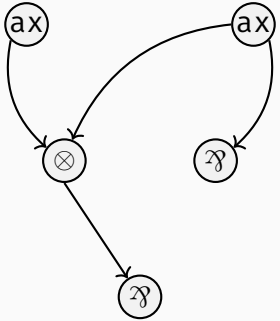
- Delete 1 of the 2 premises of each $\parr$-link;
  do you always get an (undirected) *tree*, i.e. an *acyclic* and *connected* graph?
- If so, then you've got an MLL proof net



A correctness criterion for MLL

Most common criterion: Danos & Regnier 1989, *The structure of multiplicatives*

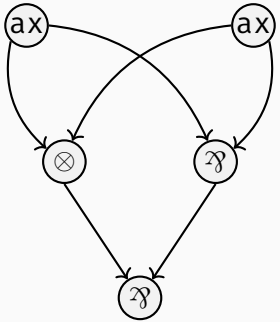
- Delete 1 of the 2 premises of each $\parr$-link;
  do you always get an (undirected) *tree*, i.e. an *acyclic* and *connected* graph?
- If so, then you've got an MLL proof net



A correctness criterion for MLL

Most common criterion: Danos & Regnier 1989, *The structure of multiplicatives*

- Delete 1 of the 2 premises of each $\parr$-link;
  do you always get an (undirected) *tree*, i.e. an *acyclic* and *connected* graph?
- If so, then you've got an MLL proof net



Methodological discussion: complexity, algorithms, graphs (1)

- "The Danos–Regnier correctness criterion has exponential time complexity, and linear-time criteria came later [Guerrini 1999; Murawski & Ong 2000]"

Methodological discussion: complexity, algorithms, graphs (1)

- "The Danos–Regnier correctness criterion has exponential time complexity, and linear-time criteria came later [Guerrini 1999; Murawski & Ong 2000]"

Point 1: Correctness criteria don't have complexities, algorithms do

- "The Danos–Regnier criterion admits a naive exponential-time algorithm, as well as a clever linear-time algorithm" (bonus: also for MLL+Mix!)

Methodological discussion: complexity, algorithms, graphs (1)

- "The Danos–Regnier correctness criterion has exponential time complexity, and linear-time criteria came later [Guerrini 1999; Murawski & Ong 2000]"

Point 1: Correctness criteria don't have complexities, algorithms do

- "The Danos–Regnier criterion admits a naive exponential-time algorithm, as well as a clever linear-time algorithm" (bonus: also for MLL+Mix!)

  $\rightsquigarrow$ Gabow, Kaplan & Tarjan 1999, *Unique Maximum Matching Algorithms*

  (details in my LMCS 2020 paper)

Point 2: To study proof nets, learn about graph theory and algorithmics

Methodological discussion: complexity, algorithms, graphs (1)

- "The Danos–Regnier correctness criterion has exponential time complexity, and linear-time criteria came later [Guerrini 1999; Murawski & Ong 2000]"

Point 1: Correctness criteria don't have complexities, algorithms do

- "The Danos–Regnier criterion admits a naive exponential-time algorithm, as well as a clever linear-time algorithm" (bonus: also for MLL+Mix!)

  $\rightsquigarrow$ Gabow, Kaplan & Tarjan 1999, *Unique Maximum Matching Algorithms*

  (details in my LMCS 2020 paper)

Point 2: To study proof nets, learn about graph theory and algorithmics

- Precursor: Retoré in the 1990s
- N. & Straßburger 2023, *The Complexity of BV and Pomset Logic*
- Di Guardia, Laurent, Tortora de Falco & Vaux Auclair 2025, *Yeo's Theorem for Locally Colored Graphs: the Path to Sequentialization in Linear Logic*

Methodological discussion: complexity, algorithms, graphs (2)

1. **Correctness criteria don't have complexities, algorithms do**
2. **To study proof nets, learn about graph theory and algorithmics**

Let us apply this to *cyclic MLL*, a non-commutative logic:

Methodological discussion: complexity, algorithms, graphs (2)

1. **Correctness criteria don't have complexities, algorithms do**
2. **To study proof nets, learn about graph theory and algorithmics**

Let us apply this to *cyclic MLL*, a non-commutative logic:

> *Correctness of* [*cyclic MLL*] *proof structures, restricted to those ones that only allow a cut-free sequentialization, is quadratic in the size of the input proof structure* (*Mogbil, 2001*). *As future work we aim at classifying the complexity class of the proposed new correctness criterion.* — Abrusci & Maieli 2019

Methodological discussion: complexity, algorithms, graphs (2)

1. **Correctness criteria don't have complexities, algorithms do**
2. **To study proof nets, learn about graph theory and algorithmics**

Let us apply this to *cyclic MLL*, a non-commutative logic:

> *Correctness of* [*cyclic MLL*] *proof structures, restricted to those ones that only allow a cut-free sequentialization, is quadratic in the size of the input proof structure* (*Mogbil, 2001*). *As future work we aim at classifying* ~~*the complexity class of the proposed new correctness criterion*~~*.* — Abrusci & Maieli 2019

Methodological discussion: complexity, algorithms, graphs (2)

1. **Correctness criteria don't have complexities, algorithms do**
2. **To study proof nets, learn about graph theory and algorithmics**

Let us apply this to *cyclic MLL*, a non-commutative logic:

> *Correctness of* [*cyclic MLL*] *proof structures, restricted to those ones that only allow a cut-free sequentialization, is quadratic in the size of the input proof structure* (*Mogbil, 2001*). *As future work we aim at classifying* ~~*the complexity class of the proposed new correctness criterion*~~*.* — Abrusci & Maieli 2019

Today's theorem: an immediate corollary of [Melliès 2004]

Correctness of cyclic MLL proof structures with cuts is decidable in linear time.

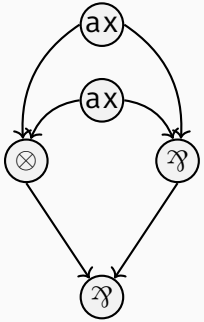
Cyclic MLL proof nets (1)

Sequent calculus for cyclic MLL: non-commutativity

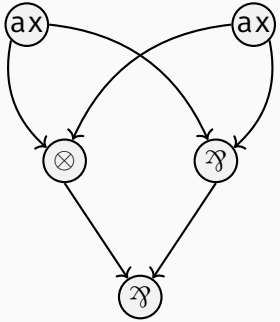
replace exchange $\dfrac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, B, A, \Delta}$ by $\dfrac{\vdash \Gamma, \Delta}{\vdash \Delta, \Gamma}$ cyclic exchange

Proof nets "drawn on the plane without crossings":

in cyMLL:



not in cyMLL:



Cyclic MLL proof nets (2)

Proof nets "drawn on the plane", but the notion of *planar graph* is insufficient:
both graphs on the previous slide are the same...

From Nagayama & Okada 2003[1] (emphasis mine):

> *Definition 3.2. A marked D–R graph drawing is said to be uniformly directed if the L-edge, R-edge and C-edge for a link is* drawn in a fixed cyclic order uniformly *for all tensor-links and par-links, or the links of degree 3.*

$\longrightarrow$ we must consider graphs endowed with a *rotation system*:
for each vertex, a cyclic order on its incident edges.
(This order is different for our two examples.)

[1] *A graph-theoretic characterization theorem for multiplicative fragment of non-commutative linear logic*

Combinatorial maps

Definition

Combinatorial map = connected undirected graph + rotation system
(for each vertex, a cyclic order on its incident edges)

Combinatorial maps

Definition

Combinatorial map = connected undirected graph + rotation system
(for each vertex, a cyclic order on its incident edges)

Theorem: Heffter–Edmonds–Ringel principle

Combinatorial maps $\cong$ homeo. classes of *cellular embeddings of graphs on surfaces*
(cellular = *faces* of the embedding are disks)

Planar map = the surface is a sphere (compactified plane)
$\longrightarrow$ cyclic MLL proof nets must be planar maps

Digression: combinatorial maps vs linear λ-terms

Combinatorial maps appear in the study of the *enumerative and bijective combinatorics of untyped λ-terms*

Began in the 2010s, still ongoing, e.g. Bodini, Singh & Zeilberger 2025

family of lambda terms	family of rooted maps	OEIS
linear	3-valent (of genus $g \geq 0$)	A062980
planar	planar 3-valent	A002005
unitless linear	bridgeless 3-valent ($g \geq 0$)	A267827
unitless planar	bridgeless planar 3-valent	A000309
β-normal linear/$\sim$	(all maps of genus $g \geq 0$)	A000698
β-normal planar	planar	A000168
β-normal unitless linear/$\sim$	bridgeless ($g \geq 0$)	A000699
β-normal unitless planar	bridgeless planar	A000260

Table 1. Known correspondences [3, 7, 53–56] between families of lambda terms and rooted maps, as combinatorial classes. Here "$\sim$" stands for an equivalence relation defined in [53], and the rest of the terminology is explained in Sections 3 and 4.1. (Indices on the right refer to the Online Encyclopedia of Integer Sequences [34].)

(table from Zeilberger 2018, *A theory of linear typings as flows on 3-valent graphs*)

A correctness criterion for cyclic MLL (1)

For *cut-free* proof structures: MLL correctness + planarity = cyMLL correctness.

Planarity of a combinatorial map can be decided in linear time

Count *faces* as "orbits" from rotation system $\rightsquigarrow$ Euler characteristic

$\longrightarrow$ Cut-free cyclic MLL correctness in linear time.

A correctness criterion for cyclic MLL (1)

For *cut-free* proof structures: MLL correctness + planarity = cyMLL correctness.

Planarity of a combinatorial map can be decided in linear time

Count *faces* as "orbits" from rotation system $\rightsquigarrow$ Euler characteristic

$\longrightarrow$ Cut-free cyclic MLL correctness in linear time. Extension to cuts:

The Melliès (2004) correctness criterion, rephrased for combinatorial maps

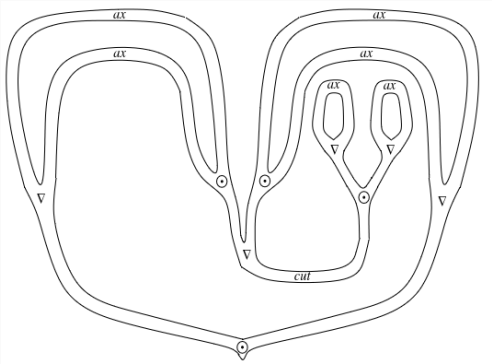
A proof structure with cuts is a cyclic MLL proof net iff

- it is an MLL proof net and a planar map
- 1 *face* contains all its conclusions, but no upper corner of a $\parr$-link

(also testable in linear time!)

A correctness criterion for cyclic MLL (2)

Melliès (2004) draws proof nets as *ribbons* and gives a correctness criterion involving their *borders* (connected comp. of boundary)



(with $\odot = \otimes$ and $\nabla = \parr$)

A correctness criterion for cyclic MLL (2)

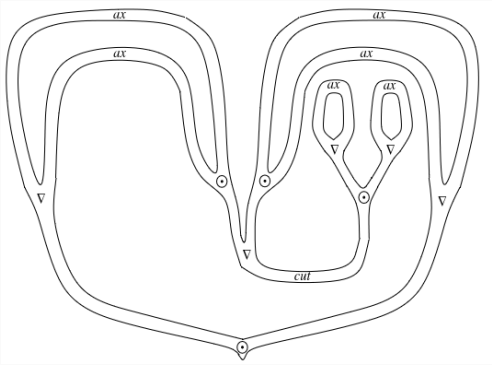
Melliès (2004) draws proof nets as *ribbons* and gives a correctness criterion involving their *borders* (connected comp. of boundary)

Amounts to the same thing:

$$\begin{pmatrix}\text{graphs on}\\ \text{surfaces}\end{pmatrix} \to \text{ribbons}$$

G on $S \mapsto \varepsilon$-neighborhood of G in S

faces $\mapsto$ borders

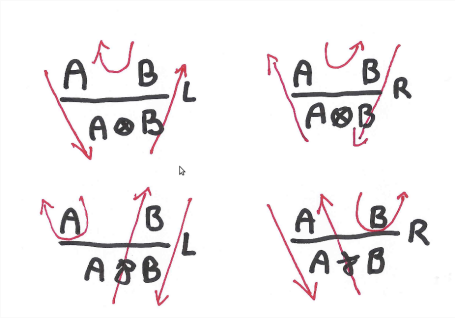


(with $\odot = \otimes$ and $\nabla = \parr$)

Long trip switchings as embedded graphs (1)

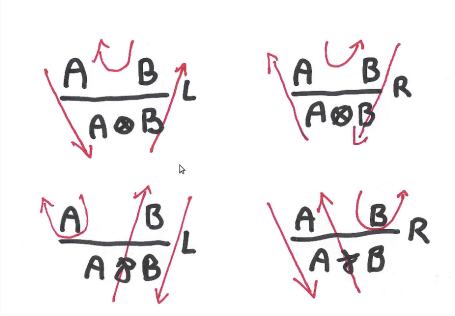
Let's apply combinatorial maps to usual (commutative) MLL.
Girard's original *long trip* correctness criterion:

- On each $\otimes$-link and $\parr$-link, choose 1 of 2 possible set of routing instructions around the link
- Is the orbit a single cycle?



Long trip switchings as embedded graphs (1)

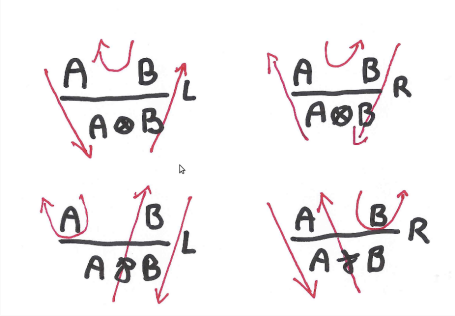
Let's apply combinatorial maps to usual (commutative) MLL.
Girard's original *long trip* correctness criterion:



A long trip is a way to travel around a Danos–Regnier tree… But what does the routing around a $\otimes$-link mean?

Long trip switchings as embedded graphs (1)

Let's apply combinatorial maps to usual (commutative) MLL.
Girard's original *long trip* correctness criterion:



A long trip is a way to travel around a Danos–Regnier tree… But what does the routing around a $\otimes$-link mean? It's a choice of *rotation system*! Trips are *faces*.

Long trip switchings as embedded graphs (2)

Routing choice = rotation system = turns graph into combinatorial map

$\longrightarrow$ the equivalence of the long trips and Danos–Regnier criteria is an instance of:

Topologically obvious property

A connected graph is a tree if and only if all its cellular embeddings on surfaces have a single face.

Long trip switchings as embedded graphs (2)

Routing choice = rotation system = turns graph into combinatorial map

$\longrightarrow$ the equivalence of the long trips and Danos–Regnier criteria is an instance of:

Topologically obvious property

A connected graph is a tree if and only if all its cellular embeddings on surfaces have a single face.

Girard's *The Blind Spot* (§11.3.6): the long trip criterion "remains irreplaceable in the case of non-commutative logic"

- perhaps not the most enlightening POV, actually…
- but combinatorial maps shed light on why long trips work for non-commutative variants of MLL (cf. Abrusci & Ruet)

Take-away messages

1. **Correctness criteria don't have complexities, algorithms do**
2. **To study proof nets, learn about graph theory and algorithmics**
   - including *graphs embedded on surfaces* = *combinatorial maps*, especially (but not only) for cyclic / non-commutative MLL

Take-away messages

1. **Correctness criteria don't have complexities, algorithms do**
2. **To study proof nets, learn about graph theory and algorithmics**
   - including *graphs embedded on surfaces* = *combinatorial maps*, especially (but not only) for cyclic / non-commutative MLL
3. Girard is not always right ;p

1. **Correctness criteria don't have complexities, algorithms do**
2. **To study proof nets, learn about graph theory and algorithmics**
   - including *graphs embedded on surfaces* = *combinatorial maps*, especially (but not only) for cyclic / non-commutative MLL
3. Girard is not always right ;p