Around finite semantics for second-order multiplicative-additive linear logic

Nguyễn Lê Thành Dũng (a.k.a. Tito) — `nltd@nguyentito.eu`
Laboratoire d'informatique de Paris Nord (LIPN), Université Paris 13
GDRI-LL plenary meeting, Paris, December 3rd, 2018

Based on joint work with:

- Thomas Seiller (CNRS, LIPN), my PhD advisor
- Paolo Pistone (Univ. Tübingen)
- Lorenzo Tortora de Falco (Univ. Roma Tre)

Theorem

Second order Multiplicative-Additive Linear Logic (MALL2) *admits a finite denotational semantics.*

This talk: motivation and applications (intro);
multiple constructions of finite models of MALL2.

Some of this is very recent work in progress.
Correctness not guaranteed.

Why finite semantics?

The simply-typed λ-calculus admits finite denotational models, e.g. FinSet. Applications:

- *semantic evaluation* technique,
  often useful for studying complexity in STλ
- finite semantics for λY (i.e. STλ + fixpoints) lead to semantic approach to *higher-order model checking* (Aehlig, Salvati–Walukiewicz...)

Why finite semantics?

The simply-typed λ-calculus admits finite denotational models, e.g. FinSet. Applications:

- *semantic evaluation* technique,
  often useful for studying complexity in STλ
- finite semantics for λY (i.e. STλ + fixpoints) lead to semantic approach to *higher-order model checking* (Aehlig, Salvati–Walukiewicz...)

Propositional linear logic admits finite models, e.g. coherence spaces, Scott model. Yields refinements of the above, e.g.:

- Terui, RTA'12: semantic evaluation in the Scott model
  Slogan: “better semantics, faster computation”
- Grellois–Melliès: HOMC with LL semantics

Why finite semantics? An example

Church encoding of bitstrings:
$\mathsf{Str}[A] = (A \to A) \to (A \to A) \to (A \to A)$.

Theorem (Hillebrand & Kanellakis, LICS'96)

The languages decided by STλ*-terms of type* $\mathsf{Str}[A] \to \mathsf{Bool}$ *are exactly the* regular *languages.*

Proof idea: build a DFA whose states are $[\![\mathsf{Str}[A]]\!]$.

- an example of the semantic evaluation technique
- also, a correspondence Church encodings / finite automata; HOMC generalizes this to infinite trees

Finite semantics for polymorphism

Previous examples work for *monomorphic* type systems.

This seems impossible with impredicative polymorphism.
For instance in System F:

Proposition

Any non-trivial semantics must be injective on $\forall X.\, \mathsf{Str}[X]$.

Finite semantics for polymorphism

Previous examples work for *monomorphic* type systems.

This seems impossible with impredicative polymorphism.
For instance in System F:

Proposition

Any non-trivial semantics must be injective on $\forall X.\, \mathsf{Str}[X]$.

This example relies fundamentally on *non-linearity*.

Actually, there are finite models:

- with impredicative quantification;
- with exponentials;
- but not with both!

Applications for MALL2

Theorem

MALL2 *has a non-trivial finite semantics.*

Consequences on the expressive power of *recursive types*:

- μMALL cannot be embedded in MALL2, since infinite datatypes, e.g. initial algebras, cannot be encoded
- characterization of regular languages in second-order Elementary Linear Logic
  - this is Baillot's characterization of P in ELL, minus type fixpoints

Towards a syntactic model

Remark: in propositional MALL (MALL0), each formula has finitely many cut-free proofs.
$\rightarrow$ Hope for a *syntactic model* of MALL2, injective on MALL0.

Difficulty of 2nd order case: arbitrarily large $\exists$ witnesses.

Towards a syntactic model

Remark: in propositional MALL (MALL0), each formula has finitely many cut-free proofs.
$\rightarrow$ Hope for a *syntactic model* of MALL2, injective on MALL0.

Difficulty of 2nd order case: arbitrarily large $\exists$ witnesses.
Solution: *observational quotient* of the syntax.

- Choose observations which "cannot inspect witnesses"
- Intuition from programming languages: $\exists$ = abstract types, dual to $\forall$ = generic programs

Equivalence for propositional observations

Definition

Let A be a MALL2 formula and $\pi, \pi' : A$. Define $\pi \sim_A \pi'$ as: for any *propositional* MALL formula B, for any proof ρ of $A \vdash B$, $\mathbf{cut}(\pi, \rho)$ and $\mathbf{cut}(\pi', \rho)$ have the same normal form.

- $\sim$ is a congruence: the quotient is a model of MALL2
- A existential-free $\Rightarrow \sim_A$ trivial
- Example: the proofs of $\exists X.\, X$ cannot be distinguished
    - Impredicative encodings of units work, e.g. $\top \equiv \exists X.\, X$.

Theorem

For any MALL2 *formula A, there are* finitely many *classes for $\sim_A$.*

Proved with *proof nets* for MLL2, extended to MALL2 by a trick.

The observational quotient is non-effective

The observational quotient seems very concrete and simple.

However, given a type A, one cannot enumerate $A/\sim_A$,
even though it is finite. Indeed, one cannot check its emptiness:

Theorem (Lafont 1996)

MALL2 *is undecidable.*

More: adapting Lafont's proof gives (caveat: to be checked)

Proposition

Given a MALL2 *type A and $\pi, \pi' : A$, $\pi \sim_A \pi'$ is undecidable.*

(However, for a *fixed* A, $\sim_A$ is decidable.)

$\longrightarrow$ Search for *effective* finite models.
To overcome undecidability, enlarge the semantics.

Non-syntactic finite models of MALL2

We will look at two different proposals.

1. *Coherence spaces*, using Girard's interpretation of polymorphism with *stable functors*
    - Usual argument for superiority of coherence spaces over Scott domains: *small* and legible interpretations of types
    - Here they will even be finite!
2. A sort of Geometry of Interaction construction inspired by the finiteness proof of the observational quotient

Non-syntactic finite models of MALL2

We will look at two different proposals.

1. *Coherence spaces*, using Girard's interpretation of polymorphism with *stable functors*
    - Usual argument for superiority of coherence spaces over Scott domains: *small* and legible interpretations of types
    - Here they will even be finite!
2. A sort of Geometry of Interaction construction inspired by the finiteness proof of the observational quotient

All of this is very recent.
I started looking at coherence spaces last week and stumbled against technical problems. (Only $\simeq$ 12 hours ago did I finally convince myself composition works!)

Coherence spaces: variable types

We consider the category CohI:

- objects: coherence spaces (graphs X over vertex set $|X|$)
- morphisms $X \hookrightarrow Y$: *embeddings*, i.e. a graph isomorphism between X and an induced subgraph of Y

An open type is represented as a functor $F : \mathsf{CohI}^n \to \mathsf{CohI}$.
For instance $(- \multimap -)$ is indeed *covariant*:
if $X \subseteq X'$, $Y \subseteq Y'$, then $X \multimap Y \subseteq X' \multimap Y'$.

We require F to be *stable*: preserving filtered colimits and pullbacks (think directed unions and finite intersections).

Coherence spaces: variable cliques

If F is a stable functor, what are "terms of type F"?

A *variable clique* is a family $c_X \in \text{Cliques}(F(X))$, such that for all embeddings $f : X \hookrightarrow Y$, $c_X = f^{-1}(c_Y)$.

For an inclusion $X \subseteq Y$, this can be written $c_X = c_Y \cap |F(X)|$.

One may build the category interpreting types with 1 variable:

- objects: stable functors $F : \mathsf{CohI} \to \mathsf{CohI}$
- morphisms: variable cliques of $F \multimap G$

Coherence spaces: the composition problem

Variable cliques $c_X \in \mathrm{Cliques}(F(X) \multimap G(X))$ correspond to linear functions $f_X : F(X) \multimap G(X)$ making this diagram commute for all $\iota : X \hookrightarrow Y$:

$$\begin{tikzcd} F(Y) \arrow[r, "f_Y"] & G(Y) \arrow[d, "G(\iota)^-"] \\ F(X) \arrow[u, "F(\iota)^+"] \arrow[r, "f_X"] & G(X) \end{tikzcd}$$

(In other words $f_X(d) = f_Y(d) \cap |G(X)|$ for $d \subseteq X \subseteq Y$.)

Unlike naturality diagrams – which do *not* hold in general for variable cliques – a priori it is unclear that these can be pasted horizontally...

So composition, although it works (?), is a subtle consequence of *stability* (so it also works for the model of System F).

Stable functors of finite degree (1)

If $x \in |F(X_1, \ldots, X_n)|$, then there are *finite* $X_1^0 \subseteq X_1, \ldots, X_n^0 \subseteq X_n$ such that $x \in |F(X_1^0, \ldots, X_n^0)|$.
This is because F is stable (Girard's *normal form theorem*).

But the X_i may be *unbounded*, e.g. $F(X) = !X$.

We restrict to functors *of finite degree*: if $x \in |F(X_1, \ldots, X_n)|$, there are $X_1^0, \ldots, X_n^0$ of cardinality $\leq d$ such that ...
Call $\deg(F)$ the smallest such d.

We say F is *finite* if it maps finite spaces to finite spaces and is of finite degree.

Stable functors of finite degree (2)

By restricting to finite functors, we have excluded $?/!$ but we still interpret MALL2.

$$\deg(F^{\perp}) = \deg(F) \quad \deg(\forall Y.\, F(X_1, \ldots, X_n, Y)) \leq \deg(F)$$

$$\deg(F \otimes G) \leq \deg(F) + \deg(G) \quad \deg(F \oplus G) \leq \max(\deg(F), \deg(G))$$

And if F is finite, then $\forall Y.\, F(\ldots, Y)$ also is (the X_i^0's are closely related to the interpretation of $\forall$, this is where degree and cardinality interact).

Theorem

Coherence spaces provide a finite semantics for MALL2.

Effectivity: I haven't had the time to check yet...

A Geometry of Interaction-like model

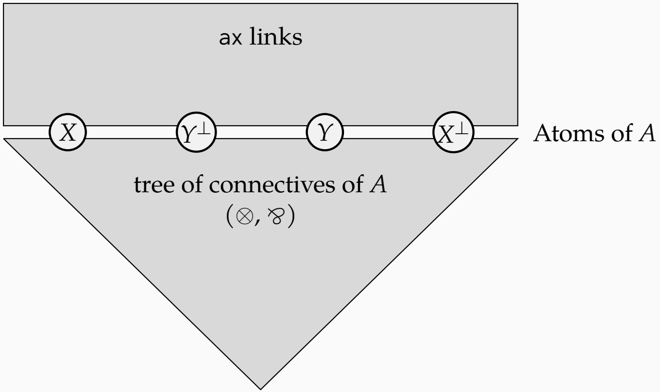
Before presenting the next model, to motivate the intuitions, I need to come back to the proof of:

Theorem

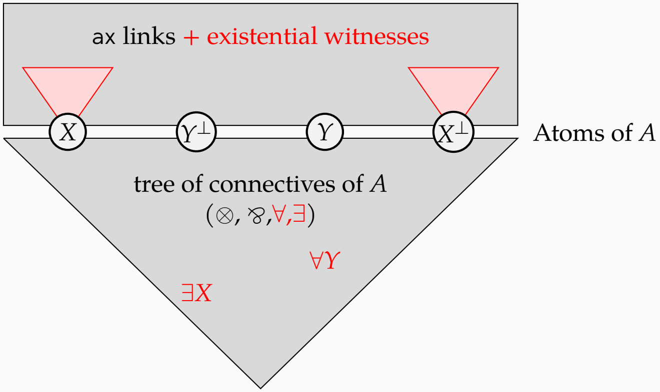
For any MALL2 *formula A, there are* finitely many *classes for* $\sim_A$.

From now on, familiarity with proof nets and GoI is assumed.

What a propositional MLL proof net looks like

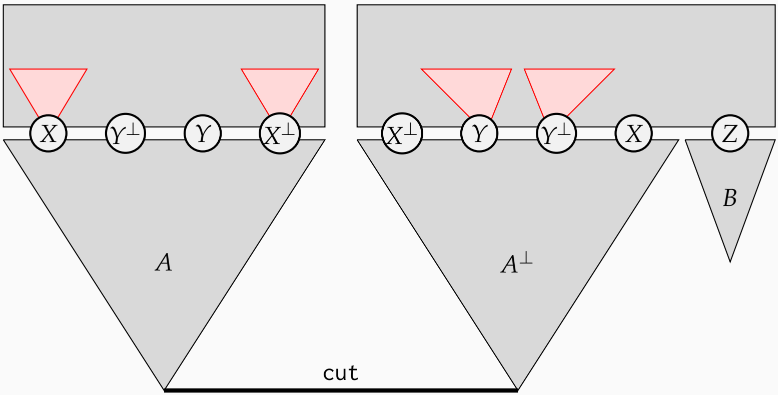


What a MLL2 proof net looks like



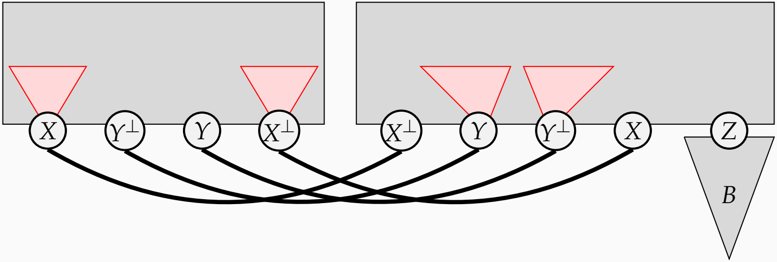
Finiteness theorem (1): proof/observation interaction

A: MLL2 formula; B: propositional MLL formula

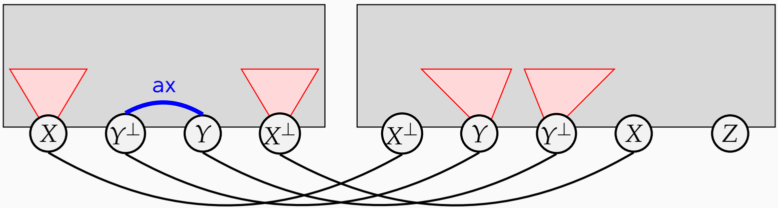


Finiteness theorem (1): proof/observation interaction

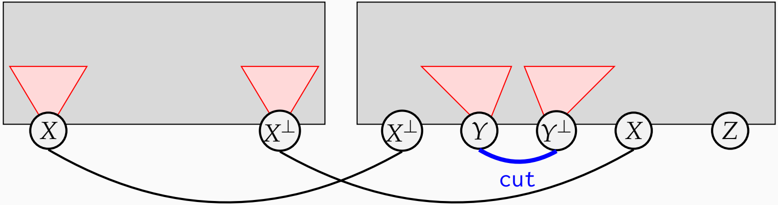
A: MLL2 formula; B: propositional MLL formula



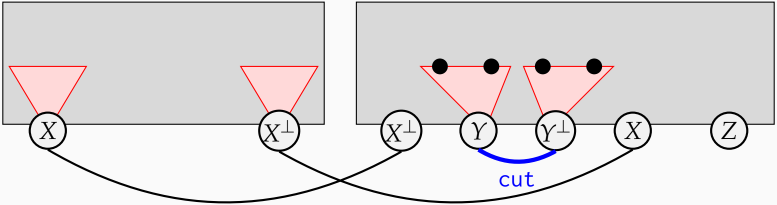
Finiteness theorem (2): eliminating two cuts



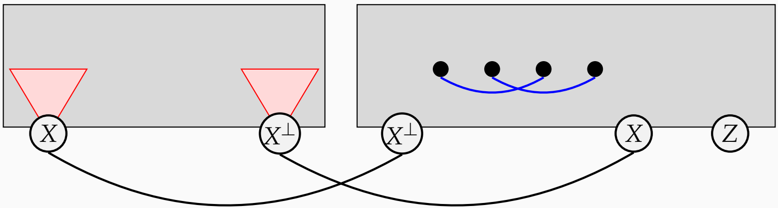
Finiteness theorem (2): eliminating two cuts



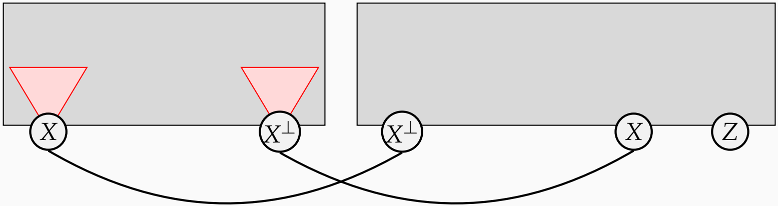
Finiteness theorem (2): eliminating two cuts



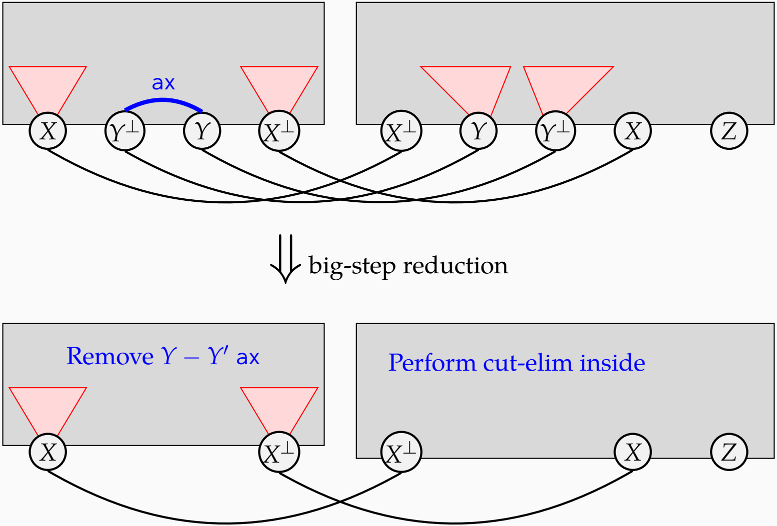
Finiteness theorem (2): eliminating two cuts



Finiteness theorem (2): eliminating two cuts

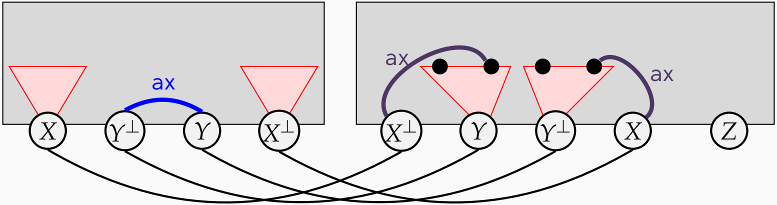


Finiteness theorem (3): big-step reduction



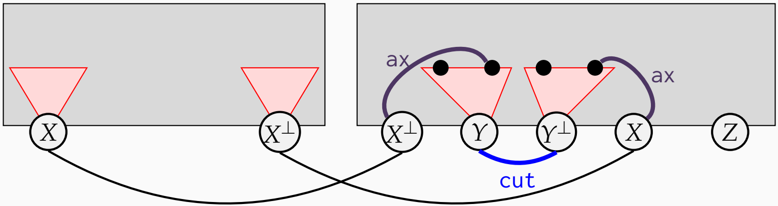
Finiteness theorem (4): normalization

New redexes may appear during reduction.



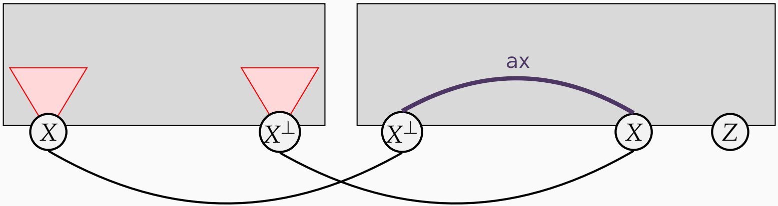
Finiteness theorem (4): normalization

New redexes may appear during reduction.



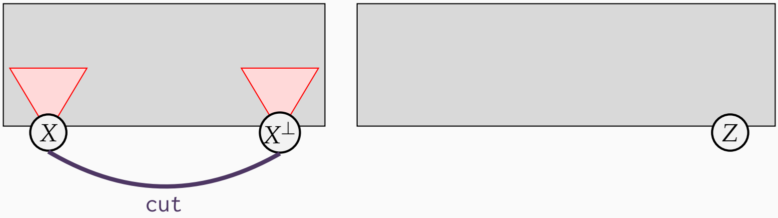
Finiteness theorem (4): normalization

New redexes may appear during reduction.



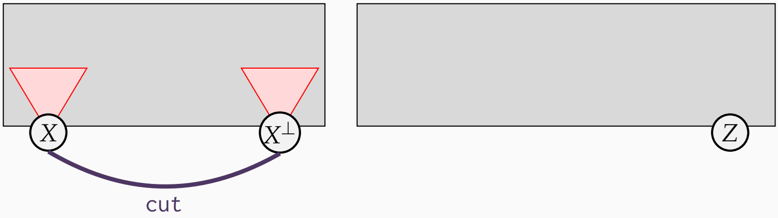
Finiteness theorem (4): normalization

New redexes may appear during reduction.



Finiteness theorem (4): normalization

New redexes may appear during reduction.



Lemma (Progress)

As long as there remains a `cut`*-link, there is a redex for* $\Rightarrow$.

This is the tricky part of the proof.

MLL2 normalization as a dialogue

One can see this big-step reduction as a *dialogue* between proof and observation, through the interface of the cut formula.

Bound on the number of rounds of the dialogue
$\rightsquigarrow$ bound on number of classes in observational quotient.

We would like to represent proofs as “strategies” for this dialogue, in such a way that strategies compose.

MLL2 normalization as a fixed point

To a proof, we may associate a function sending a (partial) pairing of $\exists$ variables to a pairing of $\forall$ variables.
The "dialogue" then computes a *minimum common fixed point*.
$\longrightarrow$ as in the GoI's *feedback equation*.

A general setting for normalization as feedback: the Int construction on traced monoidal categories.

$\mathsf{Int}(\mathcal{C})$ is automatically compact closed: we get propositional MLL for free.

The GoI-like model (1)

We now work in the category $\mathsf{Int}(\mathsf{CohS})$:

- objects: pairs (A, E) of coherence spaces
- morphisms $(A, E) \to (A', E')$: stable fn $A \& E' \to E \& A'$
- composition $(A, E) \to (A', E') \to (A'', E'')$: take

$$A \& E' \to E \& A' \qquad A' \& E'' \to E' \& A''$$

  feed right E' to left E', left A' to right A',
  and compute minimum fixpoint.
- monoidal product: $(A, E) * (A', E') = (A \& A', E \& E')$
- duality: $(A, E)^{\perp} = (E, A)$

Intuition: ask E, answer A. Will be used with
$E = \{\exists \text{ variable pairings}\}$, $A = \{\forall \text{ pairings}\}$.

The GoI-like model (2)

A variable type should be a functor
$F : \mathsf{Int}(\mathsf{CohS})^n \times (\mathsf{Int}(\mathsf{CohS})^{\mathrm{op}})^n \to \mathsf{Int}(\mathsf{CohS})$.

And an element of type F should be a "sufficiently uniform" family of morphisms $1 \to F(X, X)$.

In the case $F(X, X) = K * X^m * (X^\perp)^n$, and $X = (A, E)$,
that is a stable function $E_K \& E^m \& A^n \to A_K \& A^m \& E^n$,
which should just redirect:

- the A on the left (from $X^\perp$) to those on the right (from X)
- the E on the left (from X) to those on the right (from $X^\perp$)

corresponding respectively to arcs $X^\perp \longrightarrow X$ and $X \longrightarrow X^\perp$.
So it would be a function from E_K to such a set of directed arcs, generalizing sets of axiom links. ($\simeq$ Seiller's interaction graphs)

The GoI-like model (3)

Write $L(m,n)$ for the set of directed arcs either from $\{1,\ldots,m\}$ to $\{1,\ldots,n\}$ or from $\{1,\ldots,n\}$ to $\{1,\ldots,m\}$.
With the right coherence relation, the cliques of $L(m,n)$ are codeterministic bipartite directed graphs.

Informal conjecture:

$$\forall X.\, K * X^m * (X^\perp)^n = K * (L(m,n), 1)$$

$$\exists X.\, K * X^m * (X^\perp)^n = K * (1, L(m,n))$$

$\forall$ adds a part to the answers.
$\exists$ adds a part to the questions.

Note: $\forall X$ and $\exists X$ commute with $(K * -)$ if K contains no X.
As expected, since $* = \otimes = \parr$.

The GoI-like model (3)

How to reflect composition of "sufficiently uniform" families
$K * X^m * (X^\perp)^n \to K' * X^{m'} * (X^\perp)^{n'} \to K'' * X^{m''} * (X^\perp)^{n''}$
as an operation on the $\forall$? Take

$A \& E' \to E \& A' \& L(n{+}m', m{+}n') \quad A' \& E'' \to E' \& A'' \& L(n'{+}m'', m'{+}n'')$

Feedback matching input/output, we get:

$$A \& E'' \to E \& A'' \& L(n + m', m + n') \& L(n' + m'', m' + n'')$$

Need morphism
$L(n + m', m + n') \& L(n' + m'', m' + n'') \to L(n + m'', m + n'')$:
should correspond to execution by alternating paths (à la GoI)!
First resolve dialogue of $\exists/\forall$, then propositional normalization.

$\longrightarrow$ particle-style GoI = wave-style GoI + uniformity

Inversion of polarities

Types of the form $(A, \top)$ are morally *positive*, they can be duplicated/erased (because $\&$ = cartesian product of CohS).

Positivation: $!(A, E) = (E \to A, \top)$.
Seems to interpret propositional MELL.

$\forall$ seems to be positive, since it acts on the left of the pair!
In particular $\forall X. F$, where the only variable of F is X, is interpreted as positive.
As opposed to usual polarities: $\forall$ inversible, therefore negative.

Morally, $\forall$ "plays" a set of arcs, it is analogous to a $\oplus$;
whereas $\exists$ "reacts", analogously to $\&$.

(Extension to additives will probably need to handle polarity.)

$$A \otimes B \equiv \forall X.\, (A \multimap B \multimap X) \multimap X$$

$$A \oplus B \equiv \forall X.\, !(A \multimap X) \multimap !(B \multimap X) \multimap X$$

"The second-order translations do invert polarities. [...] *This inversion of polarities is due to deep reasons – so deep that they remain completely obscure."*

— Girard, *The Blind Spot*, §12.B.2