A gentle introduction to Girard’s Transcendental Syntax

LIPN – Université Sorbonne Paris Nord
Boris Eng Thomas Seiller
A bit of context

*Geometry of Interaction (GoI)*

A lot of definitions...But in our case:
A bit of context

Geometry of Interaction (GoI)

A lot of definitions...But in our case:

Girard’s original Geometry of Interaction [GoI I, 1989].
A bit of context

Geometry of Interaction (GoI)

A lot of definitions...But in our case :

Girard’s original Geometry of Interaction [GoI I, 1989].

Goal: study the dynamics of linear logic from computation (operator algebras).
A bit of context

*Geometry of Interaction (GoI)*

A lot of definitions...But in our case:

Girard’s original Geometry of Interaction [GoI I, 1989].

Goal: study the dynamics of linear logic from computation (operator algebras).

Transcendental Syntax [GoI VI, 2013]: the successor.
A bit of context

*Geometry of Interaction (GoI)*

A lot of definitions...But in our case:

Girard’s original Geometry of Interaction [GoI I, 1989].

Goal: study the *dynamics* of linear logic from *computation* (operator algebras).

Transcendental Syntax [GoI VI, 2013]: the successor.

Goal: linear logic (proof-nets) as *emerging* from computation *without semantics*. 
A bit of context

*Geometry of Interaction (GoI)*

A lot of definitions...But in our case:

Girard’s original Geometry of Interaction [GoI I, 1989].

 Goal: study the dynamics of linear logic from computation (operator algebras).

Transcendental Syntax [GoI VI, 2013]: the successor.

 Goal: linear logic (proof-nets) as emerging from computation without semantics.

 Computational bricks: "stellar resolution" (not the only possibility).
A bit of context

Geometry of Interaction (GoI)

A lot of definitions...But in our case:

Girard’s original Geometry of Interaction [GoI I, 1989].

Goal: study the dynamics of linear logic from computation (operator algebras).

Transcendental Syntax [GoI VI, 2013] : the successor.

Goal: linear logic (proof-nets) as emerging from computation without semantics.

Computational bricks: "stellar resolution" (not the only possibility).

Logical correctness: by symmetric computational testing.
Stellar Resolution

Between tilings and logic programming

"Flexible" tiles [stars] with (un)polarised terms [rays]. Group as [constellations].

\[
g(x) \rightarrow \phi_1 \rightarrow +a(x) \rightarrow -b(x) \\
\phi_2 \rightarrow -a(f(y)) \rightarrow +c(y)
\]
"Flexible" tiles [stars] with (un)polarised terms [rays]. Group as [constellations].

\[
g(x) \quad +a(x) \quad -a(f(y)) \quad +c(y)
\]

\[
\phi_1 \quad \phi_2
\]

\[
-b(x)
\]
Stellar Resolution

Between tilings and logic programming

"Flexible" tiles [stars] with (un)polarised terms [rays]. Group as [constellations].

\[ g(x) \cdot \phi_1 \quad +a(x) \quad -a(f(y)) \quad +c(y) \quad -b(x) \cdot \phi_2 \]

Evaluation: link-contraction by Robinson’s Resolution rule.
Stellar Resolution
Between tilings and logic programming

"Flexible" tiles [stars] with (un)polarised terms [rays]. Group as [constellations].

\[
g(x) \cdot \phi_1 \quad +a(x) \quad \cdot \quad -a(f(y)) \quad +c(y) \quad \phi_2 \cdot \neg b(x) \cdot
\]

Evaluation: link-contraction by Robinson’s Resolution rule.

Execution: construct all possible connected & maximal tilings then evaluate them.
Encoding proof-structures

Computational content of proofs

ax ax ax

\(\exists\) \(\otimes\)

cut
Encoding proof-structures

Computational content of proofs

1 \rightarrow 7 \rightarrow 8 \leftarrow \text{cut}
Encoding proof-structures

Computational content of proofs

\[
\begin{array}{c}
\text{ax} \\
1 \\
\Rightarrow \\
2 \\
\text{ax} \\
3 \\
\Rightarrow \\
4 \\
\text{ax} \\
5 \\
\Rightarrow \\
6 \\
\otimes \\
7 \\
\Leftarrow \\
8 \\
\text{cut} \\
\end{array}
\]

\[
\begin{array}{c}
\Rightarrow \\
\downarrow \\
\text{cut} \\
\Rightarrow \\
\end{array}
\]

\[
\begin{array}{c}
\Rightarrow \\
\downarrow \\
\text{Ex} \\
\Rightarrow \\
\end{array}
\]

\[
\begin{array}{c}
\Rightarrow \\
\downarrow \\
\text{Ex} \\
\Rightarrow \\
\end{array}
\]

\[
\begin{array}{c}
\text{cut} \\
\Rightarrow \\
\end{array}
\]

\[
\begin{array}{c}
\Rightarrow \\
\downarrow \\
\text{Ex} \\
\Rightarrow \\
\end{array}
\]

\[
\begin{array}{c}
\Rightarrow \\
\downarrow \\
\text{Ex} \\
\Rightarrow \\
\end{array}
\]
Encoding proof-structures

Computational content of proofs

\[ \begin{aligned}
\otimes & \quad \text{cut} \\
1 & \quad \otimes \\
\otimes & \quad \text{cut} \\
2 & \quad \otimes \\
\otimes & \quad \text{cut} \\
3 & \quad \otimes \\
\otimes & \quad \text{cut} \\
4 & \quad \otimes \\
\otimes & \quad \text{cut} \\
5 & \quad \otimes \\
\otimes & \quad \text{cut} \\
6 & \quad \otimes \\
\otimes & \quad \text{cut} \\
7 & \quad \otimes \\
\otimes & \quad \text{cut} \\
8 & \quad \otimes \\
\end{aligned} \]
Encoding proof-structures

Logical content of proofs

Danos-Regnier correctness: is axioms+test a tree for any test?

Stellar logical correctness: does \( \text{Ex}(\text{uniEF26} \, \text{ax} \, S \sqcup \text{uniEF26} \, \text{test} \, S, \phi) \) satisfy some property \( P \)?

\( \Rightarrow_{\text{MLL}}: |\text{Ex}(\text{uniEF26} \, \text{ax} \, S \sqcup \text{uniEF26} \, \text{test} \, S, \phi)| = \text{one.pnum}. \)

\( \Rightarrow_{\text{MLL+MIX}}: \text{Ex}(\text{uniEF26} \, \text{ax} \, S \sqcup \text{uniEF26} \, \text{test} \, S, \phi) \) terminates.

Orthogonality: \( \text{Ex}(\text{uniEF26} \, \text{one.pnum} \sqcup \text{uniEF26} \, \text{two.pnum}) \) satisfies \( P \) \iff \( \text{uniEF26} \, \text{one.pnum} \bot \text{uniEF26} \, \text{two.pnum} \).

\( 1 \rightarrow ax \rightarrow 2 \rightarrow ax \rightarrow 3 \rightarrow ax \rightarrow 4 \rightarrow ax \rightarrow 5 \rightarrow ax \rightarrow 6 \)

\( \Rightarrow_L \)

\( \Rightarrow_R \)

\( \vdash \)

\( \text{cut} \)
Encoding proof-structures

Logical content of proofs

Danos-Regnier correctness: is axioms+test a tree for any test?
Encoding proof-structures

Logical content of proofs

Danos-Regnier correctness: is axioms+test a tree for any test?
Stellar logical correctness: does $\text{Ex}(\Phi_{\text{ax}} \cup \Phi_{\text{test}})$ satisfy some property $P$?
Encoding proof-structures

Logical content of proofs

Danos-Regnier correctness: is axioms+test a tree for any test?

Stellar logical correctness: does $\text{Ex}(\Phi^{\text{ax}}_{\mathcal{I}} \cup \Phi^{\text{test}}_{\mathcal{I},\varphi})$ satisfy some property $P$?

$\downarrow$ MLL: $|\text{Ex}(\Phi^{\text{ax}}_{\mathcal{I}} \cup \Phi^{\text{test}}_{\mathcal{I},\varphi})| = 1$. 

$\uparrow$ MLL+MIX: $\text{Ex}(\Phi^{\text{ax}}_{\mathcal{I}} \cup \Phi^{\text{test}}_{\mathcal{I},\varphi})$ terminates.
Encoding proof-structures

Logical content of proofs

Danos-Regnier correctness: is axioms+test a tree for any test?
Stellar logical correctness: does \( \text{Ex}(\Phi^\text{ax} \cup \Phi^\text{test}) \) satisfy some property \( P \)?

\( \Downarrow \) MLL: \( |\text{Ex}(\Phi^\text{ax} \cup \Phi^\text{test})| = 1. \)

\( \Downarrow \) MLL+MIX: \( \text{Ex}(\Phi^\text{ax} \Phi^\text{test}) \) terminates.
Encoding proof-structures

*Logical content of proofs*

Danos-Regnier correctness: is axioms+test a tree for any test?

Stellar logical correctness: does $\text{Ex}(\Phi^\text{ax}_\mathcal{I} \cup \Phi^\text{test}_\mathcal{I},\phi)$ satisfy some property $P$?

$\Downarrow$ MLL: $|\text{Ex}(\Phi^\text{ax}_\mathcal{I} \cup \Phi^\text{test}_\mathcal{I},\phi)| = 1$.

$\Downarrow$ MLL+MIX: $\text{Ex}(\Phi^\text{ax}_\mathcal{I} \cup \Phi^\text{test}_\mathcal{I},\phi)$ terminates.

Orthogonality. $\text{Ex}(\Phi_1 \cup \Phi_2)$ satisfies $P \iff \Phi_1 \perp \Phi_2$. 
Two notions of type

Unified in the same framework

Types as labels (type theory). $A, B ::= X_i \mid X_i^\perp \mid A \otimes B \mid A \Rightarrow B$. 
Two notions of type

Unified in the same framework

Types as labels (type theory). $A, B ::= X_i \mid X_i^\perp \mid A \otimes B \mid A \oslash B$.

$\Downarrow A \mapsto \text{Tests}(A)$ finite \quad $\Phi$ logically correct $\iff \Phi \perp \text{Tests}(A)$. 
Two notions of type

*Unified in the same framework*

Types as labels (type theory). $A, B ::= X_i \mid X_i^\bot \mid A \otimes B \mid A \bowtie B$.

$\Downarrow$ $A \leftrightarrow \text{Tests}(A)$ finite $\quad \Phi$ logically correct $\iff \Phi \perp \text{Tests}(A)$.

Types as behaviour classes (realisability).
Two notions of type

Unified in the same framework

Types as labels (type theory). \( A, B ::= X_i \mid X_i^\perp \mid A \otimes B \mid A \Rightarrow B. \)
\[ \Downarrow \quad A \leftrightarrow \text{Tests}(A) \text{ finite} \quad \Phi \text{ logically correct } \iff \Phi \perp \text{Tests}(A). \]

Types as behaviour classes (realisability).

- Pre-type : set of constellation \( A; \)
Two notions of type

*Unified in the same framework*

**Types as labels (type theory).** $A, B ::= X_i \mid X_i^\perp \mid A \otimes B \mid A \nabla B$.

$\Downarrow$ $A \leftrightarrow \text{Tests}(A)$ finite $\quad \Phi$ logically correct $\iff \Phi \perp \text{Tests}(A)$.

**Types as behaviour classes (realisability).**

- Pre-type : set of constellation $A$;
- Orthogonal : $A^\perp$ (dual constellations);
Two notions of type

Unified in the same framework

Types as labels (type theory). $A, B ::= X_i \mid X_i^\perp \mid A \otimes B \mid A \Rightarrow B$.

$\Downarrow A \mapsto \text{Tests}(A)$ finite $\Phi$ logically correct $\iff \Phi \perp \text{Tests}(A)$.

Types as behaviour classes (realisability).

- Pre-type: set of constellation $A$;
- Orthogonal: $A^\perp$ (dual constellations); Conduct: $A = A^\perp\perp$;
Two notions of type

Unified in the same framework

Types as labels (type theory). \( A, B ::= X_i \mid X_i^{\perp} \mid A \otimes B \mid A \triangleright B. \)
\( \Downarrow \) \( A \leftrightarrow \text{Tests}(A) \) finite \( \Phi \) logically correct \( \iff \Phi \perp \text{Tests}(A) \).

Types as behaviour classes (realisability).

- Pre-type : set of constellation \( A \);
- Orthogonal : \( A^{\perp} \) (dual constellations) ; Conduct : \( A = A^{\perp \perp} \);
- Tensor : \( A \otimes B = \{ \Phi_A \cup \Phi_B, \Phi_A \in A, \Phi_B \in B \}^{\perp \perp} \).
Two notions of type

*Unified in the same framework*

**Types as labels (type theory).** \( A, B ::= \mathcal{X}_i | \mathcal{X}_i \perp | A \otimes B | A \bowtie B. \)

\( \Downarrow A \mapsto \text{Tests}(A) \text{ finite} \quad \Phi \text{ logically correct } \iff \Phi \perp \text{Tests}(A). \)

**Types as behaviour classes (realisability).**

- Pre-type : set of constellation \( A \);
- Orthogonal : \( A \perp \) (dual constellations); Conduct : \( A = A \perp \perp \);
- Tensor : \( A \otimes B = \{ \Phi_A \cup \Phi_B, \Phi_A \in A, \Phi_B \in B \} \perp \perp. \)

Infinitely many (sub)types + \( \Phi \in A \) usually undecidable vs \( \Phi : A \) usually decidable.
Two notions of type

*Unified in the same framework*

Types as labels (type theory). $A, B ::= X_i \mid X_i^\perp \mid A \otimes B \mid A \nRightarrow B$.

$\Downarrow$ $A \mapsto \text{Tests}(A)$ finite $\quad \Phi$ logically correct $\iff \Phi \perp \text{Tests}(A)$.

Types as behaviour classes (realisability).

- Pre-type: set of constellation $A$;
- Orthogonal: $A^\perp$ (dual constellations); Conduct: $A = A^\perp\perp$;
- Tensor: $A \otimes B = \{ \Phi_A \cup \Phi_B, \Phi_A \in A, \Phi_B \in B \}^{\perp\perp}$.

Infinitely many (sub)types + $\Phi \in A$ usually undecidable *vs* $\Phi : A$ usually decidable.

Related by adequacy: $\text{Tests}(A)^\perp \subseteq A$. 
Technical development

Current works / In progress.

• formal definition of stellar resolution & properties;
• encoding of several models (automata, circuits, tiling models, ...);
• model of MLL(+MIX) and IMELL (Intuitionistic exponentials);

Future works.

• New point of view for first/second order logic + additives + neutrals;
• Implicit computational complexity analysis.

Thank you for listening.
Technical development

Current works / In progress.

- formal definition of stellar resolution & properties;

Future works.

- New point of view for first/second order logic + additives + neutrals;
- Implicit computational complexity analysis.

Thank you for listening.
Technical development

Current works / In progress.

- formal definition of stellar resolution & properties;
- encoding of several models (automata, circuits, tiling models, ...);
Technical development

Current works / In progress.

- formal definition of stellar resolution & properties;
- encoding of several models (automata, circuits, tiling models, ...);
- model of $\text{MLL}(+\text{MIX})$ and $\text{IMELL}$ (Intuitionistic exponentials);
Technical development

Current works / In progress.

- formal definition of stellar resolution & properties;
- encoding of several models (automata, circuits, tiling models, ...);
- model of MLL(+MIX) and IMELL (Intuitionistic exponentials);

Future works.

Thank you for listening.
Technical development

Current works / In progress.

- formal definition of stellar resolution & properties;
- encoding of several models (automata, circuits, tiling models, ...);
- model of MLL(+MIX) and IMELL (Intuitionistic exponentials);

Future works.

- New point of view for first/second order logic + additives + neutrals;
Technical development

Current works / In progress.

• formal definition of stellar resolution & properties;
• encoding of several models (automata, circuits, tiling models, ...);
• model of MLL(+MIX) and IMELL (Intuitionistic exponentials);

Future works.

• New point of view for first/second order logic + additives + neutrals;
• Implicit computational complexity analysis.
Technical development

Current works / In progress.

- formal definition of stellar resolution & properties;
- encoding of several models (automata, circuits, tiling models, ...);
- model of MLL(+MIX) and IMELL (Intuitionistic exponentials);

Future works.

- New point of view for first/second order logic + additives + neutrals;
- Implicit computational complexity analysis.

Thank you for listening.