Immanence in logic

A journey towards transcendental syntax

Friday 20th January, 2023

Introduction

This is a broad overview on foundations of logic, mainly based on the works of Jean-Yves Girard but sprinkled with some personal interpretations and general knowledge. We'll delve into his four levels grounding meaning, eventually reaching his transcendental syntax.

Transcendental syntax is a shift of goals in foundations of logic : instead of establishing "the" good axiomatic system, it aims at establishing conditions of possibility for the logical language.

This is not a quest for purity : we should get new, concrete benefits to understand what is really gained with our shifts in point of view.

The basements

-1. Alethic level: truth & models

-2. Functional level: proofs & categories

-3. Interactive level: tests & games

-4. Deontic level: normativity

-1. Alethic level: truth & models

The constat

Logic is arguably about pure, deductive reasoning. So, when seeing that we have multiple logics given by axiomatic systems, one can ask two naive questions :

- How are those systems justified ?
- Why is deductive activity restrained to such closed systems ?

The constat

- How are those systems justified ?
- Why is deductive activity restrained to such closed systems ?

Model-theoretic justification : the meaning of logical rules is ascribed to the preservation of truth in some mathematical structure ($\vdash \Leftrightarrow \models$). However, that results in incompatible logics and leaking justifications.

Meta-Tarski

Justifications are leaking to a meta-language, needing a meta^2 and so on. Typically :

- The truth of $A \wedge B$ is defined to be the truth of A *and* the truth of B (Tarski pleonasm), ...
- The distinction between entailment and implication is not semantical (not its *meta*, see Achilles and the Tortoise).

Semantics of a language is then given by the syntax of its meta-language, which is basically himself - only separated for safety measures. But translations are more useful when the source and target languages are further apart.

Epicycles of completeness

At this level, completeness is often seen as "cannot prove more".
But w.r.t. which models ?

- Classical logic $\Rightarrow$ Sets, Stone spaces, ...
- Intuitionistic logic $\Rightarrow$ Kripke frames, topological spaces, ...
- Linear logic $\Rightarrow$ Coherence spaces, operator algebras, ...
- ...

One can then manipulate e.g. sets in topology and vice-versa :
don't mistake encodings for reality !

Epicycles of completeness

At this level, completeness is often seen as "cannot prove more". But w.r.t. which models ?

We know how to get trivial completeness with syntactic models, so it brings no new information : see e.g. the Lindenbaum–Tarski algebra for classical model theory, yielding boolean algebra for classical logic.

We fall back on mathematical arguments to justify the pertinence of various interpretations, lacking the kind of absolute criterion logic is fond of.

Epicycles of completeness

At this level, completeness is often seen as "cannot prove more". But w.r.t. which models ?

We cannot avoid speaking of unwanted models in 1st-order logic :

- Non-standard numbers in PA
- Non well-founded sets in ZFC
- Countable interpretation of reals in analysis
- ...

Typically, those "non-standard numbers" (not numbers) are unavoidable because we'll never see coming, exactly how we'll never see a proof of $Cons(PA)$ coming in $PA + \neg Cons(PA)$.

In many completeness theorems, models are infinite branches in any proof of A : they function as counter-arguments against A before being studied as mathematical structures on their own rights (the spice of model theory).

Hilbert program

An ounce of immanence :

The normative features are not searched in external structures anymore but directly applicable to syntactical artifacts. This ambitious point of view is justified by its universality, despite the diversity of syntaxes for deductive systems (e.g. natural deduction VS sequent calculus).

Hilbert program

An ounce of immanence :

We seek for the absence of both proofs of A and $\neg$A for any A. Incompleteness tells us that it's both :

- Out of reach : sanctions logicism & finitism, dogma that deductive reasoning is analytic (in Kant meaning, i.e. self-sufficient).
- Not enough : theories can make mistakes such as $T + \neg Cons(T)$ ("not seen, not caught").

Moreover in practice, consistency is justified from an exhibited model, not the converse. It's the end of naive foundations - those hoping to internalize everything.

-2. Functional level: proofs & categories

The Brouwer-Heyting-Kolmogorov interpretation

Truth remains an external notion, so we can aim to understand logic through what it actually produces : formulas and proofs.

What is a proof, outside of any particular deductive system ?

The Brouwer-Heyting-Kolmogorov interpretation

What is a proof, outside of any particular deductive system ?

- We know how to recognize proofs of atomic formulas.
- A proof of $A \wedge B$ is a pair (a, b) with a proof of A and a proof of B.
- A proof of $A \vee B$ is a pair $\langle i; p \rangle$ where either $i = 0$ and p is a proof of A, or $i = 1$ and p is a proof of B.
- There is no proof of $\bot$.
- **A proof of $A \Rightarrow B$ is a function which, given any proof of A, produces a proof of B.**

This is BHK, the standard interpretation of intuitionistic logic. The implication is the most important and is defined through its use/behavior, not some rule to follow : it replaces the why by the how.

The Brouwer-Heyting-Kolmogorov interpretation

Quantifiers $\exists, \forall$ are handled like generalized $\vee$ resp. $\Rightarrow$, with their first component (resp. input) being in the domain of discourse :

In PA, a proof of $\forall x \exists y . A \vee B$ is a function which, for any given number x, outputs a number y and decides whether A or B holds for x and y by exhibiting a proof of either one.

Intuitionistic logic is a constructive logic, i.e. different proofs of the same formula matter (e.g. above, not same number output nor choices between A or B).

Fresh air compared to classical logic : formulas are demands for objects that can be used, we already know that deductive activity is not limited to yes/no questions ! E.g. "how much is $2 + 2$?" VS "does $2 + 2$ equal 4 ?". Only jerks answer "yes" to "do you have the hour ?".

The Brouwer-Heyting-Kolmogorov interpretation

$\neg A := A \Rightarrow \bot$. Extensionally, only one function to the empty set of proofs of $\bot$: no (denotational) constructive content for negative propositions.

$$\neg\neg\neg A \Rightarrow \neg A \equiv ((A \Rightarrow \bot) \Rightarrow \bot) \Rightarrow A \Rightarrow \bot$$

$$nDNE = \lambda d.\lambda a.d(\lambda n.n(a))$$

$$\neg\neg(A \vee \neg A) \equiv ((A \vee (A \Rightarrow \bot)) \Rightarrow \bot) \Rightarrow \bot$$

$$nnLEM = \lambda d.d(\lambda n.n(\langle 1; \lambda a.n(\langle 0; a\rangle)\rangle))$$

We can embed classical logic in IL by double negation translation : against the completeness dogma "proving more requires giving up models", IL is structurally stronger than CL despite having "more" models.

The Hauptsatz

At the level of truth, semantics discards any content from proofs. This very lossy approach is justified by the disparate appearance of syntactic proofs. However, a deep invariant lies in proof systems.

The cut rule in sequent calculus (dual to the identity rule) is the only rule which allows us to use proofs. It corresponds to elimination rules in natural deduction.

$$\frac{\Gamma \vdash \Delta, A \qquad A, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ cut$$

$$\frac{A \wedge B}{A}\ \wedge_{e1}$$

$$\frac{A \qquad A \Rightarrow B}{B}\ \Rightarrow_{e}$$

Cuts allow proofs to be modular, to plug lemmas together : almost every proof uses it.

The Hauptsatz

Gentzen's *Hauptsatz* (1934) : First-order IL / CL proofs can be transformed into proofs of the same formula without cuts. It's surprising to be able discard e.g. Modus Ponens !

The proof is tedious. Consistency follows immediately from it, but we gain much more : we can study the structure of proofs and their invariants along cut elimination.

IL cut-elimination is confluent : single explicit content of a given proof, irrelevant of the cut-elimination algorithm used. We have access to its denotation !

CL is algorithmically inconsistent : reducing a proof with weakening or contraction on both sides on sequents leads to any cut-free proof, i.e. does not give any information.

The Hauptsatz

Given the proofs p of $\vdash A \wedge B$ and q of $A \vdash C$, how to prove $\vdash C$?

$$\dfrac{\begin{array}{c} p \\ \vdash A \wedge B \end{array} \qquad \dfrac{\dfrac{\begin{array}{c} q \\ A \vdash C \end{array}}{A, B \vdash C}\, weak_L}{A \wedge B \vdash C}\, \wedge_L}{\vdash C}\, cut$$

We can look at what the cut-elimination procedure will do on this proof : it first explicits the conclusion of p as a pair of proofs a of A and b of B (e.g. by removing cuts between $X \vdash A \wedge B$ and $\vdash X$).

$$\dfrac{\dfrac{\begin{array}{c} a \\ \vdash A \end{array} \qquad \begin{array}{c} b \\ \vdash B \end{array}}{\vdash A \wedge B}\, \wedge_R \qquad \dfrac{\dfrac{\begin{array}{c} q \\ A \vdash C \end{array}}{A, B \vdash C}\, weak_L}{A \wedge B \vdash C}\, \wedge_L}{\vdash C}\, cut$$

The Hauptsatz

Then, it will remove the cut on $A \wedge B$.

$$\dfrac{\dfrac{a}{\vdash A} \qquad \dfrac{\dfrac{b}{\vdash B} \qquad \dfrac{\dfrac{q}{A \vdash C}}{A, B \vdash C}\, weak_L}{A \vdash C}\, cut}{\vdash C}\, cut$$

Finally, it removes the cut on the weakened formula B. By doing so, we discard b and the content of our initial "glue" between p and q by directly plugging a and q.

$$\dfrac{\dfrac{a}{\vdash A} \qquad \dfrac{q}{A \vdash C}}{\vdash C}\, cut$$

Curry-Howard-Lambek correspondance

What is a proof of the formula $A \Rightarrow (B \Rightarrow C) \vdash (A \wedge B) \Rightarrow C$?

$$\dfrac{\dfrac{\dfrac{[x]^*}{A \wedge B}\wedge_{e2}}{B} \qquad \dfrac{\dfrac{[y]^*}{A \wedge B}\wedge_{e1}}{A} \qquad \dfrac{[f]}{A \Rightarrow (B \Rightarrow C)}\Rightarrow_e}{B \Rightarrow C}\Rightarrow_e}{\dfrac{C}{(A \wedge B) \Rightarrow C}\Rightarrow_i^{x,y}}$$

Curry-Howard-Lambek correspondance

What is a program of the type $A \Rightarrow (B \Rightarrow C) \vdash (A \wedge B) \Rightarrow C$?

$$\dfrac{\dfrac{\dfrac{[x]^*}{x : A \wedge B}}{x.2 : B}\wedge_{e2} \qquad \dfrac{\dfrac{[y]^*}{y : A \wedge B}}{y.1 : A}\wedge_{e1} \qquad \dfrac{[f]}{f : A \Rightarrow (B \Rightarrow C)}}{f(y.1) : B \Rightarrow C}\Rightarrow_e}{f(y.1)(x.2) : C}\Rightarrow_e$$
$$\dfrac{f(y.1)(x.2) : C}{\lambda p.f(p.1)(p.2) : (A \wedge B) \Rightarrow C}\Rightarrow_i^{x,y}$$

Curry-Howard-Lambek correspondance

The execution of the program corresponds exactly to the cut elimination algorithm ! E.g. removing a Modus Ponens $\Rightarrow_e$ is function application : the "domain of discourse" is not (necessarily) an external world anymore and its objects can be used following this dynamic.

Curry-Howard-Lambek correspondance

What are the morphisms of a Closed Cartesian Category $\mathbf{C}$ in $\mathbf{C}(C^{B^A}, C^{A\times B})$?

Classical model theory misses a bit the point with constructive logics, because we don't want to interpret merely formulas i.e. provability but proofs as well (which are our individuals of logic). Category theory offers a framework for denotational semantics, in which :

- Formulas are interpreted as objects.
- Proofs are interpreted as morphisms.
- Composition and identity are interpreted by cut and identity.
- Proofs interpretations are invariant under cut-elimination.

Curry-Howard-Lambek correspondance

Some basic examples :

- The $\wedge, \Rightarrow$ fragment of IL is *isomorphic* to the simply-typed lambda calculus and is interpreted by CCCs.
- Classical logic is interpreted by partial orders (e.g. boolean topoi for set theory), reflecting its yes/no format ("is there a morphism ?" VS "which morphism(s) ?").
- The connection is extended to geometry with propositions as spaces and proofs as continuous maps, see e.g. HoTT (with equalities as homotopies).

A functional approach to mathematics

This correspondance takes functions as primitives rather than e.g. sets. For example, here are some definitions of System F (isomorphic to intuitionistic 2nd-order PA, while only having two connectives ($\to$ and $\forall$) and no axioms) :

- $A \vee B := \forall X.(A \to X) \to (B \to X) \to X$.
  $inl : \forall A.\forall B.A \to (A \vee B) = \lambda a \lambda f \lambda g.f(a)$.
- $\mathbb{N} := \forall X.X \to (X \to X) \to X$.
  $3 : \mathbb{N} = \lambda n \lambda s.s(s(s(n)))$.

Denotational models will ensure us that modulo cut-elimination, *inl* is the unique proof of its formula and proofs of $\mathbb{N}$ really are the natural numbers ($0 := \lambda n \lambda s.n$, $1 := \lambda n \lambda s.s(n)$, ...), characterized up to isomorphism.

A functional approach to mathematics

The prevalence of functions and isomorphisms in this setup emphasizes **structuralism**, in which structures emerging from the entailment relation supersedes any prior meaning ascribed to the formulas.

Category theory embodies this idea with the *principle of equivalence* : reasoning should be invariant under equivalent objects. This fails in ZFC where we can compare sets which can be encoded in different ways (e.g. does $(0,0) = 2$?).

Most type theories respect the principle of invariance[1]. HoTT was created by studying the possible content of proofs of equalities : they become equivalences, which is the univalence principle[2].

[1] All definable properties are invariant by isomorphism - taken from *Structuralism, Invariance, and Univalence (S.A., 2014)*

[2] I like to see the principles of invariance and univalence as soundness resp. completeness theorems for the principle of equivalence

Snack break & meditations

We achieved many benefits by explaining logic through proofs :

- The evanescent individuals and domains of discourse are internalized : they are proofs.
- We explicit and exploit the differences between e.g. Modus Ponens and implication : it's the same as building (introduction rule depending on the formalism) vs executing a program (with its intrinsic dynamics).
- The normative requests on logic is clarified as a prediction about the proof behaviors, beyond the axiomatic systems they are built in.
- We get additional, computational criterion to justify axioms.
- Answers can be used, generalizing the yes/no format of non-constructive logics.
- Powerful correspondances can be made through denotational models.

Snack break & meditations

However, we still have many issues to clarify :

- How do we know that a proof has its ascribed behavior ? We need a proof of the proof and so on ... The infinite regression is still there : incompleteness is not internalized.
- Introduction rules make us fall back in axiomatic systems.
- How to exploit negative propositions ?
- We have nothing to replace the proof/counter-model duality when looking for internal justifications.
- It's not clear when two proofs really are "the same", as different proofs have the same extension while lots of differences seems purely bureaucratic.
- The formulas are still something external.

-3. Interactive level: tests & games

Linear logic: primacy of tests over rules

How do we ensure that proofs do what we want ? We currently simply check that they are built "following our rules", but we can miss some of them : we could use other means to achieve the same behaviors, e.g. implement $A \Rightarrow B$ in Python.

We can first look at linear logic, where proofs are identified among more general objects by a geometric criteria.

Linear logic: primacy of tests over rules

In linear logic, we deal with situations instead of laws : e.g. implication is causal, i.e. a cut with A and $A \multimap B$ yields B while A and $A \multimap B$ are being "consumed". Laws are retrieved with an exponential modality $!A$, meaning that we can use A as much as we want (or discard it).

LL is discovered in coherence spaces : denotational semantics of IL where proofs of $A \Rightarrow B$ are split into two connectives, $!A \multimap B$.

LL isolates infinity in the contraction rule (which duplicates proofs) to remain constructive while retaining a lot of symmetries : in particular, the negation is involutive.

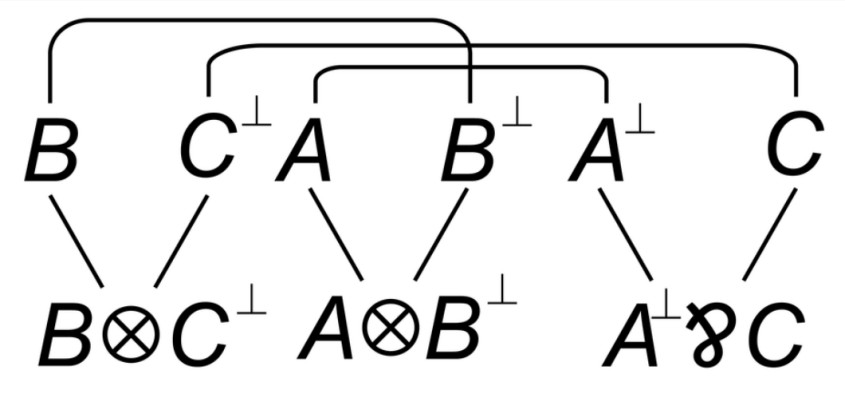
Linear logic: primacy of tests over rules

LL has its own favorite syntax (like natural deduction for IL) : proof structures. Allows to drastically simplify the identification of proofs, the cut-elimination and a non-sequential writing of proofs as graphs.

A proof structure is a proof net if its switching graphs are connected and acyclic $\Leftrightarrow$ if it corresponds to a derivation in LL sequent calculus.

Linear logic: primacy of tests over rules

Here is a proof-net for transitivity, with two switching graphs :



$$\vdash (A \multimap B) \multimap ((B \multimap C) \multimap (A \multimap C)) =$$

$$\vdash (A^{\perp} \parr B)^{\perp} \parr ((B^{\perp} \parr C)^{\perp} \parr (A^{\perp} \parr C)) =$$

$$\vdash (A \otimes B^{\perp}) \parr (B \otimes C^{\perp}) \parr (A^{\perp} \parr C) \equiv$$

$$\vdash (A \otimes B^{\perp}), (B \otimes C^{\perp}), (A^{\perp} \parr C)$$

Negation as testing

In order to understand the status of those tests, we can aim to define what is a test of A in the same spirit as the BHK approach :

- We know how to test atomic proofs.
- A test of $(a, b) : A \wedge B$ is either a test for $a : A$ **or** a test for $b : B$.
- A test of $\langle i; p \rangle : A \vee B$ is a test for $p : A$ **and** a test for $p : B$ (depending on i).
- A test of $f : A \Rightarrow B$ is a proof $a : A$ **and** a test for $f(a) : B$.

Negation as testing

Writing tests of A as $T(A)$, we have $T(A \wedge B) := T(A) \vee T(B)$, $T(A \vee B) := T(A) \wedge T(B)$, $T(A \Rightarrow B) := A \wedge T(B)$, ... So we are lead to identify tests of A with proofs of $\neg A$.

It then remains to generalize proofs to arguments, so we can have ones for A and $\neg A$ to make them interact. We can achieve this by different ways, e.g. with proof structures or by adding a "give up" axiom which allows to prove anything[1].

Then, the negation in LL is the inversion between tests and tested, a shift in point of view. The switching graphs of A become arguments for $\neg A$.

[1] Ludics does this approach, rebuilding LL starting from a generalized, alogical sequent calculus : a cut between arguments of A and $\neg A$ unfold them until one gives up.

Negation as testing

At the heart of LL lies an orthogonality relation. We write $a \perp b$ if the arguments a and b "interact properly". We then define the set of arguments passing the tests of A by $A^\perp = \{x \mid \forall a \in A, x \perp a\}$ and derive the expected properties from orthogonality such as :

- $A \subseteq A^{\perp\perp}$
- $A^\perp = A^{\perp\perp\perp}$
- $A \subseteq B \Rightarrow B^\perp \subseteq A^\perp$

A formula is then any set equal to its bi-orthogonal ("closed by interactions"). This gives us an internal characterization of behaviors, from interacting arguments. An appropriate choice of orthogonality leads to different LL semantics : phase spaces, (quantum / probabilistic) coherence spaces, geometry of interaction, ludics, ...

A dialectical interpretation

The study of logic at this level requires to interpret those tests, which don't directly fit in the previous paradigms.

We turn towards game semantics by interpreting **arguments as strategies** and **proofs as winning strategies**. Negation becomes the exchange of players. Then, for any strategy of A, either it is winning or there exists a strategy beating it.

Completeness is stated as follow : If A is a closed Π^1 formula and S is a winning strategy for the game $|A|$, then there is a cut-free proof a of A such that $S = |a|$. So, we recover our duality internally for 1st-order logic[1].

[1] Provided that we get back tests as well, it should be the case but I need to check it.

A dialectical interpretation

We succeeded to address much of our previous concerns :

- We achieve a new internal duality with tests.
- Rules can be justified internally, as an equivalence to a battery of tests characterizing some set of arguments.
- Negation is not an absence of proof anymore but the exchange between questions & answers : it becomes constructive.
- And more diverse advances (on implicit complexity, proof search, proof identifications, extensions of correspondances, semantics for cut-elimination itself ...).

The last uninvited guest is the referee, i.e. the rules of the game. We're now equipped to address our second starting question : can logic exist outside of axiomatic systems ?

-4. Deontic level: normativity

Prison break

Everything cannot be internalized : basic meaning of incompleteness. Common sense anyway : how would we trust a consistency proof of PA in PA, or a juge judging himself ? But the subsequent proliferation of axiomatic systems (predicativists, Tarski languages tower, ...) clashes with the want of unified reasoning : they have inherent limitations and are inconsistants together.

However, Gentzen analytic proofs prefigure a way out of the bunkers.

Prison break

Completeness only applies to Π^1 formulas (matching Σ^0_1 formulas in arithmetic). The same class of formulas benefits from the subformula property : their cut-free proofs only mention subformulas of its conclusion.

It means that when asking a question A, the given cut-free proof of A only uses components from the question itself : it doesn't depend on the axiomatic system in which it is formulated ! Such proofs are called analytic. We have a true internal completeness result : all cut-free proofs of A are already there.

This property is lost with more powerful systems such as PA with the induction on natural numbers. It offers a internal criteria separating simple 1st-order logic to be almost analytical (modulo cut-elimination) from richer, mathematical theories which are then synthetic.

A Kantian grid

We'll now start from zero and reconstruct the whole logical activity in the light of this Kantian-inspired reading grid :

	Analytic	Synthetic
A posteriori	Constat	Usine
A priori	Performance	Usage

A Kantian grid

We start from an analytic space, where we can do and say anything - but it remains meaningless. We can structure it by picking a model of computation : that'll orient us but not restrict our capacities.

Those objects may be executed, though they can always be manipulated as-is : a program can be read, an Erdös check can be framed, 2^{32} does not have to be "explicited" in 4294967296, ...

They are separated by the halting problem : we cannot reduce performances to constats, e.g. replace a calculator (efficient but subject to errors) by a logarithm table.

We'll then reconstruct every logical artifact (tests, proofs, formulas) from it.

A Kantian grid

We can make computational objects interact with each other, by picking an orthogonality relation. It allows us to finally escape the infinite regression asking for meta-meta-languages or meta-meta proofs : tests of A are themselves tested by arguments for A !

This is at the cost of not being able to say who's at fault when an interaction fails - a well-known phenomena in sciences (theory vs experience). We can then anchoring ourselves by fixing finite "primitive tests" to get a point of view between tests and tested.

Those tests which are called "usine" are made to predict the subsequent use of the tested objects, which is usually unbounded.

A Kantian grid

Girard gives an example with DVDs and DVD players :

To assert that something is a DVD, we'll make it pass a protocol with a fixed DVD player-test. But we want it to work with any DVD player, such DVD players being defined with the same protocol w.r.t. a fixed DVD-test.

Then, the logical certainty is the adequation between the usine and usage is the guarantee that any tested DVD and DVD player interact well together.

DVD-test	$\Leftrightarrow$ ✓	DVD player-test
$\Updownarrow$ ✓		$\Updownarrow$ ✓
DVD players	$\Leftrightarrow$?	DVDs

A Kantian grid

In traditional systems :

- The constat is the actual proofs & formulas syntax.
- The performance is the execution of untyped programs.
- The usine is the type checker, the algorithm $f(p, A)$ checking that p is a proof of A.
- The usage is the cut rule.

So with more technical terms, we have :

	Analytic	Synthetic
A posteriori	Constat: Data	Usine: Cut-free proofs
A priori	Performance: Programs	Usage: Deductive proofs

The new order

In this new approach, by picking a point of view, the subject is purely on the side of questions/formulas, which are synthetic (due to the choice of tests and their predictions), while the object constituting the answer/proof remains solely analytic.

Then, the usine/usage adequation comes by establishing (with external tools) that our tests are complementary : they may otherwise be too laxist to not catch future bad behaviors, or too strict to refuse good ones.

Traditionally, this equilibrium corresponds to the adequation between introduction and elimination rules : we don't miss any function with the behavior ascribed to its formula. This system-free logic - labelled apodictic - is quite satisfying, but we don't quite reach yet the expressivity of common 1st-order theories such as PA. What is missing ?

The new order

1st-order theories often use 2nd-order principles under disguise : typically, natural numbers and axiom schemes are Π^1 formulas. But then, the negation cannot be internalized because a Σ^1 would appear.

If we see predicates as universally quantified propositions, the negation of a 1st-order formula becomes the existence of a valuation to deny the predicate : this formula is traditionally expressed externally as a the existence of counter-model.

$$A[P] := \forall X. Axioms_P(X) \Rightarrow A[X/P]$$
$$\neg A[P] \equiv \exists X. Axioms_P(X) \wedge \neg A[X/P]$$

The new order

Which tests can we pick for the second-order existential ? The formula cannot guess its underlying witness in advance. However, the proof knows it so it can bring corresponding tests for the occurrences of the witness called the *mould*, filling the role of a user manual.

Moulds are subject to a conflict of interest : they should be strict to ensure that the answer behaves as expected, but may be laxist because they are on the answer side. The answer now incorporates a synthetic element, and the tests for molds commits us to a specific framework : Girard calls this part of logic epidictic.

This matches what we observe in more expressive theories : when asking a question about prime numbers, the answer may come with an obscure concept from number theory. We reach here along incompleteness the limits of unified reasoning, where tests are always either too laxist or too strict.

Conclusion

Transcendental syntax is still an early project (especially its handling of 2nd-order), it remains to see which fruits it will bear :

- Fragments of LL are reconstructed in apodictic logic, as well as new connectives. Two new constants replace LL four constants which lack correctness criteria.
- Arithmetic is reconstructed from epidictic logic. There are some hope to understand the content of negative theorems or even dependent types from it.
- Girard did a lot of work to inspire logic with quantum physics, though I don't understand those well. LL is a natural setup for it but several possible links need to be investigated : $\parr$ as superposition, cuts as measures, proofs as wave functions, ...

Conclusion

Thank you !

Any questions ?