The Internal Language of Quasi-toposes

Pietro Sabelli
j.w.w. Maria Emilia Maietti

Department of Logic
Institute of Philosophy of the Czech Academy of Sciences

112th Peripatetic Seminar on Sheaves and Logic
University of Nottingham, 28-29 March 2026

The Internal Language of a category

Idea

An *internal language* is a device to turn facts about a category $\mathcal{C}$ into statements of a formal theory $\mathsf{Int}(\mathcal{C})$, so to have a syntax-semantics correspondence.

$$\mathcal{C} \vDash \varphi \Leftrightarrow \mathsf{Int}(\mathcal{C}) \vdash \varphi$$

The Internal Language of a category

Idea

An *internal language* is a device to turn facts about a category $\mathcal{C}$ into statements of a formal theory $\mathsf{Int}(\mathcal{C})$, so to have a syntax-semantics correspondence.

$$\mathcal{C} \vDash \varphi \Leftrightarrow \mathsf{Int}(\mathcal{C}) \vdash \varphi$$

Example

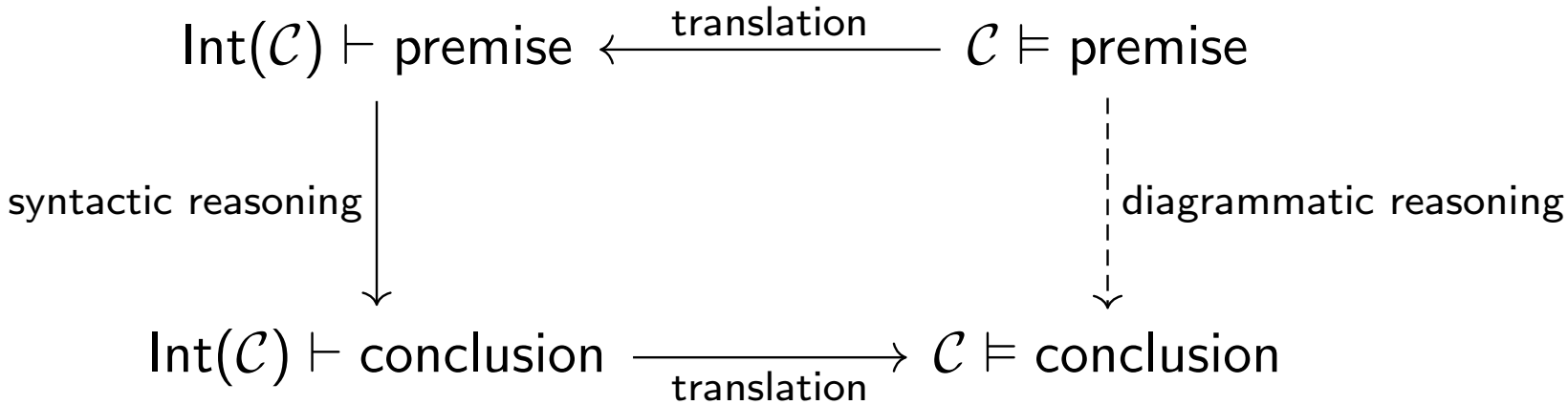
In a topos $\mathcal{E}$, an arrow $f : A \to B$ is an epimorphism whenever the following formula of $\mathsf{Int}(\mathcal{E})$ is derivable.

$$\forall y \in B \; \exists x \in A \,.\, y =_B f(x)$$

The above formula expresses the *surjectivity* of f seen as a function symbol of the syntax.

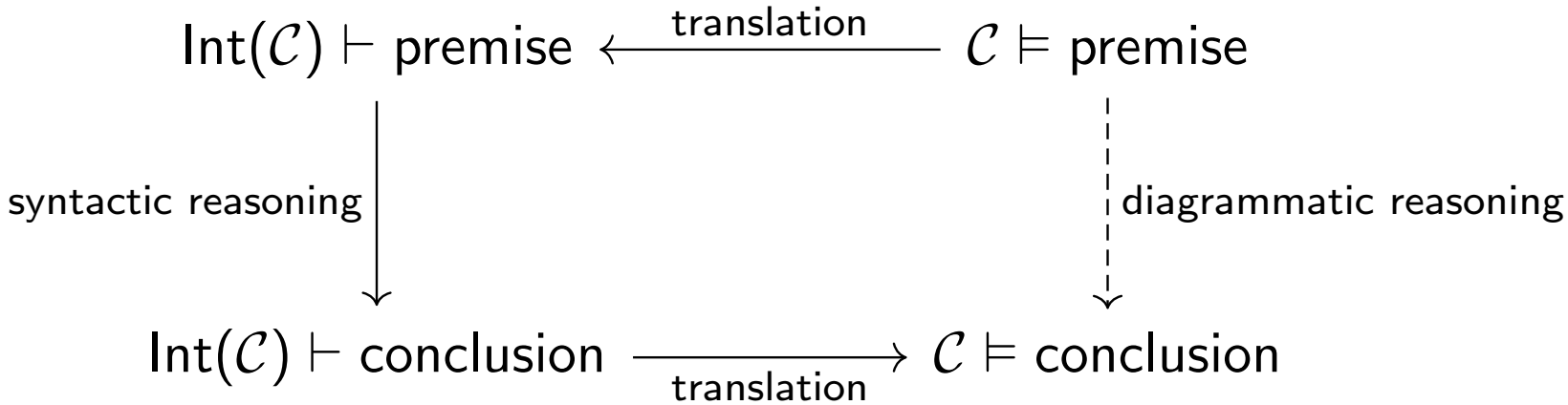
Internal Language: a powerful tool (cont'd)

The main use case is to avoid complex and unintuitive diagrammatic reasoning.



Internal Language: a powerful tool (cont'd)

The main use case is to avoid complex and unintuitive diagrammatic reasoning.



Example

Given a pair of jointly monic arrows $R \rightrightarrows A$ in an arithmetic pretopos, construct its transitive closure $R \rightarrowtail R^{+}$.

The Benabou-Mitchell language for toposes

The internal language of an (arithmetic) topos is given as a theory defined on top of a *simply typed*, *higher-order* calculus.

J. Lambek and P. J. Scott. "Introduction to higher order categorical logic". 1986

The Benabou-Mitchell language for toposes

The internal language of an (arithmetic) topos is given as a theory defined on top of a *simply typed*, *higher-order* calculus.

Observation

More of a *many-sorted logic* than a type theory: type constructors are characterized by ad hoc axioms. For example, terms of type $\mathbb{N}$ are governed by Peano's axioms.

J. Lambek and P. J. Scott. "Introduction to higher order categorical logic". 1986

The Benabou-Mitchell language for toposes

The internal language of an (arithmetic) topos is given as a theory defined on top of a *simply typed*, *higher-order* calculus.

Observation

More of a *many-sorted logic* than a type theory: type constructors are characterized by ad hoc axioms. For example, terms of type $\mathbb{N}$ are governed by Peano's axioms.

Theorem

There is a pair of functors

$$\mathrm{Synt} : \mathbf{Theory}_{st} \rightleftarrows \mathbf{Topos}_{st} : \mathrm{Int}$$

such that every topos $\mathcal{E}$ *is equivalent to* $\mathrm{Synt}(\mathrm{Int}(\mathcal{E}))$.

J. Lambek and P. J. Scott. "Introduction to higher order categorical logic". 1986

The Benabou-Mitchell language for toposes

The internal language of an (arithmetic) topos is given as a theory defined on top of a *simply typed*, *higher-order* calculus.

Observation

More of a *many-sorted logic* than a type theory: type constructors are characterized by ad hoc axioms. For example, terms of type $\mathbb{N}$ are governed by Peano's axioms.

Theorem

There is a pair of functors

$$\mathrm{Synt} : \mathbf{Theory}_{st} \rightleftarrows \mathbf{Topos}_{st} : \mathrm{Int}$$

such that every topos $\mathcal{E}$ *is equivalent to* $\mathrm{Synt}(\mathrm{Int}(\mathcal{E}))$.

However, it is not the case that every theory is equivalent to the internal language of some topos.

J. Lambek and P. J. Scott. "Introduction to higher order categorical logic". 1986

The Modular Correspondence

It provides *extensional dependently typed* calculi à la Martin-Löf for various categorical structures.

Categories	**Calculus**
Lex	$\mathsf{N}_1, \mathsf{Eq}, \Sigma$
Exact	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \mathsf{Q}$
LCC	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi$
Topos	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi, \Omega$

M. E. Maietti. "Modular correspondence between dependent type theories and categories including pretopoi and topoi". 2005

The Modular Correspondence

It provides *extensional dependently typed* calculi à la Martin-Löf for various categorical structures.

Categories	**Calculus**
Lex	$\mathsf{N}_1, \mathsf{Eq}, \Sigma$
Exact	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \mathsf{Q}$
LCC	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi$
Topos	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi, \Omega$

Theorem

There is a pair of functors

$$\mathbf{Th} \longleftarrow \mathbf{Th}_{st} \underset{\mathsf{Int}}{\overset{\mathsf{Synt}}{\rightleftarrows}} \mathbf{Top}_{st} \longrightarrow \mathbf{Top}$$

such that $\mathcal{E} \simeq \mathsf{Synt}(\mathsf{Int}(\mathcal{E}))$ *and* $\mathbf{T} \simeq \mathsf{Int}(\mathsf{Synt}(\mathbf{T}))$.

M. E. Maietti. "Modular correspondence between dependent type theories and categories including pretopoi and topoi". 2005

A Rosetta Stone for Toposes

Calculus	**Topos**
Context, (Closed) Type	Object
Dependent type	Arrow
Dependent mono-type	Monomorphism
Term	Section
Type constructors	*Topos structure*
Singleton set N_1	Terminal object
Dependent sum Σ	Dependent coproduct
Dependent product Π	Dependent product
Equality Eq	Equalizer
Omega type Ω	Subobject classifier

To correctly describe the Omega type Ω, one needs to exploit the correspondence between monomorphisms in the semantics and mono-types (i.e. types with at most one element) in the syntax.

Quasi-toposes

Definition

In a category, a monomorphism is *strong* if it right orthogonal to any epimorphism.

Definition

A *quasi-topos* is a finitely complete, finitely cocomplete, locally cartesian closed category with a *strong*-subobject classifier.

J. Penon. "Quasi-topos". 1973

Quasi-toposes

Definition

In a category, a monomorphism is *strong* if it right orthogonal to any epimorphism.

Definition

A *quasi-topos* is a finitely complete, finitely cocomplete, locally cartesian closed category with a *strong*-subobject classifier. Moreover, we say that a quasi-topos is:

- *solid* if the unique map $0 \to 1$ is a strong monomorphism;
- *boolean* if each poset of strong-subobjects is a Boolean algebra;
- *arithmetic* if it has a natural numbers object.

J. Penon. "Quasi-topos". 1973

Quasi-toposes

Definition

In a category, a monomorphism is *strong* if it right orthogonal to any epimorphism.

Definition

A *quasi-topos* is a finitely complete, finitely cocomplete, locally cartesian closed category with a *strong*-subobject classifier. Moreover, we say that a quasi-topos is:

- *solid* if the unique map $0 \to 1$ is a strong monomorphism;
- *boolean* if each poset of strong-subobjects is a Boolean algebra;
- *arithmetic* if it has a natural numbers object.

Example

The category of assemblies **Asm** is an arithmetic, solid, boolean quasi-topos, which is not a topos.

J. Penon. "Quasi-topos". 1973

Taming strong-monomorphisms

Inspired by the design of the Minimalist Foundation, we introduce a primitive kind of type to represent (proof-irrelevant) propositions.

Categories	**Calculus**
Lex	$\mathsf{N}_1, \mathsf{Eq}, \Sigma$
Exact	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \mathsf{Q}$
LCC	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi$
Quasi-topos	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi, \mathsf{N}_0, +, \mathsf{Q}, \mathcal{P}(1)$
Topos	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi, \Omega$

M. E. Maietti. "A minimalist two-level foundation for constructive mathematics". 2009

O. Wyler. "Lecture Notes on Topoi and Quasitopoi". 1991

Taming strong-monomorphisms

Inspired by the design of the Minimalist Foundation, we introduce a primitive kind of type to represent (proof-irrelevant) propositions.

Categories	**Calculus**
Lex	$\mathsf{N}_1, \mathsf{Eq}, \Sigma$
Exact	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \mathsf{Q}$
LCC	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi$
Quasi-topos	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi, \mathsf{N}_0, +, \mathsf{Q}, \mathcal{P}(1)$
Topos	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi, \Omega$

Observation

Sometimes quasi-toposes are defined with respect to an arbitrary subclass of monomorphisms stable under pullbacks, containing all equalizers, and closed under composition.

M. E. Maietti. "A minimalist two-level foundation for constructive mathematics". 2009

O. Wyler. "Lecture Notes on Topoi and Quasitopoi". 1991

A Rosetta Stone for quasi-toposes

Calculus	**Quasi-topos**
Context, (Closed) Type	Object
Dependent type	Arrow
Dependent mono-type	Monomorphism
Dependent proposition	Strong monomorphism
Term	Section
Type constructors	*Quasitopos structure*
Empty set N_0	Initial object
Singleton set N_1	Terminal object
Dependent sum Σ	Dependent coproduct
Dependent product Π	Dependent product
Disjoint sum $+$	Binary coproduct
Quotient set Q	Coequalizer
Equality Eq	Equalizer
Powerset of the singleton $\mathcal{P}(1)$	Strong-subobject classifier

A calculus for solid and arithmetic quasi-toposes

We can exploit modularity for quasi-toposes too.

Category	**Calculus**
Quasi-topos	$\mathsf{N}_1, \mathsf{Eq}, \Sigma, \Pi, \mathsf{N}_0, +, \mathsf{Q}, \mathcal{P}(1)$
+ solid	+ strong falsum constant
+ boolean	+ Law of the Excluded Middle (LEM)
+ arithmetic	+ $\mathbb{N}$

Recall that the falsum constant can be encoded impredicatively as

$$\bot :\equiv (\forall x \in \mathcal{P}(1))x$$

A falsum constant is *strong* if $\bot \to \mathsf{N}_0$.

Comparison between toposes and quasi-toposes

Proposition

A quasi-topos $\mathcal{E}$ is a topos if and only if one of the following two equivalent conditions is met:

1. (Semantical) every monomorphism of $\mathcal{E}$ is strong;
2. (Logical) the *Rule of Unique Choice* (RC!) holds in $\mathrm{Int}(\mathcal{E})$, i.e. a proof of a statement of the form

$$\forall x \in A \; \exists! y \in B \,.\, \varphi(x, y)$$

always yields an explicit function $f \in A \to B$ such that $\forall x \in A \,.\, \varphi(x, f(x))$.

Applications to Constructive Mathematics

Proposition

The category of assemblies **Asm** *validates the Type-theoretic Church's Thesis*

$$\forall f \in \mathbb{N} \to \mathbb{N}\ \exists e \in \mathbb{N}\,.\, f = \varphi_e \qquad \text{(TCT)}$$

where φ_e is the e-th Turing machine.

M. E. Maietti and P. Sabelli and D. Trotta. "Effectiveness and Continuity in Intuitionistic Quasi-toposes of Assemblies". To appear

Applications to Constructive Mathematics

Proposition

The category of assemblies **Asm** *validates the Type-theoretic Church's Thesis*

$$\forall f \in \mathbb{N} \to \mathbb{N}\ \exists e \in \mathbb{N}\,.\, f = \varphi_e \qquad \text{(TCT)}$$

where φ_e is the e-th Turing machine.

Observation

If **Asm** were a topos, the Rule of Unique Choice would hold in Int(**Asm**). However, already in Heyting Arithmetic with finite types

$$\mathrm{TCT} + \mathrm{LEM} + \mathrm{RC!} \vdash \bot$$

M. E. Maietti and P. Sabelli and D. Trotta. "Effectiveness and Continuity in Intuitionistic Quasi-toposes of Assemblies". To appear

Applications to Constructive Mathematics (cont'd)

Theorem

Both the Minimalist Foundation and the Calculus of Constructions are consistent with TCT *together with either one of the following:*

1. *classical logic;*
2. *Brouwer's intuitionistic principles.*

Proof.

By interpreting them in (the internal language of) **Asm** formalized in a suitable meta-theory. In particular: for 1. use **ZF** as meta-theory; for 2. use **IZF** + **BP** as meta-theory. □

M. E. Maietti and P. Sabelli. "Equiconsistency of the Minimalist Foundation with its classical version". 2025