What is a constructive and predicative foundation for mathematics?

Maria Emilia Maietti
University of Padova

Proof, Truth, Computation
Summer School on the Interactions between Modern Foundations of Mathematics and Contemporary Philosophy
21-25 July 2014, Fraueninsel
Coauthors in the talk

Giovanni Sambin (Padova)

(notion of constructive and predicative foundation)

Giuseppe Rosolini

(categorical notion of quotient completion)

Samuele Maschio

(realizability models of the Minimalist Foundation)
Abstract of our talk

- Notion of constructive foundation: as a two-level theory
- Notion of predicative foundation: need of sets + collections
- constructive $\Rightarrow$ intuitionistic?
- constructive $\Rightarrow$ predicative?
What is a constructive mathematics?

Recall the usual slogans:

- **constructive** mathematics =
  mathematics with intuitionistic logic
  (for ex: F. Richman)

- **constructive** mathematics =
  intuitionistic and predicative mathematics
  (for ex: Per Martin-Löf)
Why developing constructive mathematics?

usual motivation:
constructive mathematics is a bridge

abstract maths ➔ constructive maths = computable maths

computational maths ➔ constructions = computations
why does constructive mathematics bridge maths and computations?

because

CONSTRUCTIVE mathematics

must admit

a COMPUTATIONAL interpretation

⇓

CONSTRUCTIVE functions = COMPUTABLE functions
CONSTRUCTIVE maths

= 

COMPUTATIONAL maths by construction

(= IMPLICIT COMPUTATIONAL maths)
In "Computation and Logic in the Real World". LNCS 4497, 2007

developing CONSTRUCTIVE mathematics

= 

developing COMPUTATIONAL mathematics

without using TURING MACHINES

EXPLICITELY
What is a constructive foundation?

A theory is **constructive** = its proofs have a **computational interpretation**
i.e., there exists a computable model, called **realizability model**, where

\[ \exists x \in A \phi(x) \text{ true} \quad \text{under hypothesis } \Gamma \]

\[ \Downarrow \]

\[ \phi(c) \text{ true} \quad \text{under hypothesis } \Gamma \]

We extract a functional **program** calculating \( c(\Gamma) \) from existential statements
under hypothesis \( \Gamma \)
Can you do constructive mathematics without a foundation?

NO, doing proofs formalizable (a priori) in a foundation for constructive mathematics guarantees a COMPUTATIONAL INTERPRETATION if the foundations does have it!! otherwise work explicitely with Turing machines...
Why isn’t CONSTRUCTIVE MATHEMATICS mainstream maths yet?

..and why is it mostly developed by some logicians???

An exception was ERRET BISHOP, who said

“This book (Bishop-Bridges’s “Constructive Analysis”) has a threefold purpose:
1. to present the constructive point of view,
2. to show that the constructive program can succeed, (he reproduced a large portion of functional analysis)
3. and to lay a foundation for further work.

These immediate ends tend to an ultimate goal
- to hasten the inevitable day
when constructive mathematics will be the accepted norm.
Why isn’t **CONSTRUCTIVE MATHEMATICS** mainstream maths yet?

To develop **CONSTRUCTIVE MATHEMATICS**

we need a **FOUNDATION**

because using **INTUITIONISTIC LOGIC** is not enough!!

⇒ you need a **formal CONSTRUCTIVE SET theory**!!
Why does CONSTRUCTIVE MATHEMATICS need a foundation?

to decide:

\[
\begin{align*}
\text{do you accept to use} & \quad \begin{cases} 
\text{the axiom of choice?} \\
\text{the axiom of unique choice?} \\
\text{Zorn’s lemma?} \\
\ldots?
\end{cases}
\end{align*}
\]

a bit like in classical mathematics but NOT quite!!!
what is a foundation for constructive mathematics?

in the literature

there is a VARIETY of so-called constructive foundations
### Plurality of constructive foundations

<table>
<thead>
<tr>
<th>Classical</th>
<th>Constructive</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONE standard</td>
<td>NO standard</td>
</tr>
</tbody>
</table>

**Impredicative**
- Zermelo-Fraenkel set theory
  - Friedman's IZF
  - Internal theory of topoi
  - Coquand's Calculus of Constructions

**Predicative**
- Feferman's explicit maths
  - Aczel's CZF
  - Martin-Löf's type theory
  - Feferman's constructive expl. maths
Plurality of constructive foundations

need for a minimalist foundation

as argued in

[MS’05] M.E.M. & G. Sambin “Toward a minimalist foundation for constructive mathematics” in "From sets to types..", L. Crosilla and P. Schuster eds. OUP, 2005
### Plurality of Constructive Foundations

<table>
<thead>
<tr>
<th>Classical</th>
<th>Constructive</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONE standard</td>
<td>NO standard</td>
</tr>
<tr>
<td>Impredicative</td>
<td></td>
</tr>
<tr>
<td>Zermelo-Fraenkel set theory</td>
<td></td>
</tr>
<tr>
<td>Friedman's IZF</td>
<td>Aczel's CZF</td>
</tr>
<tr>
<td>Internal theory of topoi</td>
<td>Martin-Löf's type theory</td>
</tr>
<tr>
<td>Coquand's Calculus of Constructions</td>
<td>Feferman's constructive expl. maths</td>
</tr>
</tbody>
</table>

What common core??
**how to design a minimalist foundation?**

↓

need of a notion of constructive foundation
What language to choose for the foundation?

A foundation can be formulated as a:

- TYPE THEORY
- AXIOMATIC SET THEORY
- CATEGORICAL UNIVERSE
Idea of CONSTRUCTIVE FOUNDATION

A FORMAL Constructive Foundation should bridge:

- LANGUAGE of abstract maths (of commonly used set theory)
- PROGRAMMING LANGUAGE (of type theory) for extraction of programs from proofs
Why type theory as PROGRAMMING LANGUAGE?

because

\[
\text{type theory} = \quad \text{a paradigm of a programming language}
\]

that can be viewed as a set theory

and as a base for an INTERACTIVE THEOREM PROVER
need of an INTERACTIVE THEOREM PROVER

to actually extract programs from constructive proofs

better use an INTERACTIVE THEOREM PROVER

to manipulate COMPUTER-AIDED FORMALIZED PROOFS
Our basic notion of **CONSTRUCTIVE FOUNDATION**

A FORMAL Constructive Foundation should bridge

- **LANGUAGE** of abstract maths
  - usual set theoretic language
- **INTENSIONAL** type theory
  - base for an INTERACTIVE prover
  - realizability model
  - (for proofs-as-programs extraction)
Need of \textit{INTENSIONAL TYPE THEORY} for computer formalization

why??
Need of **INTENSIONAL TYPE THEORY** for computer formalization

**INTENSIONAL** type theory  (as Martin-Löf’s type theory)

enjoys

- extraction of **PROGRAMS** from **PROOFS**
- **DECIDABLE** type checking for Program correctness
- **CONSISTENCY** (via normalization of proofs)

⇒

it is a **BASIC RELIABLE THEORY** for an **INTERACTIVE THEOREM PROVER** !!!
**LEVELS in our notion of constructive foundation**

a constructive foundation should be equipped with

<table>
<thead>
<tr>
<th>Foundation</th>
<th>extensional level (used by mathematicians to do their proofs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>↓</td>
</tr>
<tr>
<td></td>
<td>intensional level (language of computer-aided formalized proofs)</td>
</tr>
<tr>
<td></td>
<td>↓</td>
</tr>
<tr>
<td></td>
<td>realizability level (used by computer scientists to extract programs)</td>
</tr>
</tbody>
</table>

but ONLY the first TWO LEVELS form the actual FOUNDATION!!
**WHAT LINKS the intensional/extensional LEVELS**

<table>
<thead>
<tr>
<th>Foundation</th>
<th>intensional level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \downarrow ) (interpreted via a <strong>quotient completion</strong> over)</td>
</tr>
<tr>
<td></td>
<td>extensional level</td>
</tr>
</tbody>
</table>

in [M.Sambin'05]:  

the extensional level must be **abstracted** from  
the intensional one  
via a **quotient completion**  
following Sambin’s **forget-restore principle**  
in a LOCAL and MODULAR way  
preserving the intensional logic

⇒ **extensional sets are quotients of intensional sets!**
WARNING: in our notion of **CONSTRUCTIVE FOUNDATION**

PS: “all functions are COMPUTABLE” holds ONLY at the realizability level because it canNOT be lifted at the extensional level for compatibility with classical extensional levels

\[\downarrow\]

the realizability model to extract programs from constructive proofs is a third separated level!
Order of conception among levels

1. first, design the **intensional level**
   as a basic reliable theory enjoying extraction of programs from proofs

2. then, make the **quotient completion** of the first

3. lastly design the **extensional level** to be interpreted at the **intensional** one
   via the **quotient completion**
   TO VIEW **extensional sets** as **quotients** of **intensional sets**
NOTION of QUOTIENT COMPLETION

expressed in the language of CATEGORY THEORY

and analyzed in


USEFULNESS of QUOTIENT COMPLETION

makes EXPLICIT HIDDEN structures used to interpret

UNDECIDABLE equalities of an extensional theory

↓

in an intensional theory with DECIDABLE equalities

WARNING: avoid to work DIRECTLY in the QUOTIENT COMPLETION over the INTENSIONAL level

BUT do your mathematical proof at EXTENSIONAL level
what language to choose to design a foundation?

all the following:

**TYPE THEORY:** local set existence + functions as programs

usefulness: extraction of programs from proofs

suitable for: the INTENSIONAL level

\[\downarrow\text{increasing abstraction}\]

as an **LOCAL AXIOMATIC SET THEORY:** local set existence

usefulness: formalization of math proofs as in common practice

suitable for: the EXTENSIONAL level

\[\downarrow\text{increasing abstraction}\]

as a **CATEGORY:** set universes as algebraic structures

usefulness: unification of different foundations

suitable for: LINKING levels and describe their mathematical structure
Examples of our two-level constructive foundation

the two-level Minimalist Foundation in [M.09]

Extensional Minimal Type Theory

↓

Intensional Minimal Type Theory

Our two level Minimalist Foundation

the version is [M’09]

- its intensional level
  = a PREDICATIVE VERSION of Coquand’s Calculus of Constructions (Coq).
  = a FRAGMENT of Martin-Löf’s intensional type theory + one UNIVERSE

- its extensional level
  has a PREDICATIVE LOCAL set theory
  (NO choice principles)
Main advantages of our two level Minimalist Foundation

very weak foundation!!!

- its intensional level is a fragment Martin-Löf’s type theory to be compatible with PREDICATIVE CLASSICAL foundation à la Feferman with propositions given primitively

- its extensional level is also compatible with PREDICATIVE CLASSICAL foundation interpretable in $\widehat{ID}_1$
Another example of our constructive foundation

internal language of locally cartesian closed pretopoi

as in [M.05]

⇓

Martin-Loef’s type theory

another example of our constructive foundation?

| Aczel’s CZF (usual math language) | ↓ (interpreted in) | Martin-Löf’s type theory (reliable programming language) |

- Is this interpretation **LOCAL and MODULAR** on the **set theoretic constructors**?
  
  *warning: CZF+ excluded middle= ZF*

- Is the use of **choice principles** (beside universes) in the interpretation?

- can all **CZF-sets** be viewed as **quotients of intensional sets**?
a KEY application of our MINIMALIST foundation

other constructive foundations can be view as extensions
of the MINIMALIST foundation

⇒

the MINIMALIST foundation allows a MODULAR REUSE of PROOFS
at the right level (intensional or extensional)

our goal: build a MINIMALIST INTERACTIVE THEOREM PROVER!
Constructive $\Rightarrow$ Intuitionistic??
Constructive $\Rightarrow$ Intuitionistic??

yes, if we give the following proof-theoretic definition:

\[
\begin{align*}
\vdash_T \phi \lor \psi & \implies \vdash_T \phi \quad \text{or} \quad \vdash_T \psi \\
\vdash_T \exists x \in A \phi(x) & \implies \vdash_T \phi(t) \quad \text{for some term } t
\end{align*}
\]

but we want a more semantical definition
Constructive $\Rightarrow$ Intuitionistic??

Constructive $\Rightarrow$ NOT classical !!

if we STRENGTHEN our basic notion of constructive foundation
to that in [M.-Sambin’05]

- a constructive foundation = it is a two-level foundation s.t.
  - extensional level
    - $\Downarrow$ (interpreted via a quotient completion over)
  - intensional level is a PROOFS-as-PROGRAMS theory
  - i.e. is consistent with Axiom of Choice + Formal Church Thesis
Axiom of Choice

\((AC'_{Nat})\) \quad \forall x \in A \, \exists y \in B \, R(x, y) \quad \longrightarrow \quad \exists f \in A \to B \quad \forall x \in A \, R(x, f(x))
Formal Church thesis

\[(CT)\quad \forall f \in \text{Nat} \rightarrow \text{Nat} \quad \exists e \in \text{Nat} \quad \left( \forall x \in \text{Nat} \quad \exists y \in \text{Nat} \quad T(e, x, y) \land U(y) =_{\text{Nat}} f(x) \right)\]
Usefulness of our notion of proofs-as-programs theory

very useful to discriminate:

• constructive versus classical theories

• intensional versus extensional theories
No classical foundation is a proofs-as-programs theory

recall:

\[
\text{PROOFS-as-PROGRAMS theory = theory consistent with Axiom of Choice + Formal Church Thesis}
\]

\[
PA = \text{Peano Arithmetics}
\]

\[
PA + CT + AC \vdash 0 = 1
\]
NO classical theory including arithmetics is a proofs-as-programs theory and hence is constructive in our sense.
INCONSISTENCY of $\text{CT+AC+ extensionality of functions}$

recall

$HA^\omega + \text{CT} + AC + \text{extensionality of functions} \vdash \bot$

\[\Downarrow\]

No theory including arithmetics with function extensionality is

CONSISTENT with $\text{CT+AC}$

\[\Downarrow\]

this motivated in [M.-Sambin'05]

constructive foundation

= a TWO LEVEL foundation:

\[\{\begin{align*}
\text{extensional level with function extensionality} \\
\text{intensional level consistent with CT+AC}
\end{align*}\]\n

Constructive $\Rightarrow$ Predicative??
What is a predicative foundation?

**predicative**: no circular definitions

\[\downarrow\]

NO powersets

\[\downarrow\]

**logic + sets** is NOT enough
to formalize **topology** in a **predicative** foundation

\[\downarrow\]

need of power**collections**
a predicative foundation = a predicative system

with two different sizes of entities:

SETS + COLLECTIONS

( or SETS + CLASSES )

( or SETS + UNIVERSES )
What is a predicative foundation?

A predicative system equipped with:

- inductively generated sets
  - i.e. sets are equipped with INDUCTION on their generated elements

- collections: entities just defined by their elements in a predicative way
  - (NOT necessarily equipped with INDUCTION principle)
**Constructive ⇒ Predicative??**

conjectured, yes for a **STRONGLY constructive foundation**
which is a **two-level foundation s.t.**

<table>
<thead>
<tr>
<th>Foundation</th>
<th>extensional level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>↓ (interpreted via a <strong>quotient completion</strong> over)</td>
</tr>
<tr>
<td>intensional level is a <strong>STRONG PROOFS-as-PROGRAMS theory</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>i.e. is <strong>consistent with Martin Löf’s strong existential quantifier</strong></td>
</tr>
<tr>
<td></td>
<td>+ <strong>Formal Church Thesis</strong></td>
</tr>
</tbody>
</table>

52
Martin Löf’s Strong existential quantifier

written $\exists = \Sigma$

- it satisfies

$$\Gamma \vdash \exists_{x \in A} B(x) \iff \text{there exists a term } c \text{ depending on } \Gamma \text{ s.t. } \Gamma \vdash B(c)$$

- and there exists a surjective map of proofs with a retraction

$$\text{Proofs}(\Gamma \vdash \exists_{x \in A} B(x)) \xrightarrow{\text{surjective map}} \bigcup_{c \in \text{Term}(\Gamma)} \text{Proofs}(\Gamma \vdash B(c))$$
Martin L"of’s Strong existential quantifier

\[ b \in B \quad d \in C(b) \]
\[ \langle b, d \rangle \in \exists x \in B C(x) \]

with an elimination rule yielding to

two projections

\[ d \in \exists x \in B C(x) \]
\[ \pi_1(d) \in B \]
\[ d \in \exists x \in B C(x) \]
\[ \pi_2(d) \in C(\pi_1(d)) \]

\[ \Rightarrow \] strong existential quantifier validates Axiom of choice
Example of **strong proofs-as-programs theories**

<table>
<thead>
<tr>
<th>STRONGLY PROOFS-as-PROGRAMS theory</th>
<th>= theory consistent with</th>
</tr>
</thead>
<tbody>
<tr>
<td>Martin-Löf’s strong existential quantifier + Formal Church Thesis</td>
<td></td>
</tr>
</tbody>
</table>

- fragments of Martin-Löf’s intensional theory
  - with NO $\xi$-rule but with **explicit substitution rules**

- the **intensional** level of our **two-level MINIMALIST** foundation
  - (work in progress with Samuele Maschio)
A counterexample of strong proofs-as-programs theories

The Calculus of constructions (CoC) in [Coq90] is NOT a STRONG proofs-as-programs theory

\[ CoC + \text{Martin-Löf's strong existential quantifier } \subseteq \]

because \( \exists x \in \text{Prop} \quad \text{tt} \in \text{Prop} \)

and first projection gives a retraction of the obvious map

\[ \text{Prop} \to \exists x \in \text{Prop} \quad \text{tt} \]

sending \( \phi \) to \( (\phi, \ast) \).
CONCLUSIONS

- we defined a notion of constructive foundation
- we defined a notion of STRONGLY constructive foundation

open issue:

STRONGLY constructive $\Rightarrow$ predicative??

i.e.

is there an IMPREDICATIVE theory consistent

with formal Church thesis CT + Martin-Löf’s strong existential quantifier ??

our expectation:

the notion of STRONGLY constructive foundation

characterizes constructive and predicative foundations.