A Kleene realizability semantics for the minimalist foundation

S.Maschio (joint work with M.E.Maietti)

Università degli Studi di Padova

Department of Mathematics
University of Padua

TYPES 2014
Paris, May 12-15

A foundation for constructive mathematics

abstract mathematics

↑

Constructive mathematics = bridge

↓

computational mathematics

A foundation for constructive mathematics

abstract mathematics

↑

Constructive mathematics = bridge

↓

computational mathematics

Constructive mathematics=*implicit* computational mathematics!

Many foundations in (constructive) mathematics...

classical	constructive
ONE standard	NO standard

Many foundations in (constructive) mathematics...

	classical	constructive
	ONE standard	NO standard
impredicative	Zermelo-Fraenkel set theory	{ internal theory of topoi Coquand's Calculus of Constructions

Many foundations in (constructive) mathematics...

	classical	constructive
	ONE standard	NO standard
impredicative	Zermelo-Fraenkel set theory	{ 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

Many foundations in (constructive) mathematics...

	classical	constructive
	ONE standard	NO standard
impredicative	Zermelo-Fraenkel set theory	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

Necessity of a common core: the minimalist foundation (Maietti, Sambin 2005)

The formal system of the Minimalist Foundation (Maietti (2009)):

The formal system of the Minimalist Foundation (Maietti (2009)):

- 2-level theory based on versions of Martin-Löf Type Theory

The formal system of the Minimalist Foundation (Maietti (2009)):

- 2-level theory based on versions of Martin-Löf Type Theory
  + a primitive notion of propositions

The formal system of the Minimalist Foundation (Maietti (2009)):

- 2-level theory based on versions of Martin-Löf Type Theory
  + a primitive notion of propositions
- an intensional level (mTT): computational contents of proofs

The formal system of the Minimalist Foundation (Maietti (2009)):

- 2-level theory based on versions of Martin-Löf Type Theory
  + a primitive notion of propositions
- an intensional level (mTT): computational contents of proofs
- an extensional level (emTT): where to develop ordinary mathematics.

The intensional level has four sorts of types:

The intensional level has four sorts of types:

- **set**: basic $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}$,

The intensional level has four sorts of types:

- **set**: basic $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}$, all small propositions

The intensional level has four sorts of types:

- **set**: basic $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}$, all small propositions and constructors Π, Σ, $+$ and **list**;

The intensional level has four sorts of types:

- **set**: basic $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}$, all small propositions and constructors Π, Σ, $+$ and **list**;
- **coll**: all sets,

The intensional level has four sorts of types:

- **set**: basic $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}$, all small propositions and constructors Π, Σ, $+$ and **list**;
- **coll**: all sets, all propositions,

The intensional level has four sorts of types:

- **set**: basic $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}$, all small propositions and constructors Π, Σ, $+$ and **list**;
- **coll**: all sets, all propositions, the type of (codes for) small propositions prop_s,

The intensional level has four sorts of types:

- **set**: basic $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}$, all small propositions and constructors Π, Σ, $+$ and **list**;
- **coll**: all sets, all propositions, the type of (codes for) small propositions prop_s, $A \to \mathsf{prop}_s$ with A set and constructor Σ;

The intensional level has four sorts of types:

- **set**: basic $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}$, all small propositions and constructors Π, Σ, $+$ and **list**;
- **coll**: all sets, all propositions, the type of (codes for) small propositions prop_s, $A \rightarrow \mathsf{prop}_s$ with A set and constructor Σ;
- **prop**: $\perp$ and closed under connectives $\wedge, \vee, \rightarrow$, collection bounded quantifiers and **Id** in collections.

The intensional level has four sorts of types:

- **set**: basic $\mathbf{N}_0, \mathbf{N}_1, \mathbf{N}$, all small propositions and constructors Π, Σ, $+$ and **list**;
- **coll**: all sets, all propositions, the type of (codes for) small propositions prop_s, $A \rightarrow \mathsf{prop}_s$ with A set and constructor Σ;
- **prop**: $\perp$ and closed under connectives $\wedge, \vee, \rightarrow$, collection bounded quantifiers and **Id** in collections.
- **prop**$_s$ is **prop** with only set bounded quantifiers and **Id**s relative to sets.

According to Sambin, *A minimalist foundation at work* (2011):

A foundation of mathematics is a choice of what is considered relevant.

Some relevant principles

[M.E.Maietti, G. Sambin, *Toward a minimalist foundation for constructive mathematics* (2005)]

Some relevant principles

[M.E.Maietti, G. Sambin, *Toward a minimalist foundation for constructive mathematics* (2005)]

AC (axiom of choice):

Some relevant principles

[M.E.Maietti, G. Sambin, *Toward a minimalist foundation for constructive mathematics* (2005)]

AC (axiom of choice):

$$(\forall x : A)(\exists y : B)R(x, y) \rightarrow (\exists f : A \rightarrow B)(\forall x : A)R(x, \mathbf{App}(f, x))$$

Some relevant principles

[M.E.Maietti, G. Sambin, *Toward a minimalist foundation for constructive mathematics* (2005)]

AC (axiom of choice):

$$(\forall x : A)(\exists y : B)R(x, y) \to (\exists f : A \to B)(\forall x : A)R(x, \mathbf{App}(f, x))$$

Every *A*-total relation admits a choice operation

Some relevant principles

[M.E.Maietti, G. Sambin, *Toward a minimalist foundation for constructive mathematics* (2005)]

AC (axiom of choice):

$$(\forall x : A)(\exists y : B)R(x, y) \to (\exists f : A \to B)(\forall x : A)R(x, \mathbf{App}(f, x))$$

Every A-total relation admits a choice operation

CT (formal Church Thesis):

CT (formal Church Thesis):

$$(\forall f : N \to N)(\exists e : N)(\forall x : N) App(f, x) =_N \{e\}(x)$$

CT (formal Church Thesis):

$$(\forall f : N \to N)(\exists e : N)(\forall x : N)App(f, x) =_N \{e\}(x)$$

Every function between natural numbers is recursive.

CT (formal Church Thesis):

$$(\forall f : N \to N)(\exists e : N)(\forall x : N)App(f, x) =_N \{e\}(x)$$

Every function between natural numbers is recursive.

ECT (Extended Church Thesis):

ECT (Extended Church Thesis):

$$(\forall x : N)(\exists y : N)R(x, y) \to (\exists e : N)(\forall x : N)R(x, \{e\}(x))$$

ECT (Extended Church Thesis):

$$(\forall x : N)(\exists y : N)R(x, y) \to (\exists e : N)(\forall x : N)R(x, \{e\}(x))$$

It is equivalent to $\mathbf{AC}_{N,N} + \mathbf{CT}$

EXT (Extensionality of functions):

EXT (Extensionality of functions):

$$(\forall x : A)\mathsf{Id}(B, b, c) \rightarrow \mathsf{Id}((\Pi x : A)B, (\lambda x)b, (\lambda x)c)$$

EXT (Extensionality of functions):

$$(\forall x : A)\mathsf{Id}(B, b, c) \rightarrow \mathsf{Id}((\Pi x : A)B, (\lambda x)b, (\lambda x)c)$$

Equal terms in context give rise to equal functions

However it is well known that:

$$\mathbf{AC} + \mathbf{CT} + \mathbf{EXT} \vdash \bot$$

However it is well known that:

$$\mathbf{AC} + \mathbf{CT} + \mathbf{EXT} \vdash \bot$$

This is a reason for having 2 levels in the minimalist foundation!

However it is well known that:

$$\mathbf{AC} + \mathbf{CT} + \mathbf{EXT} \vdash \bot$$

This is a reason for having 2 levels in the minimalist foundation!

1. **mTT** should be consistent with **AC** and **CT**(proofs as programs) [this is work in progress]

However it is well known that:

$$\mathbf{AC} + \mathbf{CT} + \mathbf{EXT} \vdash \bot$$

This is a reason for having 2 levels in the minimalist foundation!

1. **mTT** should be consistent with **AC** and **CT**(proofs as programs) [this is work in progress]
2. in **emTT**, **EXT** must be provable (ordinary mathematics is extensional!)

However it is well known that:

$$\mathbf{AC} + \mathbf{CT} + \mathbf{EXT} \vdash \perp$$

This is a reason for having 2 levels in the minimalist foundation!

1. **mTT** should be consistent with **AC** and **CT**(proofs as programs) [this is work in progress]
2. in **emTT**, **EXT** must be provable (ordinary mathematics is extensional!)

The Kleene realizability model for **mTT**.

Why a Kleene realizability model for the Minimalist Foundation?

The Kleene realizability model for **mTT**.

Why a Kleene realizability model for the Minimalist Foundation?
To prove the consistency of $\mathbf{emTT} + \mathbf{ECT}$

The Kleene realizability model for **mTT**.

Why a Kleene realizability model for the Minimalist Foundation?
To prove the consistency of **emTT** + **ECT**

1. Extend the Kleene realizability model for **HA** to **mTT**,

The Kleene realizability model for **mTT**.

Why a Kleene realizability model for the Minimalist Foundation?
To prove the consistency of **emTT** + **ECT**

1. Extend the Kleene realizability model for **HA** to **mTT**,
2. then extend this model to the extensional level following interpretation in [Maietti'09] with coherent isomorphisms and working in the extensional completion in [Maietti-Rosolini'13].

The Kleene realizability model for **mTT**.

Why a Kleene realizability model for the Minimalist Foundation?
To prove the consistency of **emTT** + **ECT**

1. Extend the Kleene realizability model for **HA** to **mTT**,
2. then extend this model to the extensional level following interpretation in [Maietti'09] with coherent isomorphisms and working in the extensional completion in [Maietti-Rosolini'13].

Kleene realizability is

Kleene realizability is

- the main concrete instance of BHK;

Kleene realizability is

- the main concrete instance of BHK;
- based on the *double identity* of natural numbers: numbers and codes for recursive functions;

Kleene realizability is

- the main concrete instance of BHK;
- based on the *double identity* of natural numbers: numbers and codes for recursive functions;
- natural numbers are used as witnesses of the provability of formulas ($n \Vdash \phi$).

Kleene realizability is

- the main concrete instance of BHK;
- based on the *double identity* of natural numbers: numbers and codes for recursive functions;
- natural numbers are used as witnesses of the provability of formulas ($n \Vdash \phi$).

Our interpretation

$$\mathbf{mTT} \rightarrow \hat{\mathbf{ID}}_1$$

Our interpretation

$$\mathbf{mTT} \rightarrow \hat{\mathbf{ID}}_1$$

$\hat{\mathbf{ID}}_1$ predicative theory

Our interpretation

$$\mathbf{mTT} \rightarrow \hat{\mathbf{ID}}_1$$

$\hat{\mathbf{ID}}_1$ predicative theory

PA (Peano Arithmetic) $+$ some (not necessarily least) fix points for positive arithmetical operators.

A **type** (of any sort)

A **type** (of any sort)

is interpreted as a pair

$$(\mathcal{J}(A), \cong_{\mathcal{J}(A)})$$

$$A \textbf{ type } \text{(of any sort)}$$

is interpreted as a pair

$$(\mathcal{J}(A), \cong_{\mathcal{J}(A)})$$

where

1. $\mathcal{J}(A)$ is a definable class of $\widehat{\mathbf{ID}}_1$;

A **type** (of any sort)

is interpreted as a pair

$$(\mathcal{J}(A), \cong_{\mathcal{J}(A)})$$

where

1. $\mathcal{J}(A)$ is a definable class of $\widehat{\mathbf{ID}}_1$;
2. $\cong_{\mathcal{J}(A)}$ is a definable equivalence relation on $\mathcal{J}(A)$;

$$A \textbf{ type } \text{(of any sort)}$$

is interpreted as a pair

$$(\mathcal{J}(A), \cong_{\mathcal{J}(A)})$$

where

1. $\mathcal{J}(A)$ is a definable class of $\hat{\mathbf{ID}}_1$;
2. $\cong_{\mathcal{J}(A)}$ is a definable equivalence relation on $\mathcal{J}(A)$;

according to Kleene Realizability.

$$A\ \textbf{type}\ \text{(of any sort)}$$

is interpreted as a pair

$$(\mathcal{J}(A), \cong_{\mathcal{J}(A)})$$

where

1. $\mathcal{J}(A)$ is a definable class of $\hat{\mathbf{ID}}_1$;
2. $\cong_{\mathcal{J}(A)}$ is a definable equivalence relation on $\mathcal{J}(A)$;

according to Kleene Realizability.
Terms are interpreted as (codes for) recursive functions with domain given by the interpretation of the context.

Key points:

Key points:

1. for ϕ proposition,

Key points:

1. for ϕ proposition, $\mathcal{J}(\phi) := \{x | x \Vdash \phi\}$

Key points:

1. for ϕ proposition, $\mathcal{J}(\phi) := \{x | x \Vdash \phi\}$ and $\cong_{\mathcal{J}(\phi)}$ is total (proof-irrelevance);

Key points:

1. for ϕ proposition, $\mathcal{J}(\phi) := \{x|x \Vdash \phi\}$ and $\cong_{\mathcal{J}(\phi)}$ is total (proof-irrelevance);
2. equality in Π sets is interpreted as extensional equality!

Key points:

1. for ϕ proposition, $\mathcal{J}(\phi) := \{x | x \Vdash \phi\}$ and $\cong_{\mathcal{J}(\phi)}$ is total (proof-irrelevance);
2. equality in Π sets is interpreted as extensional equality!
3. for basic sets $\cong_{\mathcal{J}(A)}$ is the numerical equality;

Key points:

1. for ϕ proposition, $\mathcal{J}(\phi) := \{x|x \Vdash \phi\}$ and $\cong_{\mathcal{J}(\phi)}$ is total (proof-irrelevance);
2. equality in Π sets is interpreted as extensional equality!
3. for basic sets $\cong_{\mathcal{J}(A)}$ is the numerical equality;

Moreover $\mathbf{prop}_s$ is interpreted as the class

Moreover $\mathbf{prop}_s$ is interpreted as the class

$$\{x | prop_s(x)\}$$

Moreover $\mathbf{prop}_s$ is interpreted as the class

$$\{x | prop_s(x)\}$$

where $prop_s(x)$ is defined by using fix point formulas:

Moreover $\mathbf{prop}_s$ is interpreted as the class

$$\{x|prop_s(x)\}$$

where $prop_s(x)$ is defined by using fix point formulas:

$$\mathbf{Set}, t\varepsilon x, t \equiv_x s, t \not\varepsilon x, t \not\equiv_x s$$

Moreover $\mathbf{prop}_s$ is interpreted as the class

$$\{x|prop_s(x)\}$$

where $prop_s(x)$ is defined by using fix point formulas:

$$\mathbf{Set}, t\varepsilon x, t \equiv_x s, t \not\varepsilon x, t \not\equiv_x s$$

internalizations of being sets, membership, equality in sets and their negations.

Moreover $\mathbf{prop}_s$ is interpreted as the class

$$\{x|prop_s(x)\}$$

where $prop_s(x)$ is defined by using fix point formulas:

$$\mathbf{Set}, t\varepsilon x, t \equiv_x s, t \not\varepsilon x, t \not\equiv_x s$$

internalizations of being sets, membership, equality in sets and their negations.

REMARK! Classical logic needed for fix points

REMARK! Classical logic needed for fix points
to represent coding for family of small sets positively!

REMARK! Classical logic needed for fix points
to represent coding for family of small sets positively!

$$\mathbf{Set}(a) \wedge \forall x(x \varepsilon a \rightarrow \mathbf{Set}(b))$$

REMARK! Classical logic needed for fix points
to represent coding for family of small sets positively!

$$\mathbf{Set}(a) \wedge \forall x(x \varepsilon a \rightarrow \mathbf{Set}(b))$$

is classically equivalent to the positive formula:

REMARK! Classical logic needed for fix points
to represent coding for family of small sets positively!

$$\mathbf{Set}(a) \wedge \forall x(x \varepsilon a \rightarrow \mathbf{Set}(b))$$

is classically equivalent to the positive formula:

$$\mathbf{Set}(a) \wedge \forall x(x \not\varepsilon\, a \vee \mathbf{Set}(b))$$

REMARK! Classical logic needed for fix points
to represent coding for family of small sets positively!

$$\mathbf{Set}(a) \wedge \forall x(x \varepsilon a \rightarrow \mathbf{Set}(b))$$

is classically equivalent to the positive formula:

$$\mathbf{Set}(a) \wedge \forall x(x \not\varepsilon\, a \vee \mathbf{Set}(b))$$

We need to define $\not\varepsilon$ and $\not\equiv$ as primitives!

The model does not validate **full AC**

The model does not validate **full AC**
a realizer for

$$(\forall x : A)(\exists y : B)R(x, y)$$

The model does not validate **full AC**
a realizer for

$$(\forall x : A)(\exists y : B)R(x, y)$$

does not give a function $f : A \to B$:

The model does not validate **full AC**
a realizer for

$$(\forall x : A)(\exists y : B)R(x, y)$$

does not give a function $f : A \to B$:
the realizer doesn't need to preserve the equality in B.

The model validates $\mathbf{AC}_{N,A}$

The model validates $\mathbf{AC}_{N,A}$
In N we have numerical equality!

The model validates $\mathbf{AC}_{N,A}$
In N we have numerical equality!
The model validates unique choice $\mathbf{AC}_!$.

The model validates **CT**

The model validates **CT**
this comes from proof-irrelevance!

The model validates **CT**
this comes from proof-irrelevance!
$\Rightarrow$ the model validates $\mathbf{ECT} = \mathbf{CT} + \mathbf{AC}_{N,N}$.

Finally the model validates **EXT** and the ξ-rule

Finally the model validates **EXT** and the ξ-rule
Equality in Π-types is extensional!

Conclusion

We proved the consistency of the Minimalist Foundation with **CT**, $\mathbf{AC}_{N,N}$ and extensionality of functions

Conclusion

We proved the consistency of the Minimalist Foundation with **CT**, $\mathbf{AC}_{N,N}$ and extensionality of functions

$$\Downarrow$$

the realizability model makes explicit how to extract programs from proofs in the Minimalist Foundation.

Future work

1. to study the properties of the resulting model of the extensional level.
2. a realizability model for the intensional level validating **AC** and **CT**.