The non-normal abyss in Kleene’s Computability Theory

Sam Sanders (jww Dag Normann)

Department of Philosophy II, RUB Bochum, Germany

CCC23, Kyoto, Sept. 25, 2023
A logical abyss

In a nutshell, this talk reports on my joint project with Dag Normann (U. of Oslo) on the computability theory of the uncountable.
A logical abyss

In a nutshell, this talk reports on my joint project with Dag Normann (U. of Oslo) on the *computability theory of the uncountable*.

Kleene’s higher-order computability theory based on S1-S9 and its recent lambda calculus formulation due to the authors.
A logical abyss

In a nutshell, this talk reports on my joint project with Dag Normann (U. of Oslo) on the computability theory of the uncountable.

Kleene’s higher-order computability theory based on S1-S9 and its recent lambda calculus formulation due to the authors.

The main message: there are computational operations that are mathematically extremely close but logically extremely different.
A logical abyss

In a nutshell, this talk reports on my joint project with Dag Normann (U. of Oslo) on the computability theory of the uncountable.

Kleene’s higher-order computability theory based on S1-S9 and its recent lambda calculus formulation due to the authors.

The main message: there are computational operations that are mathematically extremely close but logically extremely different.

How different? Computable in Kleene’s quantifier $\exists^2 \approx$ Turing jump) versus computable in Kleene’s $\exists^3 \approx$ SOA) but not in weaker oracles.
A logical abyss

In a nutshell, this talk reports on my joint project with Dag Normann (U. of Oslo) on the computability theory of the uncountable.

Kleene’s higher-order computability theory based on S1-S9 and its recent lambda calculus formulation due to the authors.

The main message: there are computational operations that are mathematically extremely close but logically extremely different.

How different? Computable in Kleene’s quantifier $\exists^2$ ($\approx$ Turing jump) versus computable in Kleene’s $\exists^3$ ($\approx$ SOA) but not in weaker oracles.

Why? The ‘$\exists^2$-side’ deals (exactly) with function classes that have a built-in approximation-device for function values
Turing
Kleene computability theory

Exploring the abyss

Turing

Turing’s ‘machine’ framework (1936): first intuitively convincing notion of computing with real numbers (Entscheidungsproblem).
Turing

Turing’s ‘machine’ framework (1936): first intuitively convincing notion of computing with real numbers (Entscheidungsproblem).

Complexity theory studies computation with restricted resources.
Turing’s ‘machine’ framework (1936): first intuitively convincing notion of computing with real numbers (Entscheidungsproblem).

Complexity theory studies computation with restricted resources. Turing machines may or may not produce an output after finitely many steps: partiality and the Halting problem.
Kleene computability theory

Turing and Kleene

Kleene's S1-S9 are computation schemes that formalise $X$ is computable in $Y$ for objects $X$, $Y$ of finite type (essentially most of ordinary math). S1-S9-computability extends Turing computability; the latter is restricted to $X$, $Y$ being real numbers. S1-S8 merely provide a kind of primitive recursion while S9 hard-codes the recursion theorem in an ad hoc way.
Turing and Kleene

Kleene’s S1-S9 are computation schemes that formalise

\[ X \text{ is computable in } Y \]

for objects \( X, Y \) of finite type (essentially most of ordinary math).
Kleene’s S1-S9 are computation schemes that formalise

\[ X \text{ is computable in } Y \]

for objects \( X, Y \) of finite type (essentially most of ordinary math).

S1-S9-computability extends Turing computability; the latter is restricted to \( X, Y \) being real numbers.
Kleene’s S1-S9 are computation schemes that formalise

\[ X \text{ is computable in } Y \]

for objects \( X, Y \) of finite type (essentially most of ordinary math). S1-S9-computability extends Turing computability; the latter is restricted to \( X, Y \) being real numbers.

S1-S8 merely provide a kind of primitive recursion while S9 hard-codes the recursion theorem in an ad hoc way.
For details, consult:
A lambda calculus capturing S1-S9
A lambda calculus capturing S1-S9

Computation can be based on **fixed point operators** (Moschovakis).
A lambda calculus capturing S1-S9

Computation can be based on fixed point operators (Moschovakis).
For Kleene’s S1-S9, the ‘is less partial than’-order is essential:
A lambda calculus capturing S1-S9

Computation can be based on fixed point operators (Moschovakis).
For Kleene's S1-S9, the 'is less partial than'-order is essential:

\( \sigma x \leq_{\sigma} \sigma y \) means: the graph of \( x \) is included in the graph of \( y \).
A lambda calculus capturing S1-S9

Computation can be based on fixed point operators (Moschovakis). For Kleene's S1-S9, the ‘is less partial than’-order is essential:

\[ x^\sigma \preceq_\sigma y^\sigma \] means: the graph of \( x \) is included in the graph of \( y \).

\[ s^{\sigma \rightarrow \tau} \text{ is monotone if: } x \preceq_\sigma y \text{ implies } s(x) \preceq_\tau s(y) \text{ for all } x^\sigma, y^\sigma. \]
A lambda calculus capturing S1-S9

Computation can be based on fixed point operators (Moschovakis).

For Kleene's S1-S9, the 'is less partial than'-order is essential:

\[ x^\sigma \preceq_\sigma y^\sigma \text{ means: the graph of } x \text{ is included in the graph of } y. \]

\[ s^{\sigma \to \tau} \text{ is monotone if: } x \preceq_\sigma y \text{ implies } s(x) \preceq_\tau s(y) \text{ for all } x^\sigma, y^\sigma. \]

For monotone \( s^{\sigma \to \sigma} \), \( \mu x^\sigma.s(x) \) is the least fixed point of \( s \), i.e.
A lambda calculus capturing S1-S9

Computation can be based on fixed point operators (Moschovakis). For Kleene’s S1-S9, the ‘is less partial than’-order is essential:

\( x^\sigma \preceq_\sigma y^\sigma \) means: the graph of \( x \) is included in the graph of \( y \).

\( s^{\sigma \rightarrow \tau} \) is monotone if: \( x \preceq_\sigma y \) implies \( s(x) \preceq_\tau s(y) \) for all \( x^\sigma, y^\sigma \).

For monotone \( s^{\sigma \rightarrow \sigma} \), \( \mu x^\sigma . s(x) \) is the least fixed point of \( s \), i.e.

\[
 s(\mu x^\sigma . s(x)) = \mu x^\sigma . s(x) \text{ and } s(y) = y \to \mu x^\sigma . s(x) \preceq_\sigma y,
\]
A lambda calculus capturing S1-S9

Computation can be based on fixed point operators (Moschovakis).

For Kleene’s S1-S9, the ‘is less partial than’-order is essential:

\[ x^\sigma \preceq^\sigma y^\sigma \text{ means: the graph of } x \text{ is included in the graph of } y. \]

\( s^{\sigma \rightarrow \tau} \) is monotone if: \( x \preceq^\sigma y \) implies \( s(x) \preceq^\tau s(y) \) for all \( x^\sigma, y^\sigma \).

For monotone \( s^{\sigma \rightarrow \sigma} \), \( \mu x^\sigma . s(x) \) is the least fixed point of \( s \), i.e.

\[ s(\mu x^\sigma . s(x)) = \mu x^\sigma . s(x) \text{ and } s(y) = y \rightarrow \mu x^\sigma . s(x) \preceq^\sigma y, \]

Kleene’s S1-S9 is captured by: S1-S8 (primitive recursion) plus \( \lambda \)-abstraction plus the \( \mu^\sigma \)-operator for all finite types.
A lambda calculus capturing S1-S9

Computation can be based on fixed point operators (Moschovakis).
For Kleene’s S1-S9, the ‘is less partial than’-order is essential:

\[ x^\sigma \preceq^\sigma y^\sigma \text{ means: the graph of } x \text{ is included in the graph of } y. \]

\( s^{\sigma \rightarrow \tau} \) is monotone if: \( x \preceq^\sigma y \) implies \( s(x) \preceq^\tau s(y) \) for all \( x^\sigma, y^\sigma \).

For monotone \( s^{\sigma \rightarrow \sigma} \), \( \mu x^\sigma . s(x) \) is the least fixed point of \( s \), i.e.

\[ s(\mu x^\sigma . s(x)) = \mu x^\sigma . s(x) \text{ and } s(y) = y \rightarrow \mu x^\sigma . s(x) \preceq^\sigma y, \]

Kleene’s S1-S9 is captured by: S1-S8 (primitive recursion) plus \( \lambda \)-abstraction plus the \( \mu^\sigma \)-operator for all finite types.

Many functionals that ‘occur in nature’ are partial but take total arguments; our lambda calculus is designed for the study of those.
A lambda calculus capturing S1-S9

Computation can be based on **fixed point operators** (Moschovakis). For Kleene’s S1-S9, the ‘is less partial than’-order is essential:

\[ x^\sigma \preceq_\sigma y^\sigma \text{ means: the graph of } x \text{ is included in the graph of } y. \]

\( s^{\sigma \to \tau} \) is monotone if: \( x \preceq_\sigma y \) implies \( s(x) \preceq_\tau s(y) \) for all \( x^\sigma, y^\sigma \).

For monotone \( s^{\sigma \to \sigma} \), \( \mu x^\sigma.s(x) \) is the least fixed point of \( s \), i.e.

\[
\begin{align*}
    s(\mu x^\sigma.s(x)) &= \mu x^\sigma.s(x) \quad \text{and} \\
    s(y) &= y \to \mu x^\sigma.s(x) \preceq_\sigma y,
\end{align*}
\]

Kleene’s **S1-S9** is captured by: S1-S8 (primitive recursion) plus \( \lambda \)-abstraction plus the \( \mu^\sigma \)-operator for all finite types.

Many functionals that ‘occur in nature’ are partial but take total arguments; our lambda calculus is designed for the study of those.

Why study Kleene’s computability theory?
Why study Kleene’s computability theory?

Kleene’s S1-S9 provides a/the most general model of computation for objects of finite type.
Why study Kleene’s computability theory?

Kleene’s S1-S9 provides a/the most general model of computation for objects of finite type.

It also has a unique closure property called Gandy selection, essentially a computational version of the Axiom of Choice.
Why study Kleene’s computability theory?

Kleene’s S1-S9 provides a/the most general model of computation for objects of finite type.

It also has a unique closure property called Gandy selection, essentially a computational version of the Axiom of Choice.

\[(\forall x^\sigma)(\exists y^\tau)A(x, y) \rightarrow (\exists F^\sigma \rightarrow^\tau)(\forall x^\sigma)A(x, F(x)).\]

choice function
Why study Kleene’s computability theory?

Kleene’s S1-S9 provides a/the most general model of computation for objects of finite type.

It also has a unique closure property called Gandy selection, essentially a computational version of the Axiom of Choice.

\[
(\forall x^\sigma)(\exists y^\tau)A(x, y) \rightarrow \left(\exists F^{\sigma \rightarrow \tau}\right) (\forall x^\sigma)A(x, F(x)).
\]

Under certain restrictions, Gandy selection (and variations) guarantees:
Why study Kleene’s computability theory?

Kleene’s S1-S9 provides a/the most general model of computation for objects of finite type.

It also has a unique closure property called Gandy selection, essentially a computational version of the Axiom of Choice.

\[(\forall x^\sigma)(\exists y^\tau)A(x, y) \rightarrow (\exists F^{\sigma \rightarrow \tau}) (\forall x^\sigma)A(x, F(x)).\]

choice function

Under certain restrictions, Gandy selection (and variations) guarantees:

\[(\forall \text{computable } x^\sigma)(\exists \text{computable } y^\tau)A(x, y)\] decidable
Why study Kleene’s computability theory?

Kleene’s S1-S9 provides a/the most general model of computation for objects of finite type.

It also has a unique closure property called Gandy selection, essentially a computational version of the Axiom of Choice.

\[
\left( \forall x^\sigma \right) \left( \exists y^\tau \right) A(x, y) \rightarrow \left( \exists F^{\sigma \rightarrow \tau} \right) \left( \forall x^\sigma \right) A(x, F(x)).
\]

Under certain restrictions, Gandy selection (and variations) guarantees:

\[
\left( \forall \text{computable } x^\sigma \left( \exists \text{computable } y^\tau \right) A(x, y) \right) \downarrow \left( \exists \text{computable } F^{\sigma \rightarrow \tau} \left( \forall \text{computable } x^\sigma \right) A(x, F(x)). \right)
\]
Some oracles

The Halting problem/Turing jump has the following counterpart:
Some oracles

The Halting problem/Turing jump has the following counterpart:

$$(\forall f : \mathbb{N} \rightarrow \mathbb{N})(\exists^2(f) = 0 \iff (\exists n \in \mathbb{N})(f(n) = 0)).$$
Some oracles

The Halting problem/Turing jump has the following counterpart:

$$(\forall f : \mathbb{N} \rightarrow \mathbb{N})(\exists^2 f = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)).$$

Kleene’s quantifier $\exists^2$ is discontinuous at $f = 11 \ldots$. 
Some oracles

The Halting problem/Turing jump has the following counterpart:

\[(\forall f : \mathbb{N} \rightarrow \mathbb{N})(\exists^2(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)).\]

Kleene’s quantifier \(\exists^2\) is discontinuous at \(f = 11\ldots\).

Similarly: \(S^2_k\) decides the truth of \(\varphi \in \Pi^1_k\) (Sieg-Feferman).
Some oracles

The Halting problem/Turing jump has the following counterpart:

$$(\forall f : \mathbb{N} \rightarrow \mathbb{N})(\exists^2(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)).$$

Kleene’s quantifier $\exists^2$ is discontinuous at $f = 11\ldots$.

Similarly: $S^2_k$ decides the truth of $\varphi \in \Pi^1_k$ (Sieg-Feferman).

The ‘hyperjump’ functional $S^2_1$ is called the Suslin functional.
Some oracles

The Halting problem/Turing jump has the following counterpart:

$$(\forall f : \mathbb{N} \to \mathbb{N})(\exists^2(f) = 0 \iff (\exists n \in \mathbb{N})(f(n) = 0)).$$

Kleene’s quantifier $\exists^2$ is discontinuous at $f = 11\ldots$

Similarly: $S^2_k$ decides the truth of $\varphi \in \Pi^1_k$ (Sieg-Feferman).

The ‘hyperjump’ functional $S^2_1$ is called the Suslin functional.

Everything we do is computable in Kleene’s quantifier $\exists^3$:

$$(\forall Y : \mathbb{N}^\mathbb{N} \to \mathbb{N})(\exists^3(Y) = 0 \iff (\exists f \in \mathbb{N}^\mathbb{N})(Y(f) = 0)).$$

which yields full second-order arithmetic.
Some oracles

The Halting problem/Turing jump has the following counterpart:

\[(\forall f : \mathbb{N} \to \mathbb{N})(\exists^2(f) = 0 \iff (\exists n \in \mathbb{N})(f(n) = 0)).\]

Kleene’s quantifier \(\exists^2\) is discontinuous at \(f = 11\ldots\).

Similarly: \(S^2_k\) decides the truth of \(\varphi \in \Pi^1_k\) (Sieg-Feferman).

The ‘hyperjump’ functional \(S^2_1\) is called the Suslin functional.

Everything we do is computable in Kleene’s quantifier \(\exists^3\):

\[(\forall Y : \mathbb{N} \to \mathbb{N})(\exists^3(Y) = 0 \iff (\exists f \in \mathbb{N}^\mathbb{N})(Y(f) = 0)).\]

which yields full second-order arithmetic.

Historically, the focus of Kleene’s computability theory was normal functionals (which compute \(\exists^2\) or \(\exists^3\)).
Some oracles

The Halting problem/Turing jump has the following counterpart:

\[(\forall f : \mathbb{N} \to \mathbb{N})(\exists^2(f) = 0 \iff (\exists n \in \mathbb{N})(f(n) = 0)).\]

Kleene's quantifier \(\exists^2\) is discontinuous at \(f = 11\ldots\).

Similarly: \(S_k^2\) decides the truth of \(\varphi \in \Pi^1_k\) (Sieg-Feferman).

The 'hyperjump' functional \(S_1^2\) is called the Suslin functional.

Everything we do is computable in Kleene's quantifier \(\exists^3\):

\[(\forall Y : \mathbb{N}^\mathbb{N} \to \mathbb{N})(\exists^3(Y) = 0 \iff (\exists f \in \mathbb{N}^\mathbb{N})(Y(f) = 0)).\]

which yields full second-order arithmetic.

Historically, the focus of Kleene's computability theory was normal functionals (which compute \(\exists^2\) or \(\exists^3\)). (Hilbert-Bernays, Tait)
The non-normal abyss

Kleene’s quantifiers \( \exists^2 \) and \( \exists^3 \):

\[
(\forall f : \mathbb{N} \to \mathbb{N})(\exists^2 (f) = 0 \iff (\exists n \in \mathbb{N})(f(n) = 0)).
\]

\[
(\forall Y : \mathbb{N}^\mathbb{N} \to \mathbb{N})(\exists^3 (Y) = 0 \iff (\exists f \in \mathbb{N}^\mathbb{N})(Y(f) = 0)).
\]
The non-normal abyss

Kleene’s quantifiers $\exists^2$ and $\exists^3$:

$$(\forall f : \mathbb{N} \rightarrow \mathbb{N})(\exists^2(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)).$$

$$(\forall Y : \mathbb{N}^\mathbb{N} \rightarrow \mathbb{N})(\exists^3(Y) = 0 \leftrightarrow (\exists f \in \mathbb{N}^\mathbb{N})(Y(f) = 0)).$$

Huge abyss between $\exists^2$ and $\exists^3$ in terms of computational strength.
The non-normal abyss

Kleene’s quantifiers $\exists^2$ and $\exists^3$:

$$(\forall f : \mathbb{N} \to \mathbb{N})(\exists^2(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)).$$

$$(\forall Y : \mathbb{N}^\mathbb{N} \to \mathbb{N})(\exists^3(Y) = 0 \leftrightarrow (\exists f \in \mathbb{N}^{\mathbb{N}})(Y(f) = 0)).$$

Huge abyss between $\exists^2$ and $\exists^3$ in terms of computational strength.

This talk:

(a) we identify basic (non-normal) functionals that are computable in $\exists^2$
The non-normal abyss

Kleene’s quantifiers \( \exists^2 \) and \( \exists^3 \):

\[
(\forall f : \mathbb{N} \to \mathbb{N})(\exists^2(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)).
\]

\[
(\forall Y : \mathbb{N}^\mathbb{N} \to \mathbb{N})(\exists^3(Y) = 0 \leftrightarrow (\exists f \in \mathbb{N}^\mathbb{N})(Y(f) = 0)).
\]

Huge abyss between \( \exists^2 \) and \( \exists^3 \) in terms of computational strength.

This talk:

(a) we identify basic (non-normal) functionals that are computable in \( \exists^2 \)

(b) and for which slight variations are computable in \( \exists^3 \) but not computable in any functional \( S_k^2 \) (which decides \( \Pi_k^1 \)-formulas).
The non-normal abyss

Kleene’s quantifiers $\exists^2$ and $\exists^3$:

$$(\forall f : \mathbb{N} \to \mathbb{N})(\exists^2(f) = 0 \leftrightarrow (\exists n \in \mathbb{N})(f(n) = 0)).$$

$$(\forall Y : \mathbb{N}^\mathbb{N} \to \mathbb{N})(\exists^3(Y) = 0 \leftrightarrow (\exists f \in \mathbb{N}^\mathbb{N})(Y(f) = 0)).$$

Huge abyss between $\exists^2$ and $\exists^3$ in terms of computational strength.

This talk:

(a) we identify basic (non-normal) functionals that are computable in $\exists^2$

(b) and for which slight variations are computable in $\exists^3$ but not computable in any functional $S^2_k$ (which decides $\Pi^1_k$-formulas).

Item (a) deals (exactly) with definitions that have a built-in approximation-device for function values.
A regular abyss just beyond the continuous

We always study $f : [0, 1] \to \mathbb{R}$ for well-known function classes.
A regular abyss just beyond the continuous

We always study $f : [0, 1] \rightarrow \mathbb{R}$ for well-known function classes.

$f$ is regulated (aka regular) if the left and right limits $f(x-) \,$ and $\, f(x+)$ exist everywhere.
A regular abyss just beyond the continuous

We always study \( f : [0,1] \to \mathbb{R} \) for well-known function classes.

\( f \) is regulated (aka regular) if the left and right limits \( f(x-) \) and \( f(x+) \) exist everywhere.

\( f \) is cadlag if it is regulated and right-continuous, i.e. \( f(x) = f(x+) \) everywhere.
A regular abyss just beyond the continuous

We always study $f : [0, 1] \rightarrow \mathbb{R}$ for well-known function classes.

$f$ is regulated (aka regular) if the left and right limits $f(x-)$ and $f(x+)$ exist everywhere.

$f$ is cadlag if it is regulated and right-continuous, i.e. $f(x) = f(x+)$ everywhere.

Both have (at most) countably many points of discontinuity and a rich history (PDE, probability, Bourbaki, Scheeffer, . . .).
A regular abyss just beyond the continuous

We always study $f : [0,1] \to \mathbb{R}$ for well-known function classes.

$f$ is regulated (aka regular) if the left and right limits $f(x-) \text{ and } f(x+)$ exist everywhere.

$f$ is cadlag if it is regulated and right-continuous, i.e.
$f(x) = f(x+)$ everywhere.

$\exists^2$ computes $\sup_{x \in [p,q]} f(x)$ for any cadlag $f : [0,1] \to \mathbb{R}$ and
$p, q \in [0,1]$. 
A regular abyss just beyond the continuous

We always study $f : [0, 1] \to \mathbb{R}$ for well-known function classes.

$f$ is regulated (aka regular) if the left and right limits $f(x-) = f(x-)$ and $f(x+) = f(x+)$ exist everywhere.

$f$ is cadlag if it is regulated and right-continuous, i.e., $f(x) = f(x+)$ everywhere.

$\exists^2$ computes $\sup_{x \in [p, q]} f(x)$ for any cadlag $f : [0, 1] \to \mathbb{R}$ and $p, q \in [0, 1]$.

$\exists^3$ computes $\sup_{x \in [p, q]} f(x)$ for any regulated $f : [0, 1] \to \mathbb{R}$ and $p, q \in [0, 1]$. 

Sharp: the functional $S^k_2$, which decides $\Pi^1_k$-formulas, cannot in general compute suprema for regulated functions (holds for any $k$).

NB: right-continuity as in $f(x) = f(x-)$ allows us to approximate $f(x)$ given only $f(q)$ for all $q \in Q \cap [0, 1]$. 

A regular abyss just beyond the continuous

We always study \( f : [0, 1] \to \mathbb{R} \) for well-known function classes.

\( f \) is regulated (aka regular) if the left and right limits \( f(x-) \) and \( f(x+) \) exist everywhere.

\( f \) is cadlag if it is regulated and right-continuous, i.e. \( f(x) = f(x+) \) everywhere.

\( \exists^2 \) computes \( \sup_{x \in [p, q]} f(x) \) for any cadlag \( f : [0, 1] \to \mathbb{R} \) and \( p, q \in [0, 1] \).

\( \exists^3 \) computes \( \sup_{x \in [p, q]} f(x) \) for any regulated \( f : [0, 1] \to \mathbb{R} \) and \( p, q \in [0, 1] \).

**Sharp:** the functional \( S^2_k \), which decides \( \Pi^1_k \)-formulas, cannot in general compute suprema for regulated functions (holds for any \( k \)).
A regular abyss just beyond the continuous

We always study $f : [0, 1] \rightarrow \mathbb{R}$ for well-known function classes.

$f$ is regulated (aka regular) if the left and right limits $f(x-) \text{ and } f(x+)$ exist everywhere.

$f$ is cadlag if it is regulated and right-continuous, i.e. $f(x) = f(x+)$ everywhere.

$\exists^2$ computes $\sup_{x \in [p, q]} f(x)$ for any cadlag $f : [0, 1] \rightarrow \mathbb{R}$ and $p, q \in [0, 1]$.

$\exists^3$ computes $\sup_{x \in [p, q]} f(x)$ for any regulated $f : [0, 1] \rightarrow \mathbb{R}$ and $p, q \in [0, 1]$.

Sharp: the functional $S_k^2$, which decides $\Pi^1_k$-formulas, cannot in general compute suprema for regulated functions (holds for any $k$).

NB: right-continuity as in $f(x) = f(x-)$ allows us to approximate $f(x)$ given only $f(q)$ for all $q \in \mathbb{Q} \cap [0, 1]$. 
An abyss among the Baire functions

Baire 1 means: pointwise limit of sequence of continuous functions.
An abyss among the Baire functions

Baire 1 means: pointwise limit of sequence of continuous functions.

Baire 2 means: pointwise limit of sequence of Baire 1 functions.
An abyss among the Baire functions

Baire 1 means: pointwise limit of sequence of continuous functions.
Baire 2 means: pointwise limit of sequence of Baire 1 functions.
Effectively Baire 2 means: iterated limit of double sequence of continuous functions ($\approx$ second-order codes for Baire 2).
An abyss among the Baire functions

Baire 1 means: pointwise limit of sequence of continuous functions.  
Baire 2 means: pointwise limit of sequence of Baire 1 functions.  
Effectively Baire 2 means: iterated limit of double sequence of continuous functions (≈ second-order codes for Baire 2).  
Baire (1905) notes that Baire 2 functions can be represented as iterated limits.
An abyss among the Baire functions

**Baire 1** means: pointwise limit of sequence of continuous functions.

**Baire 2** means: pointwise limit of sequence of Baire 1 functions.

**Effectively Baire 2** means: iterated limit of double sequence of continuous functions (≈ second-order codes for Baire 2).

\[ \exists^2 \text{ computes } \sup_{x \in [p, q]} f(x) \text{ given Baire } 1 \ f : [0, 1] \to [0, 1], \]

\[ p, q \in [0, 1], \text{ and the associated sequence of continuous functions.} \]
An abyss among the Baire functions

Baire 1 means: pointwise limit of sequence of continuous functions.
Baire 2 means: pointwise limit of sequence of Baire 1 functions.
Effectively Baire 2 means: iterated limit of double sequence of continuous functions ($\approx$ second-order codes for Baire 2).

$\exists^2$ computes $\sup_{x \in [p, q]} f(x)$ given Baire 1 $f : [0, 1] \to [0, 1]$, $p, q \in [0, 1]$, and the associated sequence of continuous functions.

$\exists^3$ computes $\sup_{x \in [p, q]} f(x)$ given Baire 2 $f : [0, 1] \to [0, 1]$, $p, q \in [0, 1]$, and the associated sequence of Baire 1 functions.
An abyss among the Baire functions

Baire 1 means: pointwise limit of sequence of continuous functions.
Baire 2 means: pointwise limit of sequence of Baire 1 functions.
Effectively Baire 2 means: iterated limit of double sequence of continuous functions ($\approx$ second-order codes for Baire 2).

$\exists^2$ computes $\sup_{x \in [p,q]} f(x)$ given Baire 1 $f : [0,1] \to [0,1]$, $p, q \in [0,1]$, and the associated sequence of continuous functions.
$\exists^3$ computes $\sup_{x \in [p,q]} f(x)$ given Baire 2 $f : [0,1] \to [0,1]$, $p, q \in [0,1]$, and the associated sequence of Baire 1 functions.

Sharp: the functional $S^2_k$, which decides $\Pi^1_k$-formulas, cannot in general compute suprema for Baire 2 functions (holds for any $k$).
An abyss among the Baire functions

Baire 1 means: pointwise limit of sequence of continuous functions.  
Baire 2 means: pointwise limit of sequence of Baire 1 functions.  
Effectively Baire 2 means: iterated limit of double sequence of continuous functions ($\approx$ second-order codes for Baire 2).

$\exists^2$ computes $\sup_{x \in [p,q]} f(x)$ given Baire 1 $f : [0, 1] \to [0, 1]$, $p, q \in [0, 1]$, and the associated sequence of continuous functions.  
$\exists^3$ computes $\sup_{x \in [p,q]} f(x)$ given Baire 2 $f : [0, 1] \to [0, 1]$, $p, q \in [0, 1]$, and the associated sequence of Baire 1 functions.  

Sharp: the functional $S^2_k$, which decides $\Pi^1_k$-formulas, cannot in general compute suprema for Baire 2 functions (holds for any $k$).

Borderline: the Suslin functional $S^2_1$ computes $\sup_{x \in [p,q]} f(x)$ given effectively Baire 2 $f : [0, 1] \to [0, 1]$, $p, q \in [0, 1]$, and the associated double sequence of continuous functions.
Out there: quasi-continuity and around

- **Quasi-continuity**: For all $\epsilon > 0$, $N \in \mathbb{N}$, and $x \in [0,1]$, there is a $(a, b) \subset B(x, 1/2N)$ with $(\forall y \in (a, b))(|f(x) - f(y)| < \epsilon)$.

- **Cliquish**: For all $\epsilon > 0$, $N \in \mathbb{N}$, and $x \in [0,1]$, there is $(a, b) \subset B(x, 1/2N)$ such that $(\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon)$.

Some properties:
- Studied by Baire, Volterra, Hankel, ... starting ca 1870.
- Cliquish = continuity points are dense = pointwise discontinuous.
- There are $2^{\mathfrak{c}}$ non-measurable quasi-cont. functions and $2^{\mathfrak{c}}$ non-Borel measurable quasi-cont. functions.
- Closely related: every cliquish function is the sum of two quasi-continuous functions (on the reals).
Out there: quasi-continuity and around

**f is quasi-continuous** if for all $\epsilon > 0$, $N \in \mathbb{N}$, $x \in [0, 1]$, there is $(a, b) \subset B(x, \frac{1}{2^N})$ with $(\forall y \in (a, b))(|f(x) - f(y)| < \epsilon)$. 
Out there: quasi-continuity and around

\( f \) is quasi-continuous if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1] \), there is \((a, b) \subset B(x, \frac{1}{2N})\) with \((\forall y \in (a, b))(|f(x) - f(y)| < \epsilon)\).

\( f \) is cliquish if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1] \), there is \((a, b) \subset B(x, \frac{1}{2N})\) such that \((\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon)\).
Out there: quasi-continuity and around

\( f \) is quasi-continuous if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1], \) there is \((a, b) \subset B(x, \frac{1}{2^N})\) with \((\forall y \in (a, b))(|f(x) - f(y)| < \epsilon)\).

\( f \) is cliquish if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1], \) there is \((a, b) \subset B(x, \frac{1}{2^N})\) such that \((\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon)\).

Some properties:

- Studied by Baire, Volterra, Hankel, ... starting ca 1870.
Out there: quasi-continuity and around

\( f \) is quasi-continuous if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1] \), there is \((a, b) \subset B(x, \frac{1}{2^N})\) with \((\forall y \in (a, b))(|f(x) - f(y)| < \epsilon)\).

\( f \) is cliquish if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1] \), there is \((a, b) \subset B(x, \frac{1}{2^N})\) such that \((\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon)\).

Some properties:

- Studied by Baire, Volterra, Hankel, ... starting ca 1870.

- Cliquish = continuity points are dense = pointwise discontinuous.
Out there: quasi-continuity and around

\( f \) is quasi-continuous if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1], \) there is \((a, b) \subset B(x, \frac{1}{2^N})\) with \((\forall y \in (a, b))(|f(x) - f(y)| < \epsilon).\)

\( f \) is cliquish if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1], \) there is \((a, b) \subset B(x, \frac{1}{2^N})\) such that \((\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon).\)

Some properties:

- Studied by Baire, Volterra, Hankel, ... starting ca 1870.
- Cliquish = continuity points are dense = pointwise discontinuous.
- There are \(2^{|\mathbb{R}|}\) non-measurable quasi-cont. functions and \(2^{|\mathbb{R}|}\) non-Borel measurable quasi-cont. functions.
Out there: quasi-continuity and around

$f$ is quasi-continuous if for all $\epsilon > 0$, $N \in \mathbb{N}$, $x \in [0, 1]$, there is $(a, b) \subset B(x, \frac{1}{2^N})$ with $(\forall y \in (a, b))(|f(x) - f(y)| < \epsilon)$.

$f$ is cliquish if for all $\epsilon > 0$, $N \in \mathbb{N}$, $x \in [0, 1]$, there is $(a, b) \subset B(x, \frac{1}{2^N})$ such that $(\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon)$.

Some properties:

- Studied by Baire, Volterra, Hankel, ... starting ca 1870.
- Cliquish = continuity points are dense = pointwise discontinuous.
- There are $2^{|\mathbb{R}|}$ non-measurable quasi-cont. functions and $2^{|\mathbb{R}|}$ non-Borel measurable quasi-cont. functions.
- Closely related: every cliquish function is the sum of two quasi-continuous functions (on the reals).
Quasi-continuity and around

\( f \) is quasi-continuous if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0,1] \), there is 
\( (a, b) \subseteq B(x, \frac{1}{2^N}) \) with \( (\forall y \in (a, b))(|f(x) - f(y)| < \epsilon) \).

\( f \) is cliquish if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0,1] \), there is 
\( (a, b) \subseteq B(x, \frac{1}{2^N}) \) such that \( (\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon) \).

Closely related: every cliquish function is the sum of two 
quasi-continuous functions (on the reals).
Quasi-continuity and around

\( f \) is quasi-continuous if for all \( \varepsilon > 0, N \in \mathbb{N}, x \in [0, 1], \) there is \((a, b) \subset B(x, \frac{1}{2^N})\) with \((\forall y \in (a, b))(|f(x) - f(y)| < \varepsilon)\).

\( f \) is cliquish if for all \( \varepsilon > 0, N \in \mathbb{N}, x \in [0, 1], \) there is \((a, b) \subset B(x, \frac{1}{2^N})\) such that \((\forall y, z \in (a, b))(|f(z) - f(y)| < \varepsilon)\).

Closely related: every cliquish function is the sum of two quasi-continuous functions (on the reals).

\( \exists^2 \) computes \( \sup_{x \in [p, q]} f(x) \) given quasi-continuous \( f : [0, 1] \rightarrow [0, 1], \) \( p, q \in [0, 1] \).
Quasi-continuity and around

$f$ is quasi-continuous if for all $\epsilon > 0$, $N \in \mathbb{N}$, $x \in [0, 1]$, there is $(a, b) \subset B(x, \frac{1}{2^N})$ with $(\forall y \in (a, b))(|f(x) - f(y)| < \epsilon)$.

$f$ is cliquish if for all $\epsilon > 0$, $N \in \mathbb{N}$, $x \in [0, 1]$, there is $(a, b) \subset B(x, \frac{1}{2^N})$ such that $(\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon)$.

Closely related: every cliquish function is the sum of two quasi-continuous functions (on the reals).

$\exists^2$ computes $\sup_{x \in [p, q]} f(x)$ given quasi-continuous $f : [0, 1] \to [0, 1]$, $p, q \in [0, 1]$.

$\exists^3$ computes $\sup_{x \in [p, q]} f(x)$ given cliquish $f : [0, 1] \to [0, 1]$, $p, q \in [0, 1]$. 
Quasi-continuity and around

$f$ is quasi-continuous if for all $\epsilon > 0, N \in \mathbb{N}, x \in [0, 1]$, there is $(a, b) \subset B(x, \frac{1}{2^N})$ with $(\forall y \in (a, b))(|f(x) - f(y)| < \epsilon)$.

$f$ is cliquish if for all $\epsilon > 0, N \in \mathbb{N}, x \in [0, 1]$, there is $(a, b) \subset B(x, \frac{1}{2^N})$ such that $(\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon)$.

Closely related: every cliquish function is the sum of two quasi-continuous functions (on the reals).

$\exists^2$ computes $\sup_{x \in [p, q]} f(x)$ given quasi-continuous $f : [0, 1] \to [0, 1], p, q \in [0, 1]$.

$\exists^3$ computes $\sup_{x \in [p, q]} f(x)$ given cliquish $f : [0, 1] \to [0, 1], p, q \in [0, 1]$.

Sharp: the functional $S_k^2$, which decides $\Pi_k^1$-formulas, cannot in general compute suprema for cliquish functions (holds for any $k$).
Quasi-continuity and around

\( f \) is quasi-continuous if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1] \), there is \((a, b) \subset B(x, \frac{1}{2^N})\) with \((\forall y \in (a, b))(|f(x) - f(y)| < \epsilon)\).

\( f \) is cliquish if for all \( \epsilon > 0, N \in \mathbb{N}, x \in [0, 1] \), there is \((a, b) \subset B(x, \frac{1}{2^N})\) such that \((\forall y, z \in (a, b))(|f(z) - f(y)| < \epsilon)\).

Closely related: every cliquish function is the sum of two quasi-continuous functions (on the reals).

\( \exists^2 \) computes \( \sup_{x \in [p, q]} f(x) \) given quasi-continuous \( f : [0, 1] \rightarrow [0, 1], p, q \in [0, 1] \).

\( \exists^3 \) computes \( \sup_{x \in [p, q]} f(x) \) given cliquish \( f : [0, 1] \rightarrow [0, 1], p, q \in [0, 1] \).

Sharp: the functional \( S^2_k \), which decides \( \Pi^1_k \)-formulas, cannot in general compute suprema for cliquish functions (holds for any \( k \)).

Note that quasi-continuity allows us to approximate \( f(x) \) given only \( f(q) \) for all \( q \in \mathbb{Q} \cap [0, 1] \).
Decompositions of continuity
Decompositions of continuity

continuity $\leftrightarrow$ [weak continuity notion A + weak continuity notion B].
Decompositions of continuity

continuity $\leftrightarrow$ [\textit{weak} continuity notion A + \textit{weak} continuity notion B].

Example: continuity $\leftrightarrow$ [quasi-continuity + graph continuity].
Decompositions of continuity

continuity $\leftrightarrow$ [weak continuity notion A + weak continuity notion B].

Example: continuity $\leftrightarrow$ [quasi-continuity + graph continuity].

graph continuity: the closure of the graph contains the graph of a continuous function.
Decompositions of continuity

continuity $\leftrightarrow$ [weak continuity notion A + weak continuity notion B].

Example: continuity $\leftrightarrow$ [quasi-continuity + graph continuity].

graph continuity: the closure of the graph contains the graph of a continuous function.

As before, $\exists^2$ computes $\sup_{x \in [p, q]} f(x)$ given quasi-continuous $f : [0, 1] \to [0, 1]$, $p, q \in [0, 1]$. 
Decompositions of continuity

continuity $\iff$ [weak continuity notion A + weak continuity notion B].

**Example:** continuity $\iff$ [quasi-continuity + graph continuity].

**Graph continuity:** the closure of the graph contains the graph of a continuous function.

As before, $\exists^2$ computes $\sup_{x \in [p, q]} f(x)$ given quasi-continuous $f : [0, 1] \to [0, 1], p, q \in [0, 1]$.

The supremum functional $\lambda f. \sup_{x \in [0, 1]} f(x)$ for graph continuous $f : [0, 1] \to \mathbb{R}$ is computable in $\exists^3$, but not in weaker oracles.
Decompositions of continuity

continuity $\iff$ [weak continuity notion A + weak continuity notion B].

Example: continuity $\iff$ [quasi-continuity + graph continuity].

graph continuity: the closure of the graph contains the graph of a continuous function.

As before, $\exists^2$ computes $\sup_{x \in [p, q]} f(x)$ given quasi-continuous $f : [0, 1] \to [0, 1]$, $p, q \in [0, 1]$.

The supremum functional $\lambda f . \sup_{x \in [0, 1]} f(x)$ for graph continuous $f : [0, 1] \to \mathbb{R}$ is computable in $\exists^3$, but not in weaker oracles.

Many such decompositions exist, with numerous similar examples.
Similar theorems

The supremum principle is not special; the same abyss is observed for other basic properties.
Similar theorems

The supremum principle is not special; the same abyss is observed for other basic properties.

- **Coverings lemmas** (Cousin, Vitali, Heine-Borel, ...) for coverings $\bigcup_{x \in [0,1]} B(x, \psi(x))$ and $\psi : [0, 1] \to \mathbb{R}^+$. 

... Finally, how do we prove our negative results?
Similar theorems

The supremum principle is not special; the same abyss is observed for other basic properties.

- **Coverings lemmas** (Cousin, Vitali, Heine-Borel, ... ) for coverings \( \bigcup_{x \in [0,1]} B(x, \Psi(x)) \) and \( \Psi : [0,1] \to \mathbb{R}^+ \).
- Finding an RM-code for the set of continuity points \( C_f \).
Similar theorems

The supremum principle is not special; the same abyss is observed for other basic properties.

- **Coverings lemmas** (Cousin, Vitali, Heine-Borel, ...) for coverings $\bigcup_{x \in [0,1]} B(x, \Psi(x))$ and $\Psi : [0,1] \to \mathbb{R}^+$. 
- Finding an RM-code for the set of continuity points $C_f$. 
- Deciding whether $f$ is **continuous** at a given real $x \in [0,1]$. 

Similar theorems

The supremum principle is not special; the same abyss is observed for other basic properties.

- **Coverings lemmas** (Cousin, Vitali, Heine-Borel, ...) for coverings $\bigcup_{x \in [0,1]} B(x, \Psi(x))$ and $\Psi : [0,1] \to \mathbb{R}^+$.  
- Finding an RM-code for the set of continuity points $C_f$.  
- Deciding whether $f$ is **continuous** at a given real $x \in [0,1]$.  
- Jordan decomposition theorem.
Similar theorems

The supremum principle is not special; the same abyss is observed for other basic properties.

- **Coverings lemmas** (Cousin, Vitali, Heine-Borel, ... ) for coverings $\bigcup_{x \in [0,1]} B(x, \Psi(x))$ and $\Psi : [0,1] \to \mathbb{R}^+$.  
- Finding an RM-code for the set of continuity points $C_f$.  
- Deciding whether $f$ is **continuous** at a given real $x \in [0,1]$.  
- Jordan decomposition theorem.  
- For a Riemann integrable $f : [0,1] \to [0,1]$ with $\int_0^1 f(x)dx = 0$, find $x \in [0,1]$ such that $f(x) = 0$.  
- ...
Similar theorems

The supremum principle is not special; the same abyss is observed for other basic properties.

- **Coverings lemmas** (Cousin, Vitali, Heine-Borel, ...) for coverings $\bigcup_{x \in [0,1]} B(x, \psi(x))$ and $\psi : [0, 1] \to \mathbb{R}^+$.  
  Finding an RM-code for the set of continuity points $C_f$. 
  Deciding whether $f$ is **continuous** at a given real $x \in [0, 1]$. 
  Jordan decomposition theorem. 
  For a Riemann integrable $f : [0, 1] \to [0, 1]$ with $\int_0^1 f(x) dx = 0$, find $x \in [0, 1]$ such that $f(x) = 0$. 
  ... 

Finally, how do we prove our **negative** results?
The uncountability of $\mathbb{R}$

Cantor’s first set theory paper (1874): uncountability of $\mathbb{R}$. 

∃ can compute a CR, but $S^2_k$ cannot compute a CR (hold for any $k$).
The uncountability of \( \mathbb{R} \)

Cantor’s first set theory paper (1874): uncountability of \( \mathbb{R} \).

Cantor realisers (CR) perform the following associated operation:
The uncountability of $\mathbb{R}$

Cantor’s first set theory paper (1874): uncountability of $\mathbb{R}$.

Cantor realisers (CR) perform the following associated operation:

on input a countable set $X \subset \mathbb{R}$, output an element $y \in \mathbb{R} \setminus X$. 
The uncountability of $\mathbb{R}$

Cantor’s first set theory paper (1874): uncountability of $\mathbb{R}$.

Cantor realisers (CR) perform the following associated operation:

on input a countable set $X \subset \mathbb{R}$, output an element $y \in \mathbb{R} \setminus X$.

A countable set $X \subset \mathbb{R}$ is a pair $(X, Y)$ where $Y : \mathbb{R} \rightarrow \mathbb{N}$ is assumed to be injective (or even bijective) on $X$. 
The uncountability of $\mathbb{R}$

Cantor’s first set theory paper (1874): uncountability of $\mathbb{R}$.

Cantor realisers (CR) perform the following associated operation:

on input a countable set $X \subset \mathbb{R}$, output an element $y \in \mathbb{R} \setminus X$.

A countable set $X \subset \mathbb{R}$ is a pair $(X, Y)$ where $Y : \mathbb{R} \to \mathbb{N}$ is assumed to be injective (or even bijective) on $X$.

$\exists^3$ can compute a CR, but $S^2_k$ cannot compute a CR (hold for any $k$).
Cantor realisers

Cantor realisers (CR) perform the following operation:

on input a countable set $A \subseteq \mathbb{R}$, output an element $y \in \mathbb{R} \setminus X$.

A countable set $X \subseteq \mathbb{R}$ is a pair $(X, Y)$ where $Y : \mathbb{R} \to \mathbb{N}$ is injective (or even bijective) on $X$.

$\exists^3$ can compute a CR, but no $S^2_k$ can compute a CR.
Cantor realisers

Cantor realisers (CR) perform the following operation:

on input a countable set $A \subset \mathbb{R}$, output an element $y \in \mathbb{R} \setminus X$.

A countable set $X \subset \mathbb{R}$ is a pair $(X, Y)$ where $Y : \mathbb{R} \to \mathbb{N}$ is injective (or even bijective) on $X$.

$\exists^3$ can compute a CR, but no $S^2_k$ can compute a CR.

Our negative results are obtained by computing a CR from the functionals at hand.
Cantor realisers

Cantor realisers (CR) perform the following operation:

on input a countable set $A \subset \mathbb{R}$, output an element $y \in \mathbb{R} \setminus X$.

A countable set $X \subset \mathbb{R}$ is a pair $(X, Y)$ where $Y : \mathbb{R} \to \mathbb{N}$ is injective (or even bijective) on $X$.

$\exists^3$ can compute a CR, but no $S^2_k$ can compute a CR.

Our negative results are obtained by computing a CR from the functionals at hand using:

$$f(x) := \begin{cases} \frac{1}{2^{Y(x)+1}} & \text{if } x \in X \\ 0 & \text{otherwise} \end{cases}$$

which is $BV$, semi-continuous, cliquish, . . . and is found in the literature.
The abyss and its origin

The abyss:

(a) there are basic (non-normal) functionals that are computable in $\exists^2$.

(b) and for which slight variations are computable in $\exists^3$, but not computable in $S_k^2$. 

Mathematically close (or equivalent) notions can land on either side of the abyss!
The abyss and its origin

The abyss:

(a) there are **basic** (non-normal) functionals that are computable in $\exists^2$.

(b) and for which **slight** variations are computable in $\exists^3$, but **not** computable in $S^2_k$.

Its origin: item (a) deals **exactly** with definitions that have a built-in **approximation-device** for function values.
The abyss and its origin

The abyss:

(a) there are basic (non-normal) functionals that are computable in $\exists^2$.

(b) and for which slight variations are computable in $\exists^3$, but not computable in $S^2_k$.

Its origin: item (a) deals exactly with definitions that have a built-in approximation-device for function values.

- Like item (a): Baire 1, effectively Baire $n$, normalised bounded variation, regulated such that $f(x) = \frac{f(x-)+f(x+)}{2}$ everywhere.
The abyss and its origin

The abyss:

(a) there are basic (non-normal) functionals that are computable in $\exists^2$.

(b) and for which slight variations are computable in $\exists^3$, but not computable in $S^2_k$.

Its origin: item (a) deals exactly with definitions that have a built-in approximation-device for function values.

- Like item (a): Baire 1, effectively Baire $n$, normalised bounded variation, regulated such that $f(x) = \frac{f(x^-)+f(x^+)}{2}$ everywhere.
- Like item (b): simple continuity, semi-continuity, fragmented, bounded variation, Baire 1*, $F_\sigma$-measurable.
The abyss and its origin

The abyss:

(a) there are basic (non-normal) functionals that are computable in $\exists^2$.

(b) and for which slight variations are computable in $\exists^3$, but not computable in $S_k^2$.

Its origin: item (a) deals exactly with definitions that have a built-in approximation-device for function values.

- Like item (a): Baire 1, effectively Baire $n$, normalised bounded variation, regulated such that $f(x) = \frac{f(x-)+f(x+)}{2}$ everywhere.

- Like item (b): simple continuity, semi-continuity, fragmented, bounded variation, Baire $1^*$, $F_\sigma$-measurable.

Mathematically close (or equivalent) notions can land on either side of the abyss!
Thanks!
Questions?

Funded by the Klaus Tschira Foundation, German DFG, and RUB Bochum.