Computability in C[0,1]

Alexander Melnikov, Victoria University of Wellington

October 28, 2022

In my talk:

- What is $C[0, 1]$ and why do we care?
- Automatic continuous functions (unexpected results!)
- An effective characterization of $C[0, 1]$
- The effective universality of $C[0, 1]$
- Computable Banach–Stone duality

Part 1: A brief introduction

$C[0,1]$ stands for the Banach space of continuous functions

$$f : [0,1] \to \mathbb{R}.$$

The norm is $||f|| = \sup_{x \in [0,1]} f(x)$. The operations are pointwise.

1. In his book, Banach proved that $C[0,1]$ is universal among separable Banach spaces.
2. The space is one of the ‘standard’ and ‘classical’ spaces
3. $C[0,1]$ is well-studied
4. Its generalisations, such as $C[K]$, are also well-studied
5. Traditionally, $C[0,1]$ is important in logic

Logical aspects of $C[0,1]$:

- (Cherlin, 1980) The first-order theory of the *ring* $C[0,1]$ is not decidable
- (H. Friedman and Seress 1989,1990) Decidability in elementary analysis I, II
- (Kechris and Woodin, 1986) Ranks of differentiable functions
- $C[0,1]$ is a computable Banach space
- Every (Kleene) computable function on $[0,1]$ is continuous

Much of the early computable analysis was restricted to $C[0,1]$.

Part 2: Regular vs transducer functions

Khoussainov once told me that people still struggle to describe

automatic

$$f : [0,1] \to \mathbb{R}.$$

But it also seems there are not many such functions.

What is an automatic function $f : [0,1] \to \mathbb{R}$?

Chaudhuri, Sankaranarayanan, and Vardi (LICS 2013):

Definition 1

A function $f\colon [0,1] \to \mathbb{R}$ is regular if there exists a Büchi automaton on two tapes which accepts the graph of f.

Given representations of x and y, we have $f(x) = y$ if and only if the automaton visits the accepting state infinitely often.

(CSV 2013) Can we characterize continuous regular functions?

Every continuous regular function is actually computable.

Chaudhuri, Sankaranarayanan, and Vardi (LICS 2013):

Definition 1

A function $f \colon [0,1] \to \mathbb{R}$ is regular if there exists a Büchi automaton on two tapes which accepts the graph of f.

Given representations of x and y, we have $f(x) = y$ if and only if the automaton visits the accepting state infinitely often.

(CSV 2013) Can we characterize continuous regular functions?

Every continuous regular function is actually computable.

Chaudhuri, Sankaranarayanan, and Vardi (LICS 2013):

Definition 1

A function $f\colon [0,1] \to \mathbb{R}$ is regular if there exists a Büchi automaton on two tapes which accepts the graph of f.

Given representations of x and y, we have $f(x) = y$ if and only if the automaton visits the accepting state infinitely often.

(CSV 2013) Can we characterize continuous regular functions?

Every continuous regular function is actually computable.

In the 1990-s, Lesovik and then Konecny investigated:

Definition 2

A function $f : [0, 1] \to \mathbb{R}$ is transducer if there is a finite transducer atomaton that on input a (say, binary) representation ξ outputs a representation $f(\xi)$.

So basically we read ξ from left to right and output the next bit of $f(\xi)$ based on the state of the automaton.

Every such function is computable (thus, continuous).

Can we describe (classify, characterize) transducer functions?

In the 1990-s, Lesovik and then Konecny investigated:

Definition 2

A function $f : [0, 1] \to \mathbb{R}$ is transducer if there is a finite transducer atomaton that on input a (say, binary) representation ξ outputs a representation $f(\xi)$.

So basically we read ξ from left to right and output the next bit of $f(\xi)$ based on the state of the automaton.

Every such function is computable (thus, continuous).

Can we describe (classify, characterize) transducer functions?

Franklin, Holzl, M., Ng, and Turetsky:

Definition 1 (transducer) $\iff$ Definition 2 (regular)

It is known a similar equivalence fails for algebraic structures.

Franklin, Holzl, M., Ng, and Turetsky:

$$\text{Definition 1 (transducer)} \iff \text{Definition 2 (regular)}$$

It is known a similar equivalence fails for algebraic structures.

Theorem (FHMNT)

Suppose $f\colon [0,1] \to [0,1]$. The following are equivalent with respect to the standard binary representation:

1. f is continuous regular.
2. f can be computed by a (nondeterministic) transducer.

If we use *signed binary representation*, then f can be computed by a deterministic transducer.

The proof of $(1) \to (2)$ is tedious and not trivial.

The result surely can be extended to other bases and likely domains other than [0,1].

Theorem (FHMNT)

Suppose $f\colon [0,1] \to [0,1]$. The following are equivalent with respect to the standard binary representation:

1. f is continuous regular.
2. f can be computed by a (nondeterministic) transducer.

If we use *signed binary representation*, then f can be computed by a deterministic transducer.

The proof of (1) $\to$ (2) is tedious and not trivial.

The result surely can be extended to other bases and likely domains other than [0,1].

Theorem (FHMNT)

Suppose $f\colon [0,1] \to [0,1]$. The following are equivalent with respect to the standard binary representation:

1. f is continuous regular.
2. f can be computed by a (nondeterministic) transducer.

If we use *signed binary representation*, then f can be computed by a deterministic transducer.

The proof of (1) $\to$ (2) is tedious and not trivial.

The result surely can be extended to other bases and likely domains other than [0,1].

Theorem (FHMNT)

Suppose $f\colon [0,1] \to [0,1]$. The following are equivalent with respect to the standard binary representation:

1. f is continuous regular.
2. f can be computed by a (nondeterministic) transducer.

If we use *signed binary representation*, then f can be computed by a deterministic transducer.

The proof of (1) $\to$ (2) is tedious and not trivial.

The result surely can be extended to other bases and likely domains other than [0,1].

What about classification? We work in (signed) binary.

CSV 2013, Gorman et al. 2020. Lisovik et al 1989, 1998:

$f \in C[0,1]$ is regular (transducer) then f:

(1) maps rationals to rationals in linear time

(2) Lipschitz

(3) linear outside of a measure zero nowhere dense set

Definition

We say that $f \in C[0,1]$ is awesome if it satisfies (1), (2), (3).

What about classification? We work in (signed) binary.

CSV 2013, Gorman et al. 2020. Lisovik et al 1989, 1998:

$f \in C[0,1]$ is regular (transducer) then f:

(1) maps rationals to rationals in linear time

(2) Lipschitz

(3) linear outside of a measure zero nowhere dense set

Definition

We say that $f \in C[0,1]$ is awesome if it satisfies (1), (2), (3).

It is Σ^0_2 to say that a computable f is regular.

Theorem (FHMNT)

Checking whether an awesome f is transducer (equivalently, regular) is a Σ^0_2-complete problem.

Nothing that we know so far helps to classify such functions...

Sorry...

It is Σ_2^0 to say that a computable f is regular.

Theorem (FHMNT)

Checking whether an awesome f is transducer (equivalently, regular) is a Σ_2^0-complete problem.

Nothing that we know so far helps to classify such functions...

Sorry...

Part 3: The global characterization of $C[0,1]$

Suppose we are given some separable Banach space B.

Then B is a Hilbert space iff it satisfies the parallelogram low:

$$||x - y||^2 + ||x + y||^2 = 2||x||^2 + 2||y||^2,$$

for all $x, y \in B$. This is a closed property (Π^0_1).

Lebesgue spaces also admit an arithmetical (Π^0_3) characterization (Brown,McNicholl, M. 2020).

[McNicholl] How hard is it to decide whether $B \cong C[0, 1]$?

Suppose we are given some separable Banach space B.

Then B is a Hilbert space iff it satisfies the parallelogram low:

$$||x - y||^2 + ||x + y||^2 = 2||x||^2 + 2||y||^2,$$

for all $x, y \in B$. This is a closed property (Π^0_1).

Lebesgue spaces also admit an arithmetical (Π^0_3) characterization (Brown,McNicholl, M. 2020).

[McNicholl] How hard is it to decide whether $B \cong C[0, 1]$?

Recall $C[0,1]$ is *not* computably categorical (M.Ng 2012).

Also, it is a universal space with undecidable theory.

Theorem (FHMNT)

$C[0,1]$ admits an arithmetical characterization.

Essentially, $B \cong C[0,1]$ iff it has a teeth-basis, and this can be expressed as an arithmetical property in the given presentation.

The proof uses computable presentations of Banach spaces (will be defined soon).

Recall $C[0,1]$ is *not* computably categorical (M.Ng 2012).

Also, it is a universal space with undecidable theory.

Theorem (FHMNT)

$C[0,1]$ admits an arithmetical characterization.

Essentially, $B \cong C[0,1]$ iff it has a teeth-basis, and this can be expressed as an arithmetical property in the given presentation.

The proof uses computable presentations of Banach spaces (will be defined soon).

Recall $C[0,1]$ is *not* computably categorical (M.Ng 2012).

Also, it is a universal space with undecidable theory.

Theorem (FHMNT)

$C[0,1]$ admits an arithmetical characterization.

Essentially, $B \cong C[0,1]$ iff it has a teeth-basis, and this can be expressed as an arithmetical property in the given presentation.

The proof uses computable presentations of Banach spaces (will be defined soon).

Part 4: Effective universality of C[0,1]

1. Banach used Hahn-Banach theorem to show $C[0,1]$ is universal among separable spaces.
2. In the 1940s, Sierpinski asked if there is a 'more effective' proof of this fact.
3. Sierpinski suggested a proof that he claimed was 'effective', but the proof used $0'$ in disguise.

1. Banach used Hahn-Banach theorem to show $C[0,1]$ is universal among separable spaces.

2. In the 1940s, Sierpinski asked if there is a ’more effective’ proof of this fact.

3. Sierpinski suggested a proof that he claimed was ‘effective’, but the proof used $0'$ in disguise.

1. Banach used Hahn-Banach theorem to show $C[0, 1]$ is universal among separable spaces.

2. In the 1940s, Sierpinski asked if there is a 'more effective' proof of this fact.

3. Sierpinski suggested a proof that he claimed was 'effective', but the proof used $0'$ in disguise.

We say that a separable Banach space B is computable if there is a dense sequence (x_i) such that:

- $||x_i - x_j||$ is a computable real uniformly in i, j

- the Banach space operations are computable functionals

Primitive recursive spaces are defined similarly.

Theorem (BBBBDKKMN 2022)

Every primitive recursive Polish space can be primitively recursively isometrically embedded into $C[0,1]$ (represented by piecewise linear functions).

The proof gives a new explicit embedding.

Our proof is **PR-uniform** and implies computable isometric universality.

Theorem (BBBBDKKMN 2022)

Every primitive recursive Polish space can be primitively recursively isometrically embedded into $C[0,1]$ (represented by piecewise linear functions).

The proof gives a new explicit embedding.

Our proof is **PR-uniform** and implies computable isometric universality.

Theorem (BBBBDKKMN 2022)

Every primitive recursive Polish space can be primitively recursively isometrically embedded into $C[0,1]$ (represented by piecewise linear functions).

The proof gives a new explicit embedding.

Our proof is **PR-uniform** and implies computable isometric universality.

Theorem (BBBBDKKMN 2022)

Every primitive recursive Polish space can be primitively recursively isometrically embedded into $C[0,1]$ (represented by piecewise linear functions).

The proof gives a new explicit embedding.

Our proof is **PR-uniform** and implies computable isometric universality.

Theorem (BBBBDKKMN 2022)

Every primitive recursive Polish space can be primitively recursively isometrically embedded into $C[0,1]$ (represented by piecewise linear functions).

The proof gives a new explicit embedding.

Our proof is **PR-uniform** and implies computable isometric universality.

Theorem (BBBBDKKMN 2022)

Every primitive recursive Polish space can be primitively recursively isometrically embedded into $C[0,1]$ (represented by piecewise linear functions).

The proof gives a new explicit embedding.

Our proof is **PR-uniform** and implies computable isometric universality.

Part 5: Computable Banach–Stone duality

Consider the space $C(K) = C(K; \mathbb{R})$, where K is compact Polish.

Theorem (Banach-Stone)

$C(K_0) \cong C(K_0) \iff K_0 \cong_{hom} K_1$.

So the homeomorphism type of K 'determines' $C(K)$.

Question (McNicholl)

Is this effective? Is K computably presentable iff $C(K)$ is?

Consider the space $C(K) = C(K; \mathbb{R})$, where K is compact Polish.

Theorem (Banach-Stone)

$C(K_0) \cong C(K_0) \iff K_0 \cong_{hom} K_1.$

So the homeomorphism type of K 'determines' $C(K)$.

Question (McNicholl)

Is this effective? Is K computably presentable iff $C(K)$ is?

Theorem (Bazhenov, Harrison-Trainor, M 2021)

Suppose K is profinite (a Stone space). Then K is computably presentable iff $C(K)$ is.

Here K is viewed up to homeomorphism.

Theorem (Kihara, Hoyrup, Selivanov 2020, Harrison-Trainor, M., Ng 2020)

For a Stone space K, TFAE:

1. K is homeomorphic to a computable space
2. K is homeomorphic to a computably compact space
3. The dual Boolean algebra $\widehat{K}$ has a computable copy.

Corollary

Suppose K is profinite. If $C(K)$ is low_4 then $C(K)$ has a computable presentation.

It follows from:

– Knight and Stob (2000),

– our result,

– effective Stone duality (H-TMN 2020 Kihara, Hoyrup, **Selivanov** 2020).

Corollary

Suppose K is profinite. If $C(K)$ is low_4 then $C(K)$ has a computable presentation.

It follows from:

– Knight and Stob (2000),

– our result,

– effective Stone duality (H-TMN 2020 Kihara, Hoyrup, **Selivanov** 2020).

Corollary

Suppose K is profinite. If $C(K)$ is low_4 then $C(K)$ has a computable presentation.

It follows from:

– Knight and Stob (2000),

– our result,

– effective Stone duality (H-TMN 2020 Kihara, Hoyrup, **Selivanov** 2020).

Happy Birthday!

Thanks

Happy Birthday!

Thanks