Computability & Complexity in Analysis

Tutorial

Vasco Brattka
Department of Mathematics & Applied Mathematics
University of Cape Town
Schedule

A. Computable Real Numbers and Functions
   1. Background
   2. Computability Notions for Real Numbers and Real Functions
   3. Representations, Computability and Continuity

B. Computable Subsets
   4. Computability Notions for Subsets and Functions
   5. Computable Metric Spaces and Computable Functional Analysis
   6. Theory of Admissible Representations

C. Complexity
   7. Computational Complexity on Real Numbers
   8. Discrete Complexity Classes and Continuous Problems
   9. Degrees of Unsolvability
Goals

The goal of this tutorial is to provide a brief introduction into the theory of computability and complexity on the real numbers.

- The objective of this theory is to study algorithmic aspects of real numbers, real number functions and subsets of real numbers.
- This theory has been developed by Turing, Banach and Mazur, Grzegorczyk, Lacombe, Hauck, Nerode, Pour-El and Richards, Kreitz and Weihrauch, Friedman and Ko, and many others.
- Computable analysis uses the point of view of computability and complexity theory to study problems in the domain of analysis.
- Such problems are relevant in many disciplines such as computational geometry, numerical analysis, theory of neural networks, quantum computing, and many other areas.
- The techniques applied in computable analysis include a mixture of computability theory, complexity theory, and topology.
The Context of Computable Analysis

- Complexity theory
- Numerical analysis
- Computability theory
- Analysis

Computable analysis
The Context of Computable Analysis

Computable analysis

- Information based complexity
- Complexity theory
- Numerical analysis
- Domain theory
- Interval analysis
- Logic
- Computability theory
- Analysis
- Constructive analysis
Aspects of Computable Analysis

Topology

Computability theory

Complexity theory
Aspects of Computable Analysis

Topology

*How are infinite objects approximated?*

Computability theory

Complexity theory
Aspects of Computable Analysis

Topology

How are infinite objects approximated?

Computability theory

How can one compute with infinite objects?

Complexity theory
Aspects of Computable Analysis

Topology

*How are infinite objects approximated?*

Computability theory

*How can one compute with infinite objects?*

Complexity theory

*How can one compute efficiently with infinite objects?*
The Starting Point of Computable Analysis

The “computable” numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means.

Alan Turing 1912–1954

The elementary operations the machine can perform are:

- move some head one position to the left or right,
- write a symbol $a \in \Sigma$ on the position under some head,
- compare the symbol under some head with the symbol $a \in \Sigma$.

The output tape is a one-way tape (the head cannot move to the left).
Notions from Computability Theory

Definition 1

- A function $f : \mathbb{N} \to \mathbb{N}$ or $f : \mathbb{N} \to \mathbb{Q}$ is called \textit{computable}, if there exists a Turing machine which can transfer each number $n \in \mathbb{N}$, encoded on the input tape, into the corresponding function value $f(n)$ which is to be written on the output tape in finite time.

- A set $A \subseteq \mathbb{N}$ or $A \subseteq \mathbb{Q}$ is called \textit{computable} or \textit{decidable} or \textit{recursive}, if its characteristic function $\chi_A : \mathbb{N} \to \mathbb{N}$ or $\chi_A : \mathbb{Q} \to \mathbb{N}$, respectively, with

\[
\chi_A(n) := \begin{cases} 
0 & \text{if } n \in A \\
1 & \text{otherwise}
\end{cases}
\]

is computable.

- A set $A \subseteq \mathbb{N}$ is called \textit{computably enumerable} or \textit{recursively enumerable}, for short \textit{c.e.} or \textit{r.e.}, if it is empty or if there exists a computable function $f : \mathbb{N} \to \mathbb{N}$ or $f : \mathbb{N} \to \mathbb{Q}$, respectively, such that $A = \text{range}(f)$. 
The Field of Computable Real Numbers

Definition 2 A real number is called computable, if there exists a Turing machine which can produce its decimal expansion (without input).
The Field of Computable Real Numbers

Definition 2 A real number is called \textit{computable}, if there exists a Turing machine which can produce its decimal expansion (without input).

Theorem 3 (Rice 1954) \textit{The set of computable real numbers $\mathbb{R}_c$ forms a real algebraically closed field.}
The Field of Computable Real Numbers

**Definition 2** A real number is called *computable*, if there exists a Turing machine which can produce its decimal expansion (without input).

**Theorem 3 (Rice 1954)** *The set of computable real numbers* \( \mathbb{R}_c \) *forms a real algebraically closed field.*

- The field \( \mathbb{R}_c \) contains all particular real numbers we ever met in analysis (such as \( \sqrt{2}, \pi, e \) etc.).
- In particular, \( \mathbb{Q} \subseteq \mathbb{A} \subseteq \mathbb{R}_c \subseteq \mathbb{R} \) (where \( \mathbb{A} \) denotes the set of algebraic numbers).
- Nevertheless, \( \mathbb{R}_c \) is only countable.
- The field \( \mathbb{R}_c \) is not complete.
- In the Russian school, computability of functions \( f : \mathbb{R}_c \to \mathbb{R}_c \) is studied (cf. the work of Markov, Sanin, Kushner, Aberth).
Characterisation of Computable Real Numbers

Theorem 4 Let $x \in [0, 1]$. Then the following are equivalent:

- $x$ is a computable real number,
- there exists a computable sequence of rational numbers $(q_n)_{n \in \mathbb{N}}$ which converges rapidly to $x$, i.e. $|q_i - x| < 2^{-i}$ for all $i$,
- $\{q \in \mathbb{Q} : q < x\}$ is a recursive set,
- there exists a recursive set $A \subseteq \mathbb{N}$ such that $x = x_A := \sum_{i \in A} 2^{-i-1}$,
- $x$ admits a computable continued fraction expansion, i.e. there exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that
  \[ x = f(0) + \frac{1}{f(1) + \frac{1}{f(2) + \frac{1}{f(3) + \ldots}}}. \]
Characterisation of Computable Real Numbers

Proof. We sketch the proof of

\[ \{ q \in \mathbb{Q} : q < x \} \text{ is a recursive set} \]

\[ \iff \exists \text{ computable } (q_n)_{n \in \mathbb{N}} \text{ with } |q_i - x| < 2^{-i} \text{ for all } i. \]

"\( \Rightarrow \)" For each \( i \) determine some \( q_i \in \mathbb{Q} \) with \( q_i < x \) and not \( q_i + 2^{-i-1} < x \). These properties are decidable and a suitable \( i \) always exists. It follows \( q_i < x \leq q_i + 2^{-i-1} \) and thus \( |q_i - x| < 2^{-i} \).

"\( \Leftarrow \)"

1. Case: \( x \in \mathbb{Q} \). It is easy to decide \( q < x \) for a given \( q \in \mathbb{Q} \).
2. Case: \( x \not\in \mathbb{Q} \). For given \( q \in \mathbb{Q} \) we determine some \( i \in \mathbb{N} \) such that

\[ |q_i - x| < 2^{-i} \text{ and } |q_i - q| > 2^{-i}. \]

Such an \( i \) exists since \( x \not\in \mathbb{Q} \). Then \( q < x \iff q < q_i \), which is decidable. \( \square \)
Characterisation of Computable Real Numbers

Proof. We sketch the proof of
\[
\{ q \in \mathbb{Q} : q < x \} \text{ is a recursive set}
\]
\[
\iff \exists \text{ computable } (q_n)_{n \in \mathbb{N}} \text{ with } |q_i - x| < 2^{-i} \text{ for all } i.
\]

“\(\implies\)” For each \(i\) determine some \(q_i \in \mathbb{Q}\) with \(q_i < x\) and not \(q_i + 2^{-i-1} < x\). These properties are decidable and a suitable \(i\) always exists. It follows \(q_i < x \leq q_i + 2^{-i-1}\) and thus \(|q_i - x| < 2^{-i}\).

“\(\impliedby\)” 1. Case: \(x \in \mathbb{Q}\). It is easy to decide \(q < x\) for a given \(q \in \mathbb{Q}\).
2. Case: \(x \not\in \mathbb{Q}\). For given \(q \in \mathbb{Q}\) we determine some \(i \in \mathbb{N}\) such that
\[
|q_i - x| < 2^{-i} \text{ and } |q_i - q| > 2^{-i}.
\]
Such an \(i\) exists since \(x \not\in \mathbb{Q}\). Then \(q < x \iff q < q_i\), which is decidable. \(\square\)

Remark 5 The proof of “\(\impliedby\)” contains a non-constructive case distinction that cannot be removed!
Left Computable and Non-computable Real Numbers

Definition 6 A real number $x \in \mathbb{R}$ is called left computable or computably enumerable, if $\{ q \in \mathbb{Q} : q < x \}$ is recursively enumerable.
Left Computable and Non-computable Real Numbers

Definition 6 A real number $x \in \mathbb{R}$ is called *left computable* or *computably enumerable*, if $\{q \in \mathbb{Q} : q < x\}$ is recursively enumerable.

And $x$ is called *right computable*, if $-x$ is left computable.
Definition 6 A real number $x \in \mathbb{R}$ is called \textit{left computable} or \textit{computably enumerable}, if \( \{ q \in \mathbb{Q} : q < x \} \) is recursively enumerable.

And $x$ is called \textit{right computable}, if $-x$ is left computable.

Example 7 (Specker 1949) For any r.e. subset $A \subseteq \mathbb{N}$ the real number

$$x_A := \sum_{i \in A} 2^{-i}$$

is left computable (it is computable, if and only if $A$ is recursive).
Left Computable and Non-computable Real Numbers

Definition 6  A real number $x \in \mathbb{R}$ is called \textit{left computable} or \textit{computably enumerable}, if \{\(q \in \mathbb{Q}: q < x\)\} is recursively enumerable.

And \(x\) is called \textit{right computable}, if \(-x\) is left computable.

Example 7 (Specker 1949)  \textit{For any r.e. subset} \(A \subseteq \mathbb{N}\) \textit{the real number}

\[
x_A := \sum_{i \in A} 2^{-i}
\]

\textit{is left computable (it is computable, if and only if} \(A\) \textit{is recursive}).

Proposition 8 (Jockusch 1969) \textit{If} \(A\) \textit{is r.e. but not recursive, then} \(x_{A \oplus A^c}\) \textit{is left computable}. 
Left Computable and Non-computable Real Numbers

Definition 6 A real number \( x \in \mathbb{R} \) is called *left computable* or *computably enumerable*, if \( \{ q \in \mathbb{Q} : q < x \} \) is recursively enumerable.

And \( x \) is called *right computable*, if \(-x\) is left computable.

Example 7 (Specker 1949) For any r.e. subset \( A \subseteq \mathbb{N} \) the real number

\[
x_A := \sum_{i \in A} 2^{-i}
\]

is left computable (it is computable, if and only if \( A \) is recursive).

Proposition 8 (Jockusch 1969) If \( A \) is r.e. but not recursive, then \( x_{A \oplus A^c} \) is left computable.

Here

\[
A \oplus A^c = \{2n : n \in A\} \cup \{2n + 1 : n \notin A\}
\]

is not r.e.
Other Classes of Computable Reals

Many other classes of computability properties of reals have been studied; some examples are:

- **strongly c.e. real**, which are reals \( x_A \) with r.e. \( A \),

- **semi-computable reals**, which are reals that are left or right computable; it turns out that these are reals which can be represented as a limit of a monotonically converging computable sequence of rationals,

- **weakly computable reals** or **d.c.e. reals**, which are reals that can be represented as differences of left computable reals; this class is a field and the arithmetical closure of the class of left computable reals,

- **computably approximable reals**, which are reals that can be obtained as a limit of a computable sequence of rationals (not necessarily rapidly converging); this class is a field which is closed under computable real functions.
Further Reading on Computability Properties of Real Numbers


Definition 9 A Turing machine $M$ computes a function $f : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$, if on input $p \in \Sigma^\omega$ the following holds true:

- if $p \in \text{dom}(f)$, then $M$ computes infinitely long and in the long run it writes the infinite function value $f(p)$ on the one-way output tape,
- if $p \notin \text{dom}(f)$, then $M$ does not produce an infinite output.
Theorem 10  The function $f : \mathbb{R} \to \mathbb{R}, x \mapsto 3x$ is not computable with respect to the decimal representation.
Theorem 10  The function \( f : \mathbb{R} \to \mathbb{R}, x \mapsto 3x \) is not computable with respect to the decimal representation.

Proof. Assume, there exists a Turing machine \( M \) which computes \( f \) with respect to the decimal representation. Let us consider the input \( p = .333\ldots \) which represents the value \( 1/3 \). With this input \( M \) has to produce output \( q = 1.000\ldots \) or output \( q = .999\ldots \), both representing the value \( 1 \). We consider the second case:

\[
\begin{array}{cccccccccc}
\cdot & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & \ldots \rightarrow p \\
\downarrow & & & & & & & & & \leftarrow M \\
\cdot & 9 & 9 & 9 & 9 & \ldots \rightarrow q = F_M(p)
\end{array}
\]
Theorem 10 The function $f : \mathbb{R} \to \mathbb{R}, x \mapsto 3x$ is not computable with respect to the decimal representation.

Proof. Assume, there exists a Turing machine $M$ which computes $f$ with respect to the decimal representation. Let us consider the input $p = .333...$ which represents the value $1/3$. With this input $M$ has to produce output $q = 1.000...$ or output $q = .999...$, both representing the value $1$. We consider the second case:

```
\begin{array}{cccccccccc}
& 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & \ldots
\end{array}
\begin{array}{c}
p
\end{array}
```

```
\begin{array}{cccccccccc}
& 9 & 9 & 9 & 9 & & & & \ldots
\end{array}
\begin{array}{c}
q = F_M(p)
\end{array}
```
Multiplication by 3 with the Decimal Representation

**Theorem 10**  *The function* $f : \mathbb{R} \to \mathbb{R}, x \mapsto 3x$ *is not computable with respect to the decimal representation.*

**Proof.** Assume, there exists a Turing machine $M$ which computes $f$ with respect to the decimal representation. Let us consider the input $p = .333\ldots$ which represents the value $1/3$. With this input $M$ has to produce output $q = 1.000\ldots$ or output $q = .999\ldots$, both representing the value 1. We consider the second case:

\[
\begin{array}{cccccccccc}
\cdot & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & \ldots \\
\cdot & 9 & 9 & 9 & 9 & 9 & 9 & 9 & & \ldots \\
\end{array}
\]

$p$

$q = F_M(p)$
Theorem 10  The function $f : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto 3x$ is not computable with respect to the decimal representation.

Proof. Assume, there exists a Turing machine $M$ which computes $f$ with respect to the decimal representation. Let us consider the input $p = .333...$ which represents the value $1/3$. With this input $M$ has to produce output $q = 1.000...$ or output $q = .999...$, both representing the value 1. We consider the second case:

$$
\begin{array}{cccccccc}
\cdot & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & \cdots & p' \\
\cdot & 9 & 9 & 9 & 9 & 9 & 9 & \cdots & q' = F_M(p')
\end{array}
$$
Multiplication by 3 with the Decimal Representation

Theorem 10  *The function* \( f : \mathbb{R} \to \mathbb{R}, x \mapsto 3x \) *is not computable with respect to the decimal representation.*

**Proof.** Assume, there exists a Turing machine \( M \) which computes \( f \) with respect to the decimal representation. Let us consider the input \( p = .333... \) which represents the value \( 1/3 \). With this input \( M \) has to produce output \( q = 1.000... \) or output \( q = .999... \), both representing the value 1. We consider the second case:

\[
\begin{array}{cccccccc}
  & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & \ldots \\
\cdot & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & \ldots \\
\hline
M
\end{array}
\]

\[
\begin{array}{cccccccc}
  & 9 & 9 & 9 & 9 & 9 & 9 & ? & \ldots \\
\cdot & 9 & 9 & 9 & 9 & 9 & 9 & ? & \ldots \\
\hline
q' = F_M(p')
\end{array}
\]

For the modified input \( p' = .3333333433... \) with value \( x' > 1/3 \) it follows that 1. is not a prefix of the corresponding output \( q' \) of the machine. But that means that \( q' \) does not represent the correct result \( f(x') \). Contradiction! \( \square \)
Conclusions

Corollary 11  *Neither addition nor multiplication is computable with respect to the decimal representation of the reals.*
Conclusions

Corollary 11  *Neither addition nor multiplication is computable with respect to the decimal representation of the reals.*

- Either the Turing machine model is not appropriate for real number computations in principle,
- or the model has to be modified such that addition and multiplication become computable.
- This can be achieved by allowing two-way output tapes. However, then the model is not closed under composition and the output after finite time would be completely useless.
- Alternatively, one can replace the decimal representation by more suitable representations. This was Turing’s suggestion in: A.M. Turing, On computable numbers, with an application to the “Entscheidungsproblem”. A correction., *Proc. London Math. Soc.* 43 (1937) 544–546.
Definition 12  A *representation* of a set $X$ is a surjective function
\[ \delta : \subseteq \Sigma^\omega \to X. \]
Representations and Computable Functions

Definition 12 A representation of a set $X$ is a surjective function $\delta : \subseteq \Sigma^\omega \rightarrow X$.

Definition 13 A function $f : \subseteq X \rightarrow Y$ is called $(\delta, \delta')$–computable, if there exists a computable function $F : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ such that

$$\delta' F(p) = f\delta(p)$$

for all $p \in \text{dom}(f\delta)$.

Analogously, $f$ is called $(\delta, \delta')$–continuous, if there exists a continuous $F$ such that the equation above holds.
Cauchy Representation

Definition 14  The **Cauchy representation** $\rho_C : \subseteq \Sigma^\omega \rightarrow \mathbb{R}$ is defined by:

$$\rho_C(w_0 \# w_1 \# w_2 \ldots) = x$$

$$: \iff \lim_{n \rightarrow \infty} \nu_Q(w_n) = x \text{ and } (\forall i < j) \ |\nu_Q(w_i) - \nu_Q(w_j)| \leq 2^{-i}. $$

Here $\nu_Q : \subseteq \Sigma^* \rightarrow \mathbb{Q}$ is supposed to be some standard notation of the rational numbers (which is not defined on words that include $\#$).
Cauchy Representation

**Definition 14** The *Cauchy representation* $\rho_C : \subseteq \Sigma^\omega \rightarrow \mathbb{R}$ is defined by:

$$\rho_C(w_0 \# w_1 \# w_2 \ldots) = x$$

$$\iff \lim_{n \rightarrow \infty} \nu_Q(w_n) = x \quad \text{and} \quad (\forall i < j) \quad |\nu_Q(w_i) - \nu_Q(w_j)| \leq 2^{-i}.$$  

Here $\nu_Q : \subseteq \Sigma^* \rightarrow \mathbb{Q}$ is supposed to be some standard notation of the rational numbers (which is not defined on words that include $\#$).

**Definition 15** Let $\delta$ be a representation of $X$. A point $x \in X$ is called $\delta$–computable, if there exists a computable sequence $p \in \Sigma^\omega$ such that $\delta(p) = x$. 
Cauchy Representation

Definition 14 The Cauchy representation $\rho_C : \subseteq \Sigma^\omega \to \mathbb{R}$ is defined by:

$$\rho_C(w_0 \# w_1 \# w_2 \ldots) = x$$

$$: \iff \lim_{n \to \infty} \nu_Q(w_n) = x \text{ and } (\forall i < j) \ |\nu_Q(w_i) - \nu_Q(w_j)| \leq 2^{-i}.$$  

Here $\nu_Q : \subseteq \Sigma^* \to \mathbb{Q}$ is supposed to be some standard notation of the rational numbers (which is not defined on words that include $\#$).

Definition 15 Let $\delta$ be a representation of $X$. A point $x \in X$ is called $\delta$–computable, if there exists a computable sequence $p \in \Sigma^\omega$ such that $\delta(p) = x$.

Corollary 16 The computable real numbers coincide with the $\rho_C$–computable real numbers.
Left and Right Representation

Definition 17  The representation $\rho_\prec : \subseteq \Sigma^\omega \rightarrow \mathbb{R}$ is defined as follows:

$$\rho_\prec(w_0\#w_1\#w_2\ldots) = x : \iff \{ q \in \mathbb{Q} : q < x \} = \{ \nu_{\mathbb{Q}}(w_i) : i \in \mathbb{N} \}. $$

Analogously, one can define $\rho_\succ$ with “>” instead of “<”.
Left and Right Representation

Definition 17  The representation $\rho_\prec : \subseteq \Sigma^\omega \rightarrow \mathbb{R}$ is defined as follows:

$$\rho_\prec(w_0 \# w_1 \# w_2 \ldots) = x : \iff \{ q \in \mathbb{Q} : q \prec x \} = \{ \nu_{\mathbb{Q}}(w_i) : i \in \mathbb{N} \}.$$ 

Analogously, one can define $\rho_\succ$ with “$>$” instead of “$<$”.

Corollary 18  The $\rho_\prec$–computable numbers are just the left computable reals and the $\rho_\succ$–computable numbers are just the right computable reals.
Left and Right Representation

Definition 17 The representation $\rho_\prec : \subseteq \Sigma^\omega \to \mathbb{R}$ is defined as follows:

$$\rho_\prec (w_0 \# w_1 \# w_2 \ldots) = x : \iff \{ q \in \mathbb{Q} : q < x \} = \{ \nu_\mathbb{Q}(w_i) : i \in \mathbb{N} \}.$$ 

Analogously, one can define $\rho_\succ$ with “>” instead of “<”.

Corollary 18 The $\rho_\prec$–computable numbers are just the left computable reals and the $\rho_\succ$–computable numbers are just the right computable reals.

Definition 19 One can define the decimal representation $\rho_{10} : \subseteq \Sigma^\omega \to \mathbb{R}$ in the straightforward way such that

- $\rho_{10}(1.414\ldots) = \sqrt{2}$,
- $\rho_{10}(3.1415\ldots) = \pi$,
- $\rho_{10}(-0.999\ldots) = -1$ etc.
Reducibility

Definition 20 Let $\delta, \delta' : \subseteq \Sigma^\omega \to X$ be representations.

- Then $\delta$ is called reducible to $\delta'$, for short $\delta \leq \delta'$, if there exists a computable function $F : \subseteq \Sigma^\omega \to \Sigma^\omega$ such that
  \[
  \delta(p) = \delta'F(p)
  \]
  for all $p \in \text{dom}(\delta)$.

- Moreover, $\delta$ is called equivalent to $\delta'$, for short $\delta \equiv \delta'$, if $\delta \leq \delta'$ and $\delta' \leq \delta$.

Similarly, one can define topological variants $\leq_t$ and $\equiv_t$ of these notions.
Reducibility

Definition 20 Let $\delta, \delta' : \subseteq \Sigma^\omega \to X$ be representations.

- Then $\delta$ is called reducible to $\delta'$, for short $\delta \leq \delta'$, if there exists a computable function $F : \subseteq \Sigma^\omega \to \Sigma^\omega$ such that
  $$\delta(p) = \delta'F(p)$$
  for all $p \in \text{dom}(\delta)$.

- Moreover, $\delta$ is called equivalent to $\delta'$, for short $\delta \equiv \delta'$, if $\delta \leq \delta'$ and $\delta' \leq \delta$.

Similarly, one can define topological variants $\leq_t$ and $\equiv_t$ of these notions.

Lemma 21 $\delta$ is reducible to $\delta'$, if and only if the identity $\text{id} : X \to X$ is $(\delta, \delta')$–computable.
Reducibility

Definition 20  Let $\delta, \delta' : \subseteq \Sigma^\omega \rightarrow X$ be representations.

- Then $\delta$ is called reducible to $\delta'$, for short $\delta \leq \delta'$, if there exists a computable function $F : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ such that

$\delta(p) = \delta' F(p)$

for all $p \in \text{dom}(\delta)$.

- Moreover, $\delta$ is called equivalent to $\delta'$, for short $\delta \equiv \delta'$, if $\delta \leq \delta'$ and $\delta' \leq \delta$.

Similarly, one can define topological variants $\leq_t$ and $\equiv_t$ of these notions.

Lemma 21  $\delta$ is reducible to $\delta'$, if and only if the identity $\text{id} : X \rightarrow X$ is $(\delta, \delta')$–computable.

Lemma 22  Reducibility is a preorder on the set of representations.
The Lattice of Real Number Representations

Naive Cauchy representation $\rho_{nC}$

Enumeration of left cuts

$\rho^<$

$\rho^>$

Enumeration of right cuts

Cauchy representation $\rho_C \equiv \rho^< \land \rho^>$

Representation $\rho_b$ for basis $b$

Characteristic functions of left cuts

Characteristic functions of right cuts

Continued fraction representation

Each arrow means $\leq$ and $\not\geq_t$. 
The Lattice of Real Number Representations

Computably approximable reals
Naive Cauchy representation $\rho_{nC}$

Left computable reals
Enumeration of left cuts

Computable reals

Right computable reals
Enumeration of right cuts

Cauchy representation $\rho_C \equiv \rho_{<} \land \rho_{>}$

Representation $\rho_b$ for basis $b$

Characteristic functions of left cuts

Characteristic functions of right cuts

Continued fraction representation

Each arrow means $\leq$ and $\not\geq_t$. 

Vasco Brattka Laboratory of Foundational Aspects of Computer Science · University of Cape Town
Some Observations on Uniformity

- The notion of a computable real number $x \in \mathbb{R}$ is quite stable under changes of the representation (Cauchy sequences, base $b$, continued fractions, characteristic functions of Dedekind cuts).
- The notion of a computable sequence $s : \mathbb{N} \to \mathbb{R}$ does sensitively rely on the underlying representation (Mostowski 1957).
- The notion of a computable function $f : \mathbb{R} \to \mathbb{R}$ does also sensitively rely on the underlying representations (Turing 1937).
- The differences between equivalence classes of representations are essentially of topological nature.
- The most natural equivalence class is the one of the Cauchy representation $\rho_C$; the underlying topology is the Euclidean one.
- From now on computability is always understood with respect to the Cauchy representation $\rho_C$. 
The following functions are computable:

- The arithmetical operations $+, -, \cdot, \div : \subseteq \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.
- The absolute value function $\text{abs} : \mathbb{R} \to \mathbb{R}, x \mapsto |x|$.
- The functions $\text{min}, \text{max} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.
- The constant functions $\mathbb{R} \to \mathbb{R}, x \mapsto c$ with computable value $c \in \mathbb{R}$.
- The projections $\text{pr}_i : \mathbb{R}^n \to \mathbb{R}, (x_1, \ldots, x_n) \mapsto x_i$.
- All polynomials $p : \mathbb{R}^n \to \mathbb{R}$ with computable coefficients.
- The exponential function and the trigonometric functions $\exp, \sin, \cos : \mathbb{R} \to \mathbb{R}$.
- The square root function and the logarithm function $\sqrt{\cdot}, \log : \subseteq \mathbb{R} \to \mathbb{R}$ with their natural domains.
Computable Real Number Functions

Proof. We sketch the proof that addition \( f : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \) is computable. Given two sequences \( (q_n)_{n \in \mathbb{N}} \) and \( (r_n)_{n \in \mathbb{N}} \) of rational numbers which rapidly converge to \( x \) and \( y \), respectively, we can compute a sequence \( (p_n)_{n \in \mathbb{N}} \) of rational numbers by \( p_n := q_{n+1} + r_{n+1} \) which converges to \( x + y \) and which fulfills

\[
|p_i - p_j| \leq |q_{i+1} - q_{j+1}| + |r_{i+1} - r_{j+1}| \leq 2^{-i-1} + 2^{-j-1} \leq 2^{-i}
\]

for all \( i > j \). Thus, \( (p_n)_{n \in \mathbb{N}} \) converges even rapidly to \( x + y = f(x, y) \). Since addition on rational number can be computed by Turing machines, it follows that \( f \) is computable as well. \( \square \)

Remark 24  Addition just requires a uniform lookahead of one step which does not depend on the input. For functions that are not uniformly continuous, such as multiplication, the modulus of continuity and thus the lookahead might depend on the input.
Cantor Space

- **Cantor space** is the metric space \((\Sigma^\omega, d_C)\) over some (finite) alphabet \(\Sigma\), endowed with the metric \(d_C\), defined by

\[
d_C(p, q) := \begin{cases} 
  2^{-\min\{i : p(i) \neq q(i)\}} & \text{if } p \neq q \\
  0 & \text{otherwise}
\end{cases}
\]

for all \(p, q \in \Sigma^\omega\).

- The metric space \((\Sigma^\omega, d_C)\) is separable and complete but not connected (in fact, it is totally disconnected).

- Let \(\tau_C\) be the topology induced by the metric \(d_C\). The topological space \((\Sigma^\omega, \tau_C)\) is also called Cantor space.

- The set \(\{w\Sigma^\omega : w \in \Sigma^*\}\) is a basis of the topology \(\tau_C\).

- If \(\Sigma\) is finite, then the space \((\Sigma^\omega, \tau_C)\) is compact.
An Open Set in Cantor space

Open set $U = \bigcup_{i=0}^{\infty} w_i \Sigma^\omega$ in Cantor space:
Proposition 25  Any representation which is equivalent to the Cauchy representation $\rho_C : \subseteq \Sigma^\omega \rightarrow \mathbb{R}$ is continuous (and admits an open and surjective restriction).
Topological Properties of the Cauchy Representation

Proposition 25  Any representation which is equivalent to the Cauchy representation \( \rho_C : \subseteq \Sigma^\omega \rightarrow \mathbb{R} \) is continuous (and admits an open and surjective restriction).

Theorem 26

1. There is no injective representation \( \delta : \subseteq \Sigma^\omega \rightarrow \mathbb{R} \), which is equivalent to \( \rho_C \).

2. There is no total representation \( \delta : \Sigma^\omega \rightarrow \mathbb{R} \), which is equivalent to \( \rho_C \) (given that \( \Sigma \) is finite).
Topological Properties of the Cauchy Representation

Proposition 25  Any representation which is equivalent to the Cauchy representation \( \rho_C : \subseteq \Sigma^\omega \to \mathbb{R} \) is continuous (and admits an open and surjective restriction).

Theorem 26

1. There is no injective representation \( \delta : \subseteq \Sigma^\omega \to \mathbb{R} \), which is equivalent to \( \rho_C \).

2. There is no total representation \( \delta : \Sigma^\omega \to \mathbb{R} \), which is equivalent to \( \rho_C \) (given that \( \Sigma \) is finite).

<table>
<thead>
<tr>
<th></th>
<th>Cantor space</th>
<th>Euclidean space</th>
</tr>
</thead>
<tbody>
<tr>
<td>connected</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>compact</td>
<td>+</td>
<td>–</td>
</tr>
</tbody>
</table>
Continuity of Computable Functions

Theorem 27 Each computable function \( f : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega \) is continuous (with respect to the Cantor topology).

Proof. Let \( M \) be a machine with \( f_M = f \), \( p \in \text{dom}(f) \) and \( v \) a prefix of \( q = f(p) \). After a number \( t \) of steps machine \( M \) with input \( p \) has just written \( v \) on the output tape:

At this time \( M \) has read at most a finite prefix \( w \) of \( p \) and we obtain \( f(w \Sigma^\omega) \subseteq v \Sigma^\omega \). Thus, \( f \) is continuous.
Continuity of Computable Real Functions

**Theorem 28** Each computable function $f : \subseteq \mathbb{R} \to \mathbb{R}$ is continuous (with respect to the Euclidean topology).

**Proof.** Let $f : \subseteq \mathbb{R} \to \mathbb{R}$ be computable, i.e. $(\rho_C, \rho_C)$–computable. Then there exists a computable function $F : \subseteq \Sigma^\omega \to \Sigma^\omega$ such that $f\rho_C(p) = \rho_C F(p)$ holds for all $p \in \text{dom}(f\rho_C)$. By the previous theorem $F$ is continuous with respect to the Cantor topology. In two steps we can prove that $f$ has to be continuous as well:

1. Since $\rho_C$ is continuous, it follows that $f\rho_C = \rho_C F$ is continuous.

2. Since $\rho_C$ has an open and surjective restriction, it follows that $f$ itself is continuous.
Theorem 29 If $f, g : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ are computable functions, then $g \circ f : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ is a computable function.
Composition of Computable Functions

Theorem 29  If $f, g : \subseteq \Sigma^\omega \to \Sigma^\omega$ are computable functions, then $g \circ f : \subseteq \Sigma^\omega \to \Sigma^\omega$ is a computable function.

Corollary 30  If $f : \subseteq \Sigma^\omega \to \Sigma^\omega$ is a computable function and $x \in \Sigma^\omega$ a computable point, then $f(x) \in \Sigma^\omega$ is a computable point as well.
Composition of Computable Functions

Theorem 29  If $f, g : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ are computable functions, then $g \circ f : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ is a computable function.

Corollary 30  If $f : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ is a computable function and $x \in \Sigma^\omega$ a computable point, then $f(p) \in \Sigma^\omega$ is a computable point as well.

Corollary 31  If $f, g : \subseteq \mathbb{R} \rightarrow \mathbb{R}$ are computable functions, then $g \circ f : \subseteq \mathbb{R} \rightarrow \mathbb{R}$ is a computable function.
Composition of Computable Functions

Theorem 29  If $f, g : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ are computable functions, then $g \circ f : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ is a computable function.

Corollary 30  If $f : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ is a computable function and $x \in \Sigma^\omega$ a computable point, then $f(p) \in \Sigma^\omega$ is a computable point as well.

Corollary 31  If $f, g : \subseteq \mathbb{R} \rightarrow \mathbb{R}$ are computable functions, then $g \circ f : \subseteq \mathbb{R} \rightarrow \mathbb{R}$ is a computable function.

Corollary 32  If $f : \subseteq \mathbb{R} \rightarrow \mathbb{R}$ is a computable function and $x \in \mathbb{R}$ a computable number, then $f(x) \in \mathbb{R}$ is a computable number as well.
Computability, Continuity and Invariance

\[ f : \mathbb{R} \to \mathbb{R} \text{ computable} \quad \Rightarrow \quad f : \mathbb{R} \to \mathbb{R} \text{ continuous} \]

\[ f : \mathbb{R} \to \mathbb{R} \text{ computably invariant}, \]
i.e. \( f \) maps computable points to computable points
Computability, Continuity and Invariance

\[ f : \mathbb{R} \rightarrow \mathbb{R} \text{ computable} \quad \Rightarrow \quad f : \mathbb{R} \rightarrow \mathbb{R} \text{ continuous} \]

\[ f : \mathbb{R} \rightarrow \mathbb{R} \text{ Markov-computable,} \]
i.e. there is an algorithm which transfers programs of \( x \in \mathbb{R}_c \) into programs of \( f(x) \)

\[ f : \mathbb{R} \rightarrow \mathbb{R} \text{ Banach-Mazur computable,} \]
i.e. \( f \) maps computable sequences to computable sequences

\[ f : \mathbb{R} \rightarrow \mathbb{R} \text{ computably invariant,} \]
i.e. \( f \) maps computable points to computable points
Computability, Continuity and Invariance

\( f : \mathbb{R} \to \mathbb{R} \) computable

\( \xrightarrow{\text{\hspace{1cm}}} \)

\( f : \mathbb{R} \to \mathbb{R} \) continuous

\( f : \mathbb{R} \to \mathbb{R} \) Markov-computable,

i.e. there is an algorithm which transfers programs of \( x \in \mathbb{R}_c \) into programs of \( f(x) \)

\\( \xrightarrow{\text{\hspace{1cm}}} \)

\( f : \mathbb{R} \to \mathbb{R} \) Banach-Mazur computable,

i.e. \( f \) maps computable sequences to computable sequences

\\( \xrightarrow{\text{\hspace{1cm}}} \)

\( f : \mathbb{R} \to \mathbb{R} \) computably invariant,

i.e. \( f \) maps computable points to computable points

\( f \mid_{\mathbb{R}_c} : \mathbb{R}_c \to \mathbb{R}_c \) continuous
Computability, Continuity and Invariance

\[ f : \mathbb{R} \to \mathbb{R} \] is computably invariant,
\[ f : \mathbb{R} \to \mathbb{R} \] Banach-Mazur computable,
\[ f : \mathbb{R} \to \mathbb{R} \] Markov-computable,
\[ f : \mathbb{R} \to \mathbb{R} \] computable, 
\[ f : \mathbb{R} \to \mathbb{R} \] continuous 

\[ f|_{\mathbb{R}_c} : \mathbb{R}_c \to \mathbb{R}_c \] continuous

---

Theorem of Ceitin-Kreisel-Lacombe-Shoenfield-Moschovakis

Theorem of Banach-Mazur
Computability, Continuity and Invariance

\( f : \mathbb{R} \to \mathbb{R} \) computable

\( f : \mathbb{R} \to \mathbb{R} \) Markov-computable,

i.e. there is an algorithm which transfers programs of \( x \in \mathbb{R}_c \) into programs of \( f(x) \)

\( f : \mathbb{R} \to \mathbb{R} \) Banach-Mazur computable,

i.e. \( f \) maps computable sequences to computable sequences

\( f : \mathbb{R} \to \mathbb{R} \) computably invariant,

i.e. \( f \) maps computable points to computable points

\( f : \mathbb{R} \to \mathbb{R} \) continuous

Theorem of Banach-Mazur

Theorem of Ceitin-Kreisel-Lacombe-Shoenfield-Moschovakis

Aberth

Pour-El & Richards

Hertling

char. function
Church’s Thesis for Real Number Functions

- Discontinuous functions $f : \mathbb{R} \rightarrow \mathbb{R}$ cannot be computable.
- They might be “simple to describe”
  (e.g. a computable point in some function space),
- or computable in some weaker sense (e.g. lower semi-computable),
- but not computably evaluable with arbitrary precision.
Church’s Thesis for Real Number Functions

- Discontinuous functions $f : \mathbb{R} \to \mathbb{R}$ cannot be computable.
- They might be “simple to describe”
  (e.g. a computable point in some function space),
- or computable in some weaker sense (e.g. lower semi-computable),
- but not computably evaluable with arbitrary precision.

**Thesis.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is computable, if and only if it can be evaluated on a physical computer with arbitrary given precision.
Further Reading on Representations

Further Reading on other Approaches & Comparisons

Text Books on Computable Analysis

Klaus Weihrauch
Computability and Complexity in Analysis
Texts in Theoretical Computer Science
Springer, Berlin, 2000

Ker-I Ko
Complexity Theory of Real Functions
Progress in Theoretical Computer Science
Birkhäuser, Boston, 1991

Marian Pour-El and J. Ian Richards
Computability in Analysis and Physics
Perspectives in Mathematical Logic
Springer, Berlin 1989
Schedule

A. Computable Real Numbers and Functions
   1. Background
   2. Computability Notions for Real Numbers and Real Functions
   3. Representations, Computability and Continuity

B. Computable Subsets
   4. Computability Notions for Subsets and Functions
   5. Computable Metric Spaces and Computable Functional Analysis
   6. Theory of Admissible Representations

C. Complexity
   7. Computational Complexity on Real Numbers
   8. Discrete Complexity Classes and Continuous Problems
   9. Degrees of Unsolvability
Computable Subsets

Question: Which subsets of Euclidean space should be called recursive, recursively enumerable, and co-recursively enumerable?
Question: Which subsets of Euclidean space should be called *recursive*, *recursively enumerable*, and *co-recursively enumerable*?

A recursive subset $A \subseteq \mathbb{R}^2$?
Question: Which subsets of Euclidean space should be called \textit{recursive}, \textit{recursively enumerable}, and \textit{co-recursively enumerable}?

A recursive subset $A \subseteq \mathbb{R}^2$?

Requirements:

- The notions should generalise the corresponding classical notions.
- The notions should have some computational meaning.
- The notions should fit together with the notion of computability.
Decidable Subsets

Definition 33 A subset $A \subseteq \mathbb{R}^n$ is called \textit{decidable}, if its characteristic function $\chi_A : \mathbb{R}^n \rightarrow \mathbb{R}$, defined by

$$\chi_A(x) := \begin{cases} 
0 & \text{if } x \in A \\
1 & \text{else}
\end{cases}$$

is computable.
Decidable Subsets

**Definition 33** A subset $A \subseteq \mathbb{R}^n$ is called *decidable*, if its characteristic function $\chi_A : \mathbb{R}^n \rightarrow \mathbb{R}$, defined by

$$\chi_A(x) := \begin{cases} 
0 & \text{if } x \in A \\
1 & \text{else}
\end{cases}$$

is computable.

**Corollary 34** *The only decidable subsets of the Euclidean space $\mathbb{R}^n$ are $\emptyset$ and $\mathbb{R}^n$ itself.*
Decidable Subsets

Definition 33 A subset $A \subseteq \mathbb{R}^n$ is called \textit{decidable}, if its characteristic function $\chi_A : \mathbb{R}^n \to \mathbb{R}$, defined by

$$\chi_A(x) := \begin{cases} 0 & \text{if } x \in A \\ 1 & \text{else} \end{cases}$$

is computable.

Corollary 34 The only decidable subsets of the Euclidean space $\mathbb{R}^n$ are $\emptyset$ and $\mathbb{R}^n$ itself.

Proof. Any computable characteristic function $\chi_A : \mathbb{R}^n \to \mathbb{R}$ is continuous. Since $\mathbb{R}^n$ is connected, the only continuous characteristic functions $\chi_A$ are those for $A = \emptyset$ and $A = \mathbb{R}^n$. \qed
Theorem 35  There is no representation $\delta$ of the reals such that one of the tests $=, <, \leq$, or more precisely one of the subsets

$$\{(x, y) : x = y\}, \{(x, y) : x < y\}, \{(x, y) : x \leq y\} \subseteq \mathbb{R}^2$$

becomes decidable, i.e. such that the corresponding characteristic function $\chi_A : \mathbb{R}^2 \to \mathbb{R}$ is $(\delta, \delta, \rho)$–computable.
Theorem 35  There is no representation $\delta$ of the reals such that one of the tests $=, <, \leq$, or more precisely one of the subsets

$$\{(x, y) : x = y\}, \{(x, y) : x < y\}, \{(x, y) : x \leq y\} \subseteq \mathbb{R}^2$$

becomes decidable, i.e. such that the corresponding characteristic function $\chi_A : \mathbb{R}^2 \to \mathbb{R}$ is $(\delta, \delta, \rho)$–computable.

Proof. Sketch for the equality test: Let us assume $\delta$ is a representation which makes the equality test decidable and let $M$ be a corresponding Turing machine. Then for each $x \in \mathbb{R}$ there is a $p \in \Sigma^\omega$ with $\delta(p) = x$ and on input $(p, p)$ machine $M$ stops after finite time with a positive answer. After this finite time $M$ has only read a finite common prefix $w_p \in \Sigma^*$ of the input tapes and we obtain $\delta(w_p \Sigma^\omega) = \{x\}$. This implies

$$\mathbb{R} = \text{range}(\delta) = \{\delta(w_p \Sigma^\omega) : p \in \text{dom}(\delta)\}$$

and since $\{w_p \in \Sigma^* : p \in \Sigma^\omega\}$ is countable, it follows that $\mathbb{R}$ is countable. Contradiction!
Computable Subsets

Definition 36 The distance function $d_A : \mathbb{R}^n \to \mathbb{R}$ of a non-empty subset $A \subseteq \mathbb{R}^n$ is defined by $d_A(x) := \inf_{y \in A} ||x - y||$. 
Computable Subsets

Definition 36  The distance function $d_A : \mathbb{R}^n \to \mathbb{R}$ of a non-empty subset $A \subseteq \mathbb{R}^n$ is defined by $d_A(x) := \inf_{y \in A} ||x - y||$. 

\[ \chi_A(x) \]

\[ d_A(x) \]
Computable Subsets

Definition 36  The distance function $d_A : \mathbb{R}^n \to \mathbb{R}$ of a non-empty subset $A \subseteq \mathbb{R}^n$ is defined by $d_A(x) := \inf_{y \in A} ||x - y||$. 

Proposition 37  Any distance function $d_A : \mathbb{R}^n \to \mathbb{R}$ is continuous.
### Computable Subsets

**Definition 36** The *distance function* $d_A : \mathbb{R}^n \rightarrow \mathbb{R}$ of a non-empty subset $A \subseteq \mathbb{R}^n$ is defined by $d_A(x) := \inf_{y \in A} \|x - y\|$.

![Graph of $\chi_A(x)$ and $d_A(x)$](image)

**Proposition 37** Any distance function $d_A : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous.

**Definition 38** Let $A \subseteq \mathbb{R}^n$ be a non-empty closed set.

- $A$ is called *recursively enumerable*, if $d_A : \mathbb{R}^n \rightarrow \mathbb{R}$ is upper semi-computable, (i.e. $(\rho^n_G, \rho^>_C)$–computable),
- $A$ is called *co-recursively enumerable*, if $d_A : \mathbb{R}^n \rightarrow \mathbb{R}$ is lower semi-computable, (i.e. $(\rho^n_G, \rho^<_C)$–computable),
- $A$ is called *recursive*, if $d_A : \mathbb{R}^n \rightarrow \mathbb{R}$ is computable.
The Classical Notions

Proposition 39  Let $A \subseteq \mathbb{N}^n$. Then

1. $A$ is r.e. closed as a subset of $\mathbb{R}^n$, if and only if it is r.e. as a subset of $\mathbb{N}^n$ in the classical sense,

2. $A$ is co-r.e. closed as a subset of $\mathbb{R}^n$, if and only if it is co-r.e. as a subset of $\mathbb{N}^n$ in the classical sense,

3. $A$ is recursive closed as a subset of $\mathbb{R}^n$, if and only if it is recursive as a subset of $\mathbb{N}^n$ in the classical sense.
The Classical Notions

Proposition 39  Let $A \subseteq \mathbb{N}^n$. Then

1. $A$ is r.e. closed as a subset of $\mathbb{R}^n$, if and only if it is r.e. as a subset of $\mathbb{N}^n$ in the classical sense,

2. $A$ is co-r.e. closed as a subset of $\mathbb{R}^n$, if and only if it is co-r.e. as a subset of $\mathbb{N}^n$ in the classical sense,

3. $A$ is recursive closed as a subset of $\mathbb{R}^n$, if and only if it is recursive as a subset of $\mathbb{N}^n$ in the classical sense.

Proof. Idea for the “only if” direction (let $n \in \mathbb{N}$):

1. If $d_A(n) < 1$, then $n \in A$.

2. If $d_A(n) > \frac{1}{2}$, then $n \in \mathbb{N} \setminus A$.

3. Follows from 1. and 2. \qed
Thesis: A closed subset $A \subseteq \mathbb{R}^2$ is recursive, if and only if it can be plotted by a physical computer with arbitrary given resolution.
Theorem 40 For a non-empty closed subset $A \subseteq \mathbb{R}^n$ the following are equivalent:

1. $A$ is co-r.e., i.e. $d_A : \mathbb{R}^n \to \mathbb{R}$ is lower semi-computable,
2. $\{(x, r) \in \mathbb{Q}^n \times \mathbb{Q} : A \cap \overline{B}(x, r) = \emptyset\}$ is r.e.,
3. $\mathbb{R}^n \setminus A = \bigcup_{(x, r) \in I} B(x, r)$ for some r.e. set $I \subseteq \mathbb{Q}^n \times \mathbb{Q}$,
4. $A = f^{-1}\{0\}$ for some computable function $f : \mathbb{R}^n \to \mathbb{R}$,
5. $\chi_A : \mathbb{R}^n \to \mathbb{R}$ is lower semi-computable.
Characterisations of Co-r.e. Subsets

Theorem 40 For a non-empty closed subset $A \subseteq \mathbb{R}^n$ the following are equivalent:

1. $A$ is co-r.e., i.e. $d_A : \mathbb{R}^n \to \mathbb{R}$ is lower semi-computable,
2. $\{(x, r) \in \mathbb{Q}^n \times \mathbb{Q} : A \cap \overline{B}(x, r) = \emptyset\}$ is r.e.,
3. $\mathbb{R}^n \setminus A = \bigcup_{(x,r) \in I} B(x, r)$ for some r.e. set $I \subseteq \mathbb{Q}^n \times \mathbb{Q},$
4. $A = f^{-1}\{0\}$ for some computable function $f : \mathbb{R}^n \to \mathbb{R},$
5. $\chi_A : \mathbb{R}^n \to \mathbb{R}$ is lower semi-computable.

Proof. 1. $\implies$ 2. Follows from $d_A(x) > \varepsilon \iff A \cap \overline{B}(x, \varepsilon) = \emptyset.$
3. $\implies$ 4. If $\mathbb{R}^n \setminus A = \bigcup_{i=0}^{\infty} B(x_i, r_i),$ then

$$f(x) := \sum_{i=0}^{\infty} \frac{\max\{0, r_i - d(x_i, x)\}}{r_i} \cdot 2^{-i-1}$$

defines a computable function $f : \mathbb{R}^n \to \mathbb{R}$ with $A = f^{-1}\{0\}.$ \qed
Co-r.e. Closed Subsets and Negative Information

Let $A \subseteq \mathbb{R}^n$ be a closed set. Then $A$ is co-r.e., if and only if

$$\{(x, r) \in \mathbb{Q}^n \times \mathbb{Q} : A \cap \overline{B}(x, r) = \emptyset\}$$

is r.e. This corresponds to “negative information” on the set $A$: 
Let $A \subseteq \mathbb{R}^n$ be a closed set. Then $A$ is \textit{co-r.e.}, if and only if
\[
\{(x, r) \in \mathbb{Q}^n \times \mathbb{Q} : A \cap \overline{B}(x, r) = \emptyset\}
\]
is r.e. This corresponds to “negative information” on the set $A$:

**Definition 41** The \textit{upper Fell topology} on $\mathcal{A} := \{A \subseteq \mathbb{R}^n : A \text{ closed}\}$ is generated by the subbase which consists of the sets
\[
\{A \in \mathcal{A} : A \cap K = \emptyset\}
\]
for all compact sets $K \subseteq \mathbb{R}^n$. 

Characterisations of R.e. Closed Subsets

**Theorem 42** For a non-empty closed subset $A \subseteq \mathbb{R}^n$ the following are equivalent:

1. $A$ is r.e. closed, i.e. $d_A : \mathbb{R}^n \to \mathbb{R}$ is upper semi-computable,
2. $\{(x, r) \in \mathbb{Q}^n \times \mathbb{Q} : A \cap B(x, r) \neq \emptyset\}$ is r.e.,
3. $A = \text{range}(f)$ for some computable sequence $f : \mathbb{N} \to \mathbb{R}^n$. 
Characterisations of R.e. Closed Subsets

Theorem 42 For a non-empty closed subset $A \subseteq \mathbb{R}^n$ the following are equivalent:

1. $A$ is r.e. closed, i.e. $d_A : \mathbb{R}^n \to \mathbb{R}$ is upper semi-computable,
2. $\{(x, r) \in \mathbb{Q}^n \times \mathbb{Q} : A \cap B(x, r) \neq \emptyset\}$ is r.e.,
3. $A = \text{range} (f)$ for some computable sequence $f : \mathbb{N} \to \mathbb{R}^n$.

Proof. 1. $\implies$ 2. Follows from $d_A(x) < \varepsilon \iff A \cap B(x, \varepsilon) \neq \emptyset$.

Idea of 2. $\implies$ 3:
Let \( A \subseteq \mathbb{R}^n \) be a closed set. Then \( A \) is r.e., if and only if

\[
\{(x, r) \in \mathbb{Q}^n \times \mathbb{Q} : A \cap B(x, r) \neq \emptyset\}
\]

is r.e. This corresponds to “positive information” on the set \( A \):
Let \( A \subseteq \mathbb{R}^n \) be a closed set. Then \( A \) is r.e., if and only if
\[
\{(x, r) \in \mathbb{Q}^n \times \mathbb{Q} : A \cap B(x, r) \neq \emptyset\}
\]
is r.e. This corresponds to “positive information” on the set \( A \):

**Definition 43** The *lower Fell topology* on \( \mathcal{A} := \{A \subseteq \mathbb{R}^n : A \text{ closed}\} \) is generated by the subbase which consists of the sets
\[
\{A \in \mathcal{A} : A \cap U \neq \emptyset\}
\]
for all open \( U \subseteq \mathbb{R}^n \).
Theorem 44  Let $A, B \subseteq \mathbb{R}^n$ be closed subsets. Then:

1. $A, B$ r.e. closed $\implies A \cup B$ r.e. closed,
2. $A, B$ co-r.e. closed $\implies A \cup B, A \cap B$ co-r.e. closed,
3. $A, B$ recursive closed $\implies A \cup B$ recursive closed.
Closure Properties

Theorem 44 Let $A, B \subseteq \mathbb{R}^n$ be closed subsets. Then:

1. $A, B$ r.e. closed $\implies A \cup B$ r.e. closed,
2. $A, B$ co-r.e. closed $\implies A \cup B, A \cap B$ co-r.e. closed,
3. $A, B$ recursive closed $\implies A \cup B$ recursive closed.

Proof.

1. Follows from 
   \[B(x, \varepsilon) \cap (A \cup B) \neq \emptyset \iff (B(x, \varepsilon) \cap A \neq \emptyset \text{ or } B(x, \varepsilon) \cap B \neq \emptyset).\]

2. $A = \mathbb{R}^n \setminus \bigcup_{i=0}^\infty B(x_i, r_i)$ and $B = \mathbb{R}^n \setminus \bigcup_{i=0}^\infty B(y_i, s_i)$ implies 
   \[A \cap B = \mathbb{R}^n \setminus \big(\bigcup_{i=0}^\infty (B(x_i, r_i) \cup B(y_i, r_i))\big).\]
   If $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ are functions with $A = f^{-1}\{0\}$ and $B = g^{-1}\{0\}$, then $A \cup B = (f \cdot g)^{-1}\{0\}$.

3. Follows from 1. and 2. \qed
A Counterexample

Example 45  There are two recursive closed subsets $A, B \subseteq \mathbb{R}$ such that $A \cap B$ is not r.e. closed.
A Counterexample

Example 45 There are two recursive closed subsets $A, B \subseteq \mathbb{R}$ such that $A \cap B$ is not r.e. closed.

Proof. Let $f : \mathbb{N} \to \mathbb{N}$ be a computable function with some non-recursive image $K := \text{range}(f)$. We define subsets $A_n \subseteq \mathbb{R}$ by

$$A_n := \begin{cases} [n - \frac{1}{2}; n - 2^{-k-2}] & \text{if } k = \min\{i \mid f(i) = n\} \\ [n - \frac{1}{2}; n] & \text{if } n \not\in K \end{cases}$$

Let $A := \bigcup_{n=0}^{\infty} A_n$ and $B := \mathbb{N} \subseteq \mathbb{R}$. Then $A \cap B = \mathbb{N} \setminus K \subseteq \mathbb{R}$ is not r.e. closed. \Box
Examples of Recursive Subsets

Proposition 46

1. The closed sets $\emptyset$ and $\mathbb{R}^n$ are recursive.

2. The closed set $\{x\} \subseteq \mathbb{R}^n$ is recursive, if and only if $x \in \mathbb{R}^n$ is computable.

3. The closed subsets $\{(x, y) : x = y\}, \{(x, y) : x \leq y\} \subseteq \mathbb{R}^2$ are recursive.

4. The closed balls $\overline{B}(x, \varepsilon)$ are recursive, if $x \in \mathbb{R}^n$ and $\varepsilon > 0$ are computable.

5. The closed interval $[a; b]$ is recursive, if and only if $a, b \in \mathbb{R}$ are computable.

6. The closed set $\text{graph}(f) := \{(x, y) \in \mathbb{R}^{n+1} : f(x) = y\}$ is recursive, if and only if $f : \mathbb{R}^n \to \mathbb{R}$ is computable (for any $f \in C(\mathbb{R}^n, \mathbb{R})$).
The Mandelbrot Set

**Open Question:** Is the Mandelbrot set \( M := \{ c \in \mathbb{C} : (\forall n) |f^n_c(0)| < 2 \} \) a recursive closed subset of \( \mathbb{R}^2 \)? (Here \( f_c : \mathbb{C} \to \mathbb{C}, z \mapsto z^2 + c \).)
The Mandelbrot Set

Open Question: Is the Mandelbrot set $M := \{c \in \mathbb{C} : (\forall n) |f^n_c(0)| < 2\}$ a recursive closed subset of $\mathbb{R}^2$? (Here $f_c : \mathbb{C} \to \mathbb{C}, z \mapsto z^2 + c$.)

Theorem 47 (Hertling 2005) If the Hyperbolicity Conjecture holds, then the Mandelbrot set is recursive.
Further Reading on Computable Subsets


Computable Metric Spaces

Definition 48  A tuple \((X, d, \alpha)\) is called a \textit{computable metric space}, if

1. \(d : X \times X \to \mathbb{R}\) is a metric on \(X\),
2. \(\alpha : \mathbb{N} \to X\) is a sequence which is dense in \(X\),
3. \(d \circ (\alpha \times \alpha) : \mathbb{N}^2 \to \mathbb{R}\) is a computable (double) sequence in \(\mathbb{R}\).

Definition 49  Let \((X, d, \alpha)\) be a computable metric space. Then the \textit{Cauchy representation} \(\delta_X : \subseteq \Sigma^\omega \to X\) can be defined by

\[
\delta_X(01^{n_0+1}01^{n_1+1}01^{n_2+1}...) := \lim_{i \to \infty} \alpha(n_i)
\]

for all \(n_i\) such that \((\alpha(n_i))_{i \in \mathbb{N}}\) converges and \(d(\alpha(n_i), \alpha(n_j)) \leq 2^{-i}\) for all \(j > i\) (and undefined for all other input sequences).
Examples of Computable Metric Spaces

Example 50

1. \((\mathbb{R}^n, d_{\mathbb{R}^n}, \alpha_{\mathbb{R}^n})\) with the Euclidean metric

\[ d_{\mathbb{R}^n}(x, y) := \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \]

and some standard enumeration \(\alpha_{\mathbb{R}^n}\) of \(\mathbb{Q}^n\) is a computable metric space.

2. \((C[0, 1], d_C, \alpha_C)\) is a computable metric space with the supremum metric

\[ d_C(f, g) := ||f - g|| := \sup_{x \in [0, 1]} |f(x) - g(x)| \]

and some standard numbering \(\alpha_C\) of \(\mathbb{Q}[x]\). The computable points in this space are exactly the computable functions \(f : [0, 1] \to \mathbb{R}\).

3. \((\mathcal{K}(X), d_\mathcal{K}, \alpha_\mathcal{K})\) with the set \(\mathcal{K}(X)\) of non-empty compact subsets of a computable metric space \((X, d, \alpha)\) and the Hausdorff metric

\[ d_\mathcal{K}(A, B) := \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b) \right\} \]

and some standard numbering \(\alpha_\mathcal{K}\) of the non-empty finite subsets of \(\text{range}(\alpha)\) is a computable metric space. The computable points are exactly the non-empty recursive compact subsets \(A \subseteq X\).
Theorem 51  Let $X$ and $Y$ be computable metric spaces with Cauchy representations $\delta_X$ and $\delta_Y$, respectively. Then a function $f : X \to Y$ is $(\delta_X, \delta_Y)$–continuous, if and only if it continuous in the ordinary sense.
Theorem 51  Let $X$ and $Y$ be computable metric spaces with Cauchy representations $\delta_X$ and $\delta_Y$, respectively. Then a function $f : X \rightarrow Y$ is $(\delta_X, \delta_Y)$–continuous, if and only if it continuous in the ordinary sense.

Proof. This “only if” part can be proved similarly as in the Euclidean case and is based on the fact that $\delta_Y$ is continuous and $\delta_X$ admits an open and surjective restriction.

$\delta_X \quad \delta_Y$

$\uparrow \quad \downarrow$

$X \quad f \quad Y$

The “if” part can be proved using continuity of $\delta_X$ and the fact that $\delta_Y$ admits an open and surjective restriction. \qed
The Intermediate Value Theorem

Theorem 52 For each continuous function $f : [0, 1] \rightarrow \mathbb{R}$ with $f(0) \cdot f(1) < 0$ there exists a point $x \in [0, 1]$ with $f(x) = 0$. 
The Intermediate Value Theorem

Theorem 52: For each continuous function $f : [0, 1] \to \mathbb{R}$ with $f(0) \cdot f(1) < 0$ there exists a point $x \in [0, 1]$ with $f(x) = 0$.

Does there exist a computable version of this theorem? In general, we can distinguish two computable forms of such theorems:

- **non-uniform**: for any suitable computable $f$ there exists a computable $x$,
- **uniform**: given some suitable continuous $f$, we can compute a zero $x$. 

The Intermediate Value Theorem

Theorem 52 For each continuous function \( f : [0, 1] \to \mathbb{R} \) with \( f(0) \cdot f(1) < 0 \) there exists a point \( x \in [0, 1] \) with \( f(x) = 0 \).

Does there exist a computable version of this theorem? In general, we can distinguish two computable forms of such theorems:

- **non-uniform**: for any suitable computable \( f \) there exists a computable \( x \),
- **uniform**: given some suitable continuous \( f \), we can compute a zero \( x \).

In the uniform case we can distinguish two subcases:

- **functional**: there exists a computable (continuous) function \( Z : \subseteq C[0, 1] \to \mathbb{R} \) such that \( Z(f) \) is a zero of \( f \) for all \( f \in F \).
- **multi-valued**: there exist a multi-valued computable (continuous) function \( Z : \subseteq C[0, 1] \Rightarrow \mathbb{R} \) such that for all \( f \in F \) there exists an \( x \in Z(f) \) and all such \( x \) are zeros of \( f \).

Here \( F \subseteq \{ f \in C[0, 1] : f(0) \cdot f(1) < 0 \} \).
Multi-Valued Computable Functions

Definition 53 A multi-valued function \( f : \subseteq X \Rightarrow Y \) is called \((\delta, \delta')\)-computable, if there exists a computable function \( F : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega \) such that

\[ \delta' F(p) \in f\delta(p) \]

for all \( p \in \text{dom}(f\delta) \).
Definition 53 A multi-valued function $f : \subseteq X \Rightarrow Y$ is called $(\delta, \delta')$–computable, if there exists a computable function $F : \subseteq \Sigma^\omega \to \Sigma^\omega$ such that

$$\delta' F(p) \in f \delta(p)$$

for all $p \in \text{dom}(f \delta)$.
Definition 53  A multi-valued function $f : \subseteq X \Rightarrow Y$ is called $(\delta, \delta')$–computable, if there exists a computable function $F : \subseteq \Sigma^\omega \to \Sigma^\omega$ such that

$$\delta' F(p) \in f\delta(p)$$

for all $p \in \text{dom}(f\delta)$.

Analogously, $f$ is called $(\delta, \delta')$–continuous, if there exists a continuous $F$ with the properties described above.
Unsolvability of the General Case

**Theorem 53** There is no general algorithm, which determines a zero for any continuous function \( f : [0, 1] \rightarrow \mathbb{R} \) with \( f(0) \cdot f(1) < 0 \).

(There is not even a corresponding continuous \( Z : \subseteq C[0, 1] \Rightarrow \mathbb{R} \)).
Unsolvability of the General Case

**Theorem 53** There is no general algorithm, which determines a zero for any continuous function $f : [0, 1] \to \mathbb{R}$ with $f(0) \cdot f(1) < 0$.

(There is not even a corresponding continuous $Z : \subseteq C[0, 1] \Rightarrow \mathbb{R}$.)
Unsolvability of the General Case

**Theorem 53** There is no general algorithm, which determines a zero for any continuous function $f : [0, 1] \to \mathbb{R}$ with $f(0) \cdot f(1) < 0$.

(There is not even a corresponding continuous $Z : \subseteq C[0, 1] \Rightarrow \mathbb{R}$.)
Theorem 53  There is no general algorithm, which determines a zero for any continuous function $f : [0, 1] \to \mathbb{R}$ with $f(0) \cdot f(1) < 0$.

(There is not even a corresponding continuous $Z : \subseteq C[0, 1] \Rightarrow \mathbb{R}$.)
**Theorem 54** There exists an algorithm which determines a zero for all given continuous functions $f : [0, 1] \to \mathbb{R}$ with $f(0) \cdot f(1) < 0$, which have exactly one zero.

(There is a corresponding computable $Z : \subseteq C[0, 1] \to \mathbb{R}$.)

**Proof.** Starting with the interval $[0, 1]$ we use the trisection method to determine smaller and smaller intervals which contain the zero:

\[
\begin{array}{cccc}
  a_i & b_i & c_i & d_i \\
\end{array}
\]

Next interval if $f(a_i)f(c_i) < 0$

is determined first

\[
\begin{array}{cccc}
  a_i & b_i & c_i & d_i \\
\end{array}
\]

Next interval if $f(b_i)f(d_i) < 0$

is determined first
Computable Versions of the Intermediate Value Theorem

The following table describes certain sets

\[ F \subseteq \{ f \in C[0, 1] : f(0) \cdot f(1) < 0 \} \]

for which zeros can be determined single-valued or at least multi-valued:

- a green (+) indicates that there exists a computable solution,
- a red (−) indicates that there is not even a continuous solution:
The following table describes certain sets

$$F \subseteq \{ f \in C[0,1] : f(0) \cdot f(1) < 0 \}$$

for which zeros can be determined single-valued or at least multi-valued:

- a green (+) indicates that there exists a computable solution,
- a red (−) indicates that there is not even a continuous solution:

<table>
<thead>
<tr>
<th></th>
<th>$f(0) \cdot f(1) &lt; 0$</th>
<th>$f(0) \cdot f(1) &lt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$f^{-1}{0}$ nowhere dense</td>
</tr>
<tr>
<td>single-valued</td>
<td>−</td>
<td>−</td>
</tr>
</tbody>
</table>
| multi-valued     | −                     | +                     | +

$f$ has exactly one zero
**Computable Versions of the Intermediate Value Theorem**

The following table describes certain sets

\[ F \subseteq \{ f \in C[0, 1] : f(0) \cdot f(1) < 0 \} \]

for which zeros can be determined single-valued or at least multi-valued:

- a green (+) indicates that there exists a computable solution,
- a red (−) indicates that there is not even a continuous solution:

<table>
<thead>
<tr>
<th></th>
<th>( f(0) \cdot f(1) &lt; 0 )</th>
<th>( f(0) \cdot f(1) &lt; 0 ) ( f^{-1}{0} ) nowhere dense</th>
<th>( f ) has exactly one zero</th>
</tr>
</thead>
<tbody>
<tr>
<td>single-valued</td>
<td>–</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>multi-valued</td>
<td>–</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

**Theorem 55 (Non-uniform)** *For any computable function* \( f : [0, 1] \to \mathbb{R} \) with \( f(0) \cdot f(1) < 0 \) *there exists a computable point* \( x \in [0, 1] \) *with* \( f(x) = 0 \).
Brouwer’s Fixed Point Theorem

**Theorem 56** Every continuous function $f : [0, 1]^n \to [0, 1]^n$ admits a fixed point $x$, i.e. $f(x) = x$. 
Brouwer’s Fixed Point Theorem

Theorem 56  Every continuous function \( f : [0, 1]^n \rightarrow [0, 1]^n \) admits a fixed point \( x \), i.e. \( f(x) = x \).

Theorem 57 (Orevkov 1963)  There exists a computable function \( f : [0, 1]^2 \rightarrow [0, 1]^2 \) without computable fixed point.
Brouwer’s Fixed Point Theorem

**Theorem 56** Every continuous function $f : [0, 1]^n \rightarrow [0, 1]^n$ admits a fixed point $x$, i.e. $f(x) = x$.

**Theorem 57 (Orevkov 1963)** There exists a computable function $f : [0, 1]^2 \rightarrow [0, 1]^2$ without computable fixed point.

**Definition 58** A subset $A \subseteq [0, 1]^n$ is called fixable, if there exists a computable function $f : [0, 1]^n \rightarrow [0, 1]^n$ such that $f(A) = A$. 
Brouwer’s Fixed Point Theorem

Theorem 56  Every continuous function \( f : [0, 1]^n \to [0, 1]^n \) admits a fixed point \( x \), i.e. \( f(x) = x \).

Theorem 57 (Orevkov 1963)  There exists a computable function \( f : [0, 1]^2 \to [0, 1]^2 \) without computable fixed point.

Definition 58  A subset \( A \subseteq [0, 1]^n \) is called fixable, if there exists a computable function \( f : [0, 1]^n \to [0, 1]^n \) such that \( f(A) = A \).

Theorem 59 (Miller 2002)  Let \( A \subseteq [0, 1]^n \) be a co-r.e. closed set. Then the following are equivalent:

1. \( A \) is fixable,
2. \( A \) contains a non-empty co-r.e. closed connected component,
3. \( A \) contains a non-empty co-r.e. closed connected subset,
4. \( f(A) \) contains a computable real for all computable \( f : [0, 1]^n \to \mathbb{R} \).
Riemann’s Mapping Theorem

Theorem 60 Let $D := \{z \in \mathbb{C} : |z| < 1\}$ be the open unit disc. For $U \subseteq \mathbb{C}$ the following are equivalent:

1. There exists a holomorphic bijection $f : D \to U$.

2. $U$ is a non-trivial, open, connected, simply connected subset of $\mathbb{C}$.
Riemann’s Mapping Theorem

**Theorem 60** Let $D := \{ z \in \mathbb{C} : |z| < 1 \}$ be the open unit disc. For $U \subseteq \mathbb{C}$ the following are equivalent:

1. There exists a holomorphic bijection $f : D \rightarrow U$.

2. $U$ is a non-trivial, open, connected, simply connected subset of $\mathbb{C}$.

**Theorem 61** (Hertling 1999) For $U \subseteq \mathbb{C}$ the following are equivalent:

1. There exists a computable holomorphic bijection $f : D \rightarrow U$.

2. $U$ is a non-trivial, open, connected, simply connected subset of $\mathbb{C}$ with a co-r.e. closed complement $\mathbb{C} \setminus U$ and an r.e. closed boundary $\partial U$. 
Riemann’s Mapping Theorem

**Theorem 60** Let $D := \{ z \in \mathbb{C} : |z| < 1 \}$ be the open unit disc. For $U \subseteq \mathbb{C}$ the following are equivalent:

1. There exists a holomorphic bijection $f : D \to U$.
2. $U$ is a non-trivial, open, connected, simply connected subset of $\mathbb{C}$.

**Theorem 61 (Hertling 1999)** For $U \subseteq \mathbb{C}$ the following are equivalent:

1. There exists a computable holomorphic bijection $f : D \to U$.
2. $U$ is a non-trivial, open, connected, simply connected subset of $\mathbb{C}$ with a co-r.e. closed complement $\mathbb{C} \setminus U$ and an r.e. closed boundary $\partial U$.

**Remark 62** This result is fully constructive. That is, given a “program” of $f$, we can compute a description of $\mathbb{C} \setminus U$ as a co-r.e. closed set and a description of $\partial U$ as an r.e. closed set and vice versa.
Definition 63 \((X, || \cdot ||, e)\) is called a **computable normed space**, if

1. \( || \cdot || : X \rightarrow \mathbb{R} \) is a norm on \( X \),

2. the linear span of \( e : \mathbb{N} \rightarrow X \) is dense in \( X \),

3. \((X, d, \alpha_e)\) with \( d(x, y) := ||x - y|| \) and

\[
\alpha_e \langle k, \langle n_0, \ldots, n_k \rangle \rangle := \sum_{i=0}^{k} \alpha F(n_i) e_i,
\]

is a computable metric space with Cauchy representation \( \delta_X \).

In this situation \((X, +, \cdot, 0)\) is always a computable vector space with respect to \( \delta_X \).
Theorem 64  Let $X, Y$ be computable normed spaces, let $T : X \to Y$ be a linear operator and let $(e_n)_{n \in \mathbb{N}}$ be a computable sequence in $X$ whose linear span is dense in $X$. Then the following are equivalent:

- $T : X \to Y$ is computable,
- $(T(e_n))_{n \in \mathbb{N}}$ is computable and $T$ is bounded,
- $T$ maps computable sequences to computable sequences and is bounded,
- $\text{graph}(T)$ is a recursive closed subset of $X \times Y$ and $T$ is bounded,
- $\text{graph}(T)$ is an r.e. closed subset of $X \times Y$ and $T$ is bounded.

In case that $X$ and $Y$ are even Banach spaces, one can omit boundedness in the last two cases.
Integration and Differentiation

Theorem 65  We use the norm $\|f\| := \sup_{x \in [0,1]} |f(x)|$ on $C[0,1]$ and the norm $\|f\|_1 := \|f\| + \|f'\|$ on $C^1[0,1]$.

- The integration operator

$$I : C[0,1] \to C[0,1], f \mapsto \left( x \mapsto \int_0^x f(t) \, dt \right)$$

is a linear computable operator.

- The differentiation operator

$$d : C[0,1] \to C[0,1], f \mapsto f'$$

is a linear unbounded operator with a closed graph.

- The differentiation operator

$$D : C^1[0,1] \to C[0,1], f \mapsto f'$$

is a computable linear operator.
Theorem 66 Let $X, Y$ be computable normed spaces, let $T : X \to Y$ be a linear operator and let $(e_n)_{n \in \mathbb{N}}$ be a computable sequence in $X$ whose linear span is dense in $X$. If

- $T$ is unbounded,
- $\text{graph}(T)$ is closed,
- $(T(e_n))_{n \in \mathbb{N}}$ is a computable sequence in $Y$,

then there exists a computable $x \in X$ such that $T(x)$ is non-computable.
Theorem of Pour-El and Richards

Theorem 66  Let $X,Y$ be computable normed spaces, let $T : X \to Y$ be a linear operator and let $(e_n)_{n \in \mathbb{N}}$ be a computable sequence in $X$ whose linear span is dense in $X$. If

- $T$ is unbounded,
- $\text{graph}(T)$ is closed,
- $(T(e_n))_{n \in \mathbb{N}}$ is a computable sequence in $Y$,

then there exists a computable $x \in X$ such that $T(x)$ is non-computable.

Corollary 67 (Myhill 1971)  There exists a computable function $f : [0, 1] \to \mathbb{R}$ with a continuous derivative $f' : [0, 1] \to \mathbb{R}$ which is not computable.
The Wave Equation

Theorem 68 (Pour-El and Richards) There exists a computable initial condition $f$ and a three-dimensional wave $u$ such that $x \mapsto u(0, x)$ is a computable function, but the unique solution of the wave equation

\[
\begin{cases}
  u_{tt} = \Delta u \\
  u(0, x) = f, \quad u_t(0, x) = 0, \quad t \in \mathbb{R}, \quad x \in \mathbb{R}^3
\end{cases}
\]

at time 1 leads to a non-computable $x \mapsto u(1, x)$. 
The Wave Equation

Theorem 68 (Pour-El and Richards) There exists a computable initial condition $f$ and a three-dimensional wave $u$ such that $x \mapsto u(0, x)$ is a computable function, but the unique solution of the wave equation

\[
\begin{aligned}
&u_{tt} = \Delta u \\
&u(0, x) = f, \quad u_t(0, x) = 0, \quad t \in \mathbb{R}, \quad x \in \mathbb{R}^3
\end{aligned}
\]

at time 1 leads to a non-computable $x \mapsto u(1, x)$.

Theorem 69 (Weihrauch and Zhong 2002) The solution operator of the wave equation $S : C^k(\mathbb{R}^3) \to C^{k-1}(\mathbb{R}^4), f \mapsto u$ is computable.
The Wave Equation

**Theorem 68 (Pour-El and Richards)** There exists a computable initial condition $f$ and a three-dimensional wave $u$ such that $x \mapsto u(0, x)$ is a computable function, but the unique solution of the wave equation

$$
\begin{aligned}
    u_{tt} &= \Delta u \\
    u(0, x) &= f, \\n    u_t(0, x) &= 0, \
\end{aligned}
\quad t \in \mathbb{R}, x \in \mathbb{R}^3
$$

at time 1 leads to a non-computable $x \mapsto u(1, x)$.

**Theorem 69 (Weihrauch and Zhong 2002)** The solution operator of the wave equation $S : C^k(\mathbb{R}^3) \to C^{k-1}(\mathbb{R}^4), f \mapsto u$ is computable.

**Remark 70** Weihrauch and Zhong proved that the operator of wave propagation is computable without any loss of the degree of differentiability, if physically appropriate Sobolev spaces are used.
The Hahn-Banach Theorem

**Theorem 71 (Metakides, Nerode and Shore 1985)** There exists a computable Banach space $X$ and an effectively separable closed linear subspace $Y \subseteq X$ as well as a computable linear functional $f : Y \to \mathbb{R}$ such that there is no computable extension $g : X \to \mathbb{R}$ with $\|g\| = \|f\|$. 
The Hahn-Banach Theorem

Theorem 71 (Metakides, Nerode and Shore 1985) There exists a computable Banach space $X$ and an effectively separable closed linear subspace $Y \subseteq X$ as well as a computable linear functional $f : Y \to \mathbb{R}$ such that there is no computable extension $g : X \to \mathbb{R}$ with $\|g\| = \|f\|$. 

Theorem 72 (Metakides and Nerode 1982) Let $X$ be a finite-dimensional computable Banach space with some effectively separable closed linear subspace $Y \subseteq X$. For any computable linear functional $f : Y \to \mathbb{R}$ with computable norm $\|f\|$ there exists a computable linear extension $g : X \to \mathbb{R}$ with $\|g\| = \|f\|$. 

The Hahn-Banach Theorem

Theorem 71 (Metakides, Nerode and Shore 1985) There exists a computable Banach space $X$ and an effectively separable closed linear subspace $Y \subseteq X$ as well as a computable linear functional $f : Y \to \mathbb{R}$ such that there is no computable extension $g : X \to \mathbb{R}$ with $\|g\| = \|f\|$.

Theorem 72 (Metakides and Nerode 1982) Let $X$ be a finite-dimensional computable Banach space with some effectively separable closed linear subspace $Y \subseteq X$. For any computable linear functional $f : Y \to \mathbb{R}$ with computable norm $\|f\|$ there exists a computable linear extension $g : X \to \mathbb{R}$ with $\|g\| = \|f\|$.

Theorem 73 Let $X$ be a computable normed space with a strictly convex dual $X^*$ and let $Y \subseteq X$ be an effectively separable closed linear subspace. For any computable linear functional $f : Y \to \mathbb{R}$ with computable norm $\|f\|$ there exists exactly one linear extension $g : X \to \mathbb{R}$ with $\|g\| = \|f\|$ and this extension is computable.
Further Reading on Computable Metric Spaces


Admissible Representations

Definition 74 A representation $\delta$ of a topological space $X$ is called \textit{admissible}, if $\delta$ is continuous and $\delta' \leq_t \delta$ holds for any other continuous representation $\delta'$ of $X$. 
Admissible Representations

Definition 74  A representation $\delta$ of a topological space $X$ is called \textit{admissible}, if $\delta$ is continuous and $\delta' \leq_t \delta$ holds for any other continuous representation $\delta'$ of $X$.

Theorem 75 (Kreitz-Weihrauch and Schröder)  Let $(X, \delta_X)$ and $(Y, \delta_Y)$ be admissibly represented topological spaces. Then a function $f : \subseteq X \to Y$ is $(\delta_X, \delta_Y)$–continuous, if and only if it is sequentially continuous.

\[
\begin{array}{c}
\Sigma^\omega \xrightarrow{F} \Sigma^\omega \\
\downarrow \delta_X \quad \downarrow \delta_Y \\
X \xrightarrow{f} Y
\end{array}
\]
Admissible Representations

Definition 74 A representation $\delta$ of a topological space $X$ is called admissible, if $\delta$ is continuous and $\delta' \leq_t \delta$ holds for any other continuous representation $\delta'$ of $X$.

Theorem 75 (Kreitz-Weihrauch and Schröder) Let $(X, \delta_X)$ and $(Y, \delta_Y)$ be admissibly represented topological spaces. Then a function $f : \subseteq X \to Y$ is $(\delta_X, \delta_Y)$–continuous, if and only if it is sequentially continuous.

Example 76 The Cauchy representation $\delta_X$ of a computable metric space $X$ is admissible with respect to the metric topology.
Theorem 77  For all represented spaces \((X, \delta_X), (Y, \delta_Y)\) there exists a representation \([\delta_X, \delta_Y]\) of \(X \times Y\) and a representation \([\delta_X \to \delta_Y]\) of the set of \((\delta_X, \delta_Y)\)–continuous functions \(f : X \to Y\) which have the following properties:
Theorem 77  For all represented spaces \((X, \delta_X), (Y, \delta_Y)\) there exists a representation \([\delta_X, \delta_Y]\) of \(X \times Y\) and a representation \([\delta_X \to \delta_Y]\) of the set of \((\delta_X, \delta_Y)\)–continuous functions \(f : X \to Y\) which have the following properties:

- **evaluation**: the map

  \[
  (f, x) \mapsto f(x)
  \]

  is \(([[\delta_X \to \delta_Y], \delta_X], \delta_Y)\)–computable,

- **type conversion**: any function \(f : Z \times X \to Y\) is \(([[\delta_Z, \delta_X], \delta_Y])\)–computable, if and only if the map

  \[
  z \mapsto (x \mapsto f(z, x))
  \]

  is \((\delta_Z, [\delta_X \to \delta_Y])\)–computable,

for all represented spaces \((X, \delta_X), (Y, \delta_Y)\) and \((Z, \delta_Z)\).
The Category of Admissibly Represented Spaces

Theorem 78 (Schröder) The category of admissibly represented sequential $T_0$-spaces is cartesian closed.
Theorem 78 (Schröder) The category of admissibly represented sequential $T_0$–spaces is cartesian closed.
Some Hyperspace Representations

**Definition 79** Let \((X, d, \alpha)\) be a computable metric space. We define representations of \(\mathcal{A}(X) := \{A \subseteq X : A \text{ closed and non-empty}\}:

1. \(\psi_<(p) = A \iff p \text{ is a “list” of all } \langle n, k \rangle \text{ with } A \cap B(\alpha(n), k) \neq \emptyset\),

2. \(\psi_(p) = A \iff p \text{ is a “list” of } \langle n_i, k_i \rangle \text{ with } X \setminus A = \bigcup_{i=0}^{\infty} B(\alpha(n_i), k_i)\),

3. \(\psi_=(p, q) = A \iff \psi_<(p) = A \text{ and } \psi_>(q) = A\),

for all \(p, q \in \Sigma^\omega\) and \(A \in \mathcal{A}(X)\). We indicate by a corresponding index with which representation \(\mathcal{A}(X)\) is endowed.
Some Hyperspace Representations

Definition 79 Let \((X, d, \alpha)\) be a computable metric space. We define representations of \(\mathcal{A}(X) := \{A \subseteq X : A \text{ closed and non-empty}\}\):

1. \(\psi_<(p) = A : \iff p \) is a “list” of all \(\langle n, k \rangle\) with \(A \cap B(\alpha(n), \bar{k}) \neq \emptyset\),

2. \(\psi_>(p) = A : \iff p \) is a “list” of \(\langle n_i, k_i \rangle\) with \(X \setminus A = \bigcup_{i=0}^{\infty} B(\alpha(n_i), \bar{k_i})\),

3. \(\psi_=\langle p, q \rangle = A : \iff \psi_<(p) = A \) and \(\psi_>(q) = A\),

for all \(p, q \in \Sigma^\omega\) and \(A \in \mathcal{A}(X)\). We indicate by a corresponding index with which representation \(\mathcal{A}(X)\) is endowed.

Lemma 80

1. \(\cup : \mathcal{A}_<(X) \times \mathcal{A}_<(X) \to \mathcal{A}_<(X), (A, B) \mapsto A \cup B \) is computable,

2. \(\cup : \mathcal{A}_=(X) \times \mathcal{A}_=(X) \to \mathcal{A}_=(X), (A, B) \mapsto A \cup B \) is computable,

3. \(\cap : \mathcal{A}_>(X) \times \mathcal{A}_>(X) \to \mathcal{A}_>(X), (A, B) \mapsto A \cap B \) is computable.
Some Admissible Representations

Example 81

- The representations $\psi$ of $A(\mathbb{R}^n)$ are admissible with respect to the Fell topology.

- The representations $\psi_<$ and $\psi_>$ of $A(\mathbb{R}^n)$ are admissible with respect to the lower and upper Fell topology, respectively.

- The representations $\rho_<$ and $\rho_>$ are admissible with respect to the lower and upper Euclidean topology, respectively. (These topologies are generated by the intervals $(q, \infty)$ and $(-\infty, q)$, respectively.)

- The representation $[\rho^n \to \rho]$ of $C(\mathbb{R}^n)$ is admissible with respect to the compact-open topology.

- The representations $[\rho^n \to \rho_<]$ of $LSC(\mathbb{R}^n)$ and $[\rho^n \to \rho_>]$ of $USC(\mathbb{R}^n)$ are admissible with respect to certain compact-open topologies.
Theorem 82 (Ziegler, B.) The dimension map $\dim : \subseteq \mathcal{A}(\mathbb{R}^n) \rightarrow \mathbb{R}$ (defined on linear subspaces) is $(\psi_\prec, \rho_\prec)$–, $(\psi_\succ, \rho_\succ)$– and $(\psi, \rho)$–computable.
Theorem 82 (Ziegler, B.) The dimension map \( \dim : \subseteq A(\mathbb{R}^n) \to \mathbb{R} \) (defined on linear subspaces) is \((\psi_\prec, \rho_\prec)\)-, \((\psi_\succ, \rho_\succ)\)- and \((\psi, \rho)\)-computable.

Theorem 83 (Ziegler, B.) Let \( A \in \mathbb{R}^{m \times n} \) be a computable matrix and \( y \in \mathbb{R}^m \) a computable vector. Then the corresponding solution space \( L := \{ x \in \mathbb{R}^n : Ax = y \} \) is recursive. In particular, if there is a unique solution \( x \in \mathbb{R}^n \), then it is computable.
Theorem 82 (Ziegler, B.) The dimension map \( \dim : \subseteq A(\mathbb{R}^n) \to \mathbb{R} \) (defined on linear subspaces) is \((\psi_\prec, \rho_\prec)–, (\psi_\succ, \rho_\succ)–\) and \((\psi, \rho)–\)computable.

Theorem 83 (Ziegler, B.) Let \( A \in \mathbb{R}^{m \times n} \) be a computable matrix and \( y \in \mathbb{R}^m \) a computable vector. Then the corresponding solution space \( L := \{x \in \mathbb{R}^n : Ax = y\} \) is recursive. In particular, if there is a unique solution \( x \in \mathbb{R}^n \), then it is computable.

Remark 84

- The above result does not hold uniformly! Given a matrix \( A \) and the vector \( y \), one cannot continuously determine a solution \( x \) in general.
- However, upon additional input of the dimension of the solution space, one obtains a uniform result.
- This explains, why there is no numerical algorithm to solve linear equations in the general case (where several solutions exist).
Theorem 85  Let \( X \) be a complete computable metric space. For any computable sequence \( (A_n)_{n \in \mathbb{N}} \) of co-r.e. closed nowhere dense subsets \( A_n \subseteq X \), there exists some computable point \( x \in X \setminus \bigcup_{n=0}^{\infty} A_n \).
Theorem 85 Let $X$ be a complete computable metric space. For any computable sequence $(A_n)_{n \in \mathbb{N}}$ of co-r.e. closed nowhere dense subsets $A_n \subseteq X$, there exists some computable point $x \in X \setminus \bigcup_{n=0}^{\infty} A_n$.

Remark 86 This result holds uniform: there is a computable multi-valued map $\subseteq A_{>}(\mathbb{R}^n) \Rightarrow X$, which, given the sequence $(A_n)_{n \in \mathbb{N}}$ with respect to $\psi_{>}$, computes a suitable point $x$. 

Vasco Brattka Department of Mathematics & Applied Mathematics · University of Cape Town
A Computable Baire Category Theorem

**Theorem 85** Let $X$ be a complete computable metric space. For any computable sequence $(A_n)_{n \in \mathbb{N}}$ of co-r.e. closed nowhere dense subsets $A_n \subseteq X$, there exists some computable point $x \in X \setminus \bigcup_{n=0}^{\infty} A_n$.

**Remark 86** This result holds uniform: there is a computable multi-valued map $\subseteq A_>(\mathbb{R}^n) \Rightarrow X$, which, given the sequence $(A_n)_{n \in \mathbb{N}}$ with respect to $\psi_>$, computes a suitable point $x$.

**Corollary 87** If $X$ is a computable complete metric space without isolated points, then there exists no computable sequence $(y_n)_{n \in \mathbb{N}}$ such that $\{y_n : n \in \mathbb{N}\}$ is the set of computable points of $X$. 
Theorem 85 Let $X$ be a complete computable metric space. For any computable sequence $(A_n)_{n \in \mathbb{N}}$ of co-r.e. closed nowhere dense subsets $A_n \subseteq X$, there exists some computable point $x \in X \setminus \bigcup_{n=0}^{\infty} A_n$.

Remark 86 This result holds uniform: there is a computable multi-valued map $\subseteq A_>(\mathbb{R}^n) \Rightarrow X$, which, given the sequence $(A_n)_{n \in \mathbb{N}}$ with respect to $\psi_>$, computes a suitable point $x$.

Corollary 87 If $X$ is a computable complete metric space without isolated points, then there exists no computable sequence $(y_n)_{n \in \mathbb{N}}$ such that $\{y_n : n \in \mathbb{N}\}$ is the set of computable points of $X$.

Corollary 88 There exists a computable but nowhere differentiable function $f : [0, 1] \to \mathbb{R}$.
Banach’s Inverse Mapping Theorem

**Theorem 89** Let $X, Y$ be Banach spaces and let $T : X \to Y$ be a linear operator. If $T$ is bijective and bounded, then $T^{-1} : Y \to X$ is bounded.
Banach’s Inverse Mapping Theorem

**Theorem 89** Let $X, Y$ be Banach spaces and let $T : X \to Y$ be a linear operator. If $T$ is bijective and bounded, then $T^{-1} : Y \to X$ is bounded.

**Question:** Which of the following statements hold true?

1. **Non-uniform inversion problem:**
   
   $T$ computable $\Rightarrow T^{-1}$ computable,

2. **Uniform inversion problem:**
   
   $T \mapsto T^{-1}$ computable.

In order to formalize the second property, one naturally uses the representations $[\delta_X \to \delta_Y]$ and $[\delta_Y \to \delta_X]$. 
Banach’s Inverse Mapping Theorem

Theorem 90  Let $X, Y$ be Banach spaces and let $T : X \to Y$ be a linear operator. If $T$ is bijective and bounded, then $T^{-1} : Y \to X$ is bounded.

Question: Which of the following statements hold true?

1. Non-uniform inversion problem: 
   $T$ computable $\implies T^{-1}$ computable,      Yes!

2. Uniform inversion problem: 
   $T \mapsto T^{-1}$ computable.                   No!

In order to formalize the second property, one naturally uses the representations $[\delta_X \to \delta_Y]$ and $[\delta_Y \to \delta_X]$.

Answer: Only the non-uniform property holds true. There is a counterexample for the uniform version.
Theorem 91  Let $f_0, \ldots, f_n : [0, 1] \to \mathbb{R}$ be computable with $f_n \neq 0$. The solution operator $L : C[0, 1] \times \mathbb{R}^n \to C^{(n)}[0, 1]$ which maps each $(y, a_0, \ldots, a_{n-1}) \in C[0, 1] \times \mathbb{R}^n$ to the unique $x \in C^{(n)}[0, 1]$ such that

$$\sum_{i=0}^{n} f_i(t)x^{(i)}(t) = y(t) \text{ with } x^{(j)}(0) = a_j \text{ for } j = 0, \ldots, n-1,$$

is computable.

Proof. The following operator is linear and computable:

$$L^{-1} : C^{(n)}[0, 1] \to C[0, 1] \times \mathbb{R}^n, x \mapsto \left(\sum_{i=0}^{n} f_ix^{(i)}, x^{(0)}(0), \ldots, x^{(n-1)}(0)\right)$$

By the computable Inverse Mapping Theorem it follows that $L$ is computable too. \qed
An Initial Value Problem

**Theorem 91** Let \( f_0, \ldots, f_n : [0, 1] \to \mathbb{R} \) be computable with \( f_n \neq 0 \). The solution operator \( L : C[0, 1] \times \mathbb{R}^n \to C^{(n)}[0, 1] \) which maps each \((y, a_0, \ldots, a_{n-1}) \in C[0, 1] \times \mathbb{R}^n\) to the unique \( x \in C^{(n)}[0, 1] \) such that
\[
\sum_{i=0}^{n} f_i(t)x^{(i)}(t) = y(t) \quad \text{with} \quad x^{(j)}(0) = a_j \quad \text{for} \quad j = 0, \ldots, n - 1,
\]
is computable.

**Proof.** The following operator is linear and computable:
\[
L^{-1} : C^{(n)}[0, 1] \to C[0, 1] \times \mathbb{R}^n, x \mapsto \left( \sum_{i=0}^{n} f_ix^{(i)}, x^{(0)}(0), \ldots, x^{(n-1)}(0) \right)
\]
By the computable Inverse Mapping Theorem it follows that \( L \) is computable too.  

**Remark 92** Here a non-uniform result has a uniform consequence (since it is applied to function spaces). This is a very general and highly non-constructive method to prove the existence of algorithms.
Further Reading on Computable Analysis


Schedule

A. Computable Real Numbers and Functions
   1. Background
   2. Computability Notions for Real Numbers and Real Functions
   3. Representations, Computability and Continuity

B. Computable Subsets
   4. Computability Notions for Subsets and Functions
   5. Computable Metric Spaces and Computable Functional Analysis
   6. Theory of Admissible Representations

C. Complexity
   7. Computational Complexity on Real Numbers
   8. Discrete Complexity Classes and Continuous Problems
   9. Degrees of Unsolvability
Time Complexity on Infinite Sequences

The time complexity of a Turing machine $M$ computing a function $f : \subseteq \Sigma^\omega \to \Sigma^\omega$ should describe the asymptotic behaviour of $M$. 
The \textit{time complexity} of a Turing machine $M$ computing a function $f : \subseteq \Sigma^\omega \to \Sigma^\omega$ should describe the asymptotic behaviour of $M$. 

\[ p(0) \ p(1) \ p(2) \ p(3) \ p(4) \ p(5) \ p(6) \ p(7) \ p(8) \ p(9) \ \ldots \ \text{input lookahead} \]

\[ q(0) \ q(1) \ q(2) \ q(3) \ q(4) \ q(5) \ q(6) \ q(7) \ q(8) \ q(9) \ \ldots \ \text{output precision} \]
The \textit{time complexity} of a Turing machine $M$ computing a function $f : \subseteq \Sigma^\omega \rightarrow \Sigma^\omega$ should describe the asymptotic behaviour of $M$.

It is reasonable to measure the number of time steps in dependency of

- the \textit{output precision} and
- the \textit{size of the input},
- while keeping track of the \textit{input lookahead}.
For any real number $x \in \mathbb{R}$ there exists a name $p \in \Sigma^\omega$ which is computable in linear times and such that $\rho_C(p) = x$.

Idea: padding method.
Time Complexity of Real Numbers

- For any real number \( x \in \mathbb{R} \) there exists a name \( p \in \Sigma^\omega \) which is computable in linear times and such that \( \rho_C(p) = x \).
  Idea: padding method.

- Thus, it makes no sense to define the complexity of \( x \in \mathbb{R} \) as the minimal complexity of all \( p \in \rho_C^{-1}\{x\} \).
Time Complexity of Real Numbers

- For any real number $x \in \mathbb{R}$ there exists a name $p \in \Sigma^\omega$ which is computable in linear times and such that $\rho_C(p) = x$.
  Idea: padding method.

- Thus, it makes no sense to define the complexity of $x \in \mathbb{R}$ as the minimal complexity of all $p \in \rho_C^{-1}\{x\}$.

- This is because $\rho_C$ is too redundant, i.e. the fibers $\rho_C^{-1}\{x\}$ are too large.
Time Complexity of Real Numbers

• For any real number $x \in \mathbb{R}$ there exists a name $p \in \Sigma^\omega$ which is computable in linear times and such that $\rho_C(p) = x$. 
  Idea: padding method.

• Thus, it makes no sense to define the complexity of $x \in \mathbb{R}$ as the minimal complexity of all $p \in \rho_C^{-1}\{x\}$.

• This is because $\rho_C$ is too redundant, i.e. the fibers $\rho_C^{-1}\{x\}$ are too large.

• A condition which is sufficient to guarantee the existence of the above minimum is compactness!
Time Complexity of Real Numbers

- For any real number $x \in \mathbb{R}$ there exists a name $p \in \Sigma^\omega$ which is computable in linear times and such that $\rho_C(p) = x$. 
  Idea: padding method.

- Thus, it makes no sense to define the complexity of $x \in \mathbb{R}$ as the minimal complexity of all $p \in \rho_C^{-1}\{x\}$.

- This is because $\rho_C$ is too redundant, i.e. the fibers $\rho_C^{-1}\{x\}$ are too large.

- A condition which is sufficient to guarantee the existence of the above minimum is compactness!

- A proper representation $\delta$ is a representation such that $\delta^{-1}(K)$ is compact for any compact $K$. 
Time Complexity of Real Numbers

• For any real number \( x \in \mathbb{R} \) there exists a name \( p \in \Sigma^\omega \) which is computable in linear times and such that \( \rho_C(p) = x \).
  Idea: padding method.

• Thus, it makes no sense to define the complexity of \( x \in \mathbb{R} \) as the minimal complexity of all \( p \in \rho_C^{-1}\{x\} \).

• This is because \( \rho_C \) is too redundant, i.e. the fibers \( \rho_C^{-1}\{x\} \) are too large.

• A condition which is sufficient to guarantee the existence of the above minimum is compactness!

• A proper representation \( \delta \) is a representation such that \( \delta^{-1}(K) \) is compact for any compact \( K \).

• Question: which spaces admit proper admissible representations?
**Signed Digit Representation**

**Definition 93** The *signed digit representation* $\rho_{sd} : \subseteq \sum^\omega \to \mathbb{R}$ is defined by

$$\rho_{sd}(a_n a_{n-1} \ldots a_0 \cdot a_{-1} a_{-2} \ldots) := \sum_{i=n}^{-\infty} a_i \cdot 2^i$$

for all sequences with $a_i \in \{1, 0, 1\}$ und $n \geq -1$ (with the additional properties that $a_n \neq 0$, if $n \geq 0$ and $a_n a_{n-1} \not\in \{11, \overline{11}\}$, if $n \geq 1$).
Signed Digit Representation

Definition 93  The signed digit representation $\rho_{sd} : \subseteq \Sigma^\omega \to \mathbb{R}$ is defined by

$$\rho_{sd}(a_n a_{n-1} \ldots a_0 \cdot a_{-1} a_{-2} \ldots) := \sum_{i=n}^{-\infty} a_i \cdot 2^i$$

for all sequences with $a_i \in \{1, 0, 1\}$ and $n \geq -1$ (with the additional properties that $a_n \neq 0$, if $n \geq 0$ and $a_n a_{n-1} \not\in \{11, 1\}$, if $n \geq 1$).

Remark 94  In contrast to the binary representation, the signed digit representation is redundant in a symmetric way:
Signed Digit Representation

Theorem 95  The signed digit representation is proper and admissible.
Signed Digit Representation

**Theorem 95** *The signed digit representation is proper and admissible.*

The path \( p = .\overline{101} \ldots \) and the corresponding intervals \( \rho_{sd}(p \leq_k \Sigma^\omega) \).
Definition 96 A Turing machine $M$ computes a real number function $f : \mathbb{R} \to \mathbb{R}$ on a domain $K \subseteq \mathbb{R}$ in time $t : \mathbb{N} \to \mathbb{N}$ with input lookahead $l : \mathbb{N} \to \mathbb{N}$, if and only if for any $p \in \rho_{sd}^{-1}(K)$:

1. $\rho_{sd} f_M(p) = f \rho_{sd}(p)$, i.e. $M$ computes $f$ upon input $p$,
2. with at most $c \cdot t(n) + c$ time steps,
3. and with reading at most $l(n)$ input symbols,

in order to produce the $n$–th output symbol.
Time Complexity of Real Number Functions

Definition 96 A Turing machine $M$ computes a real number function $f : \mathbb{R} \to \mathbb{R}$ on a domain $K \subseteq \mathbb{R}$ in time $t : \mathbb{N} \to \mathbb{N}$ with input lookahead $l : \mathbb{N} \to \mathbb{N}$, if and only if for any $p \in \rho_{sd}^{-1}(K)$:

1. $\rho_{sd} f_M(p) = f \rho_{sd}(p)$, i.e. $M$ computes $f$ upon input $p$,
2. with at most $c \cdot t(n) + c$ time steps,
3. and with reading at most $l(n)$ input symbols,

in order to produce the $n$–th output symbol.

Lemma 97 For computable $f : \mathbb{R} \to \mathbb{R}$ and any co-r.e. compact $K \subseteq \mathbb{R}$ a computable time bound $t$ and a computable input lookahead $l$ exist.
Time Complexity of Real Number Functions

Definition 96 A Turing machine $M$ computes a real number function $f : \mathbb{R} \to \mathbb{R}$ on a domain $K \subseteq \mathbb{R}$ in time $t : \mathbb{N} \to \mathbb{N}$ with input lookahead $l : \mathbb{N} \to \mathbb{N}$, if and only if for any $p \in \rho^{-1}(K)$:

1. $\rho_{sd} f_M(p) = f_{\rho_{sd}}(p)$, i.e. $M$ computes $f$ upon input $p$,
2. with at most $c \cdot t(n) + c$ time steps,
3. and with reading at most $l(n)$ input symbols,

in order to produce the $n$–th output symbol.

Lemma 97 For computable $f : \mathbb{R} \to \mathbb{R}$ and any co-r.e. compact $K \subseteq \mathbb{R}$ a computable time bound $t$ and a computable input lookahead $l$ exist.

Remark 98 For $K_\sigma$ domains one can use the “size” of the domain as a further parameter, e.g. for the domain $\mathbb{R} = \bigcup_{i=0}^{\infty} [-i, i]$ one can measure time complexity and input lookahead in dependency of the additional parameter $i$. 

Vasco Brattka Department of Mathematics & Applied Mathematics · University of Cape Town
Time Complexity of Arithmetic Functions

Theorem 99  For each of the following functions $f : \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$ and domains $K \subseteq \mathbb{R}^m$ there exists a Turing machine $M$ which computes $f$ in time $t$ with input lookahead $l$: 
Time Complexity of Arithmetic Functions

**Theorem 99** For each of the following functions $f : \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$ and domains $K \subseteq \mathbb{R}^m$ there exists a Turing machine $M$ which computes $f$ in time $t$ with input lookahead $l$:

<table>
<thead>
<tr>
<th>function</th>
<th>domain</th>
<th>time</th>
<th>lookahead</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-x_1$</td>
<td>$\mathbb{R}$</td>
<td>$n$</td>
<td>$n$</td>
</tr>
<tr>
<td>$x_1 + x_2$</td>
<td>$[-1, 1] \times [-1, 1]$</td>
<td>$n$</td>
<td>$n + c$</td>
</tr>
<tr>
<td>$x_1 \cdot x_2$</td>
<td>$[-1, 1] \times [-1, 1]$</td>
<td>$n \cdot \log n \cdot \log \log n$</td>
<td>$2n + c$</td>
</tr>
<tr>
<td>$x_1 \cdot x_2$</td>
<td>$[-1, 1] \times [-1, 1]$</td>
<td>$n \cdot \log^2 n \cdot \log \log n$</td>
<td>$n + c$</td>
</tr>
<tr>
<td>$1/x_1$</td>
<td>$[7/8, 2]$</td>
<td>$n \cdot \log n \cdot \log \log n$</td>
<td>$2n + c$</td>
</tr>
<tr>
<td>$\exp, \sin, \cos$</td>
<td>$[-1, 1]$</td>
<td>$n \cdot \log^2 n \cdot \log \log n$</td>
<td>$2n + c$</td>
</tr>
</tbody>
</table>
Theorem 99 For each of the following functions \( f : \subseteq \mathbb{R}^m \rightarrow \mathbb{R} \) and domains \( K \subseteq \mathbb{R}^m \) there exists a Turing machine \( M \) which computes \( f \) in time \( t \) with input lookahead \( l \):

<table>
<thead>
<tr>
<th>function</th>
<th>domain</th>
<th>time</th>
<th>lookahead</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-x_1)</td>
<td>(\mathbb{R})</td>
<td>(n)</td>
<td>(n)</td>
</tr>
<tr>
<td>(x_1 + x_2)</td>
<td>([-1,1] \times [-1,1])</td>
<td>(n)</td>
<td>(n + c)</td>
</tr>
<tr>
<td>(x_1 \cdot x_2)</td>
<td>([-1,1] \times [-1,1])</td>
<td>(n \cdot \log n \cdot \log \log n)</td>
<td>(2n + c)</td>
</tr>
<tr>
<td>(1/x_1)</td>
<td>([7/8,2])</td>
<td>(n \cdot \log n \cdot \log \log n)</td>
<td>(n + c)</td>
</tr>
<tr>
<td>(\exp, \sin, \cos)</td>
<td>([-1,1])</td>
<td>(n \cdot \log^2 n \cdot \log \log n)</td>
<td>(2n + c)</td>
</tr>
</tbody>
</table>

Remark 100 Computations with lookahead \( l(n) = n + c \) are called online computations. They might require a higher time complexity as computations which use a larger input lookahead.
Trade-Off Between Time and Lookahead

**Theorem 101 (Weihrauch 1991)** There exists a real number function $f : \mathbb{R} \to \mathbb{R}$ which can be computed on domain $I = [0, 1]$

1. with input lookahead $l(n) \leq n + c$ with some $c$,

2. with input lookahead $l(n) \leq n + c$ with some $c$ in polynomial time $t(n)$, if and only if $P = NP$,

3. with input lookahead $l(n) \leq c \cdot n$ with some $c$ in linear time $t(n)$. 
Trade-Off Between Time and Lookahead

Theorem 101 (Weihrauch 1991) There exists a real number function $f : \mathbb{R} \to \mathbb{R}$ which can be computed on domain $I = [0, 1]$

1. with input lookahead $l(n) \leq n + c$ with some $c$,
2. with input lookahead $l(n) \leq n + c$ with some $c$ in polynomial time $t(n)$, if and only if $P = NP$,
3. with input lookahead $l(n) \leq c \cdot n$ with some $c$ in linear time $t(n)$.

Remark 102

- If the input is itself the result of some (expensive) computation, it is desirable to use as little input lookahead as possible.
- However, saving input lookahead might increase the computation time substantially.
- In case of a composition of functions one has to balance input lookahead and time complexity in the appropriate way.
Complexity of the Derivative

Definition 103  A time bound $t : \mathbb{N} \rightarrow \mathbb{N}$ is called \textit{regular}, if it is non-decreasing and there exist constants $N, c > 0$ such that

$$2 \cdot t(n) \leq t(2n) \leq c \cdot t(n)$$

for all $n > N$ and $t(N) > 0$. 
Complexity of the Derivative

Definition 103 A time bound $t : \mathbb{N} \to \mathbb{N}$ is called regular, if it is non-decreasing and there exist constants $N, c > 0$ such that

$$2 \cdot t(n) \leq t(2n) \leq c \cdot t(n)$$

for all $n > N$ and $t(N) > 0$.

Theorem 104 (Müller 1986) Let $t$ be a regular time bound with $n \cdot \log n \cdot \log \log n \in O(t)$. If $f : [0, 1] \to \mathbb{R}$ is a function which is computable in time $t$ and which has a continuous second derivative, then the derivative $f'$ is also computable in time $t$. 
Complexity of the Derivative

Definition 103 A time bound \( t : \mathbb{N} \rightarrow \mathbb{N} \) is called regular, if it is non-decreasing and there exist constants \( N, c > 0 \) such that

\[
2 \cdot t(n) \leq t(2n) \leq c \cdot t(n)
\]

for all \( n > N \) and \( t(N) > 0 \).

Theorem 104 (Müller 1986) Let \( t \) be a regular time bound with \( n \cdot \log n \cdot \log \log n \in \mathcal{O}(t) \). If \( f : [0, 1] \rightarrow \mathbb{R} \) is a function which is computable in time \( t \) and which has a continuous second derivative, then the derivative \( f' \) is also computable in time \( t \).

Theorem 105 (Müller 1986) Let \( t \) be a regular time bound. The class of real numbers computable in time \( t \) is a real closed field.
Theorem 106 (Schröder 1995) Let $X$ be a $T_0$–space with countable base. Then the following are equivalent:

1. $X$ has an admissible representation $\delta$ which is proper (i.e. such that $\delta^{-1}(K)$ is compact whenever $K \subseteq X$ is compact),

2. $X$ is metrisable.

(Here compactness is defined without Hausdorff condition.)
Theorem 106 (Schröder 1995) Let $X$ be a $T_0$–space with countable base. Then the following are equivalent:

1. $X$ has an admissible representation $\delta$ which is proper (i.e. such that $\delta^{-1}(K)$ is compact whenever $K \subseteq X$ is compact),

2. $X$ is metrisable.

(Here compactness is defined without Hausdorff condition.)

Moreover, the following conditions are equivalent:

1. $X$ has an admissible representation $\delta$ which is fiber compact (i.e. such that $\delta^{-1}\{x\}$ is compact for all $x \in X$),

2. $X$ has the property that any open set is a $F_\sigma$ set.
Further Reading on Complexity of Real Number Functions

Complexity of Numerical Operators

- It is difficult to define a uniform notion of complexity for operators of type

\[ F : C[0, 1] \to C[0, 1] \]

such as integration or differentiation.
Complexity of Numerical Operators

• It is difficult to define a uniform notion of complexity for operators of type

\[ F : C[0, 1] \rightarrow C[0, 1] \]

such as integration or differentiation.

• Typical function spaces such as \( C[0, 1] \) are not locally compact and therefore there is no obvious way to define a uniform notion of complexity (not even parameterised over \( \mathbb{N} \)).
Complexity of Numerical Operators

• It is difficult to define a uniform notion of complexity for operators of type
  \[ F : C[0, 1] \rightarrow C[0, 1] \]
  such as integration or differentiation.

• Typical function spaces such as \( C[0, 1] \) are not locally compact and therefore there is no obvious way to define a uniform notion of complexity (not even parameterised over \( \mathbb{N} \)).

• However, one can study the complexity of such operators restricted to compact subspaces \( K \subseteq C[0, 1] \), which is often done in numerical analysis.
Complexity of Numerical Operators

- It is difficult to define a uniform notion of complexity for operators of type

\[ F : C[0, 1] \rightarrow C[0, 1] \]

such as integration or differentiation.

- Typical function spaces such as \( C[0, 1] \) are not locally compact and therefore there is no obvious way to define a uniform notion of complexity (not even parameterised over \( \mathbb{N} \)).

- However, one can study the complexity of such operators restricted to compact subspaces \( K \subseteq C[0, 1] \), which is often done in numerical analysis.

- Alternatively, one can study the time complexity of \( F(f) \) for polynomial-time computable \( f \).
Theorem 107 (Friedman 1984) The following are equivalent:

1. \( P = NP \).

2. For each polynomial-time computable \( f : [0, 1] \to \mathbb{R} \) the maximum function \( g : [0, 1] \to \mathbb{R} \), defined by

\[
g(x) := \max\{ f(y) : 0 \leq y \leq x \}
\]

for all \( x \in [0, 1] \), is polynomial-time computable.
Maximisation

Theorem 107 (Friedman 1984)  The following are equivalent:

1. $P = NP$. 

2. For each polynomial-time computable $f : [0,1] \rightarrow \mathbb{R}$ the maximum function $g : [0,1] \rightarrow \mathbb{R}$, defined by

$$g(x) := \max\{f(y) : 0 \leq y \leq x\}$$

for all $x \in [0,1]$, is polynomial-time computable.

Proof. The direction “1.$\implies$2.” of the proof is based on the idea that

$$z \leq g(x) \iff (\exists y \in [0,x]) \ z \leq f(y).$$

Using the Polynomial-Time Projection Theorem this is (approximately) decidable in polynomial time, if $P = NP$. By a binary search over $z$ one can determine $g(x)$ in polynomial time. \hfill \Box
Theorem 108 (Friedman, Ko 1986) *The following are equivalent:*

1. $\text{FP} = \#\text{P}$.

2. *For each polynomial-time computable $f : [0, 1] \to \mathbb{R}$ the integral function $g : [0, 1] \to \mathbb{R}$, defined by*

$$g(x) := \int_0^x f(t) \, dt$$

*for all $x \in [0, 1]$, is polynomial-time computable.*
Integration

Theorem 108 (Friedman, Ko 1986)  The following are equivalent:

1. $FP = \#P$.

2. For each polynomial-time computable $f : [0, 1] \rightarrow \mathbb{R}$ the integral function $g : [0, 1] \rightarrow \mathbb{R}$, defined by

$$g(x) := \int_0^x f(t) \, dt$$

for all $x \in [0, 1]$, is polynomial-time computable.

Here $\#P$ denotes the class of functions that count the number of accepting computations of a nondeterministic polynomial-time Turing machine.
Integration

Theorem 108 (Friedman, Ko 1986) The following are equivalent:

1. $FP = \#P$.

2. For each polynomial-time computable $f : [0, 1] \to \mathbb{R}$ the integral function $g : [0, 1] \to \mathbb{R}$, defined by

$$g(x) := \int_0^x f(t) \, dt$$

for all $x \in [0, 1]$, is polynomial-time computable.

Here $\#P$ denotes the class of functions that count the number of accepting computations of a nondeterministic polynomial-time Turing machine.

For the proof of “1.$\implies$2.” one can guess a number of points $(t, y)$ with $0 \leq t \leq x$ and then count those with $y \leq f(t)$ to get an approximation for the integral $g(x)$ (in case that $f$ is positive).
Ordinary Differentiable Equations

Theorem 109 (Ko 1983) The following holds:

\[ P = \text{PSPACE}. \]

\[ \implies \text{For each polynomial-time computable } f : \mathbb{R} \to \mathbb{R} \text{ which satisfies the Lipschitz condition} \]

\[ |f(x, z_1) - f(x, z_2)| \leq L \cdot |z_1 - z_2| \]

\[ \text{for some } L > 0, \text{ the unique solution } y : [0, 1] \to \mathbb{R} \text{ of the differential equation} \]

\[ y'(x) = f(x, y(x)), \quad y(0) = 0 \]

\[ \implies \text{FP = #P}. \]
Discrete Complexity Classes and Numerical Problems

The following table gives some rough classifications of numerical problems with respect to complexity classes (due to Ker-I Ko):

<table>
<thead>
<tr>
<th>Numerical Problem</th>
<th>Complexity Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimisation (functions)</td>
<td>EXPSPACE–complete</td>
</tr>
<tr>
<td>Integral Equations</td>
<td>between PSPACE and EXPSPACE</td>
</tr>
<tr>
<td>Ordinary Differential Equation</td>
<td>PSPACE–complete</td>
</tr>
<tr>
<td>Integration</td>
<td>#P–complete</td>
</tr>
<tr>
<td>Optimisation (numbers)</td>
<td>NP–complete</td>
</tr>
<tr>
<td>Roots (2-dim)</td>
<td>between UP and NP</td>
</tr>
<tr>
<td>Roots (1-dim)</td>
<td>P–complete</td>
</tr>
<tr>
<td>Differentiation</td>
<td>P (if computable)</td>
</tr>
</tbody>
</table>
The following table gives some rough classifications of numerical problems with respect to complexity classes (due to Ker-I Ko):

<table>
<thead>
<tr>
<th>Numerical Problem</th>
<th>Complexity Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimisation (functions)</td>
<td>EXPSPACE–complete</td>
</tr>
<tr>
<td>Integral Equations</td>
<td>between PSPACE and EXPSPACE</td>
</tr>
<tr>
<td>Ordinary Differential Equation</td>
<td>PSPACE–complete</td>
</tr>
<tr>
<td>Integration</td>
<td>#P–complete</td>
</tr>
<tr>
<td>Optimisation (numbers)</td>
<td>NP–complete</td>
</tr>
<tr>
<td>Roots (2-dim)</td>
<td>between UP and NP</td>
</tr>
<tr>
<td>Roots (1-dim)</td>
<td>P–complete</td>
</tr>
<tr>
<td>Differentiation</td>
<td>P (if computable)</td>
</tr>
</tbody>
</table>

UP denotes the class of sets computable in polynomial time by nondeterministic Turing machines that have at most one accepting path.
Further Reading on Complexity in Analysis


Complexity of Compact Subsets in Euclidean Space

Definition 110  A compact set $K \subseteq \mathbb{R}^2$ is called (locally) computable in time $t : \mathbb{N} \rightarrow \mathbb{N}$, if there exists function $f : (\{0, 1\}^*)^3 \rightarrow \{0, 1\}^*$ computable in time $t$ such that

$$f(0^n, \text{bin}(i), \text{bin}(j)) = \begin{cases} 
1 & \text{if } B((\frac{i}{2^{n+2}}, \frac{j}{2^{n+2}}), \frac{1}{2^{n+2}}) \cap K \neq \emptyset \\
0 & \text{if } \overline{B}((\frac{i}{2^{n+2}}, \frac{j}{2^{n+2}}), \frac{1}{2^{n+1}}) \cap K = \emptyset \\
0 \text{ or } 1 & \text{otherwise}
\end{cases}$$
Definition 110 A compact set $K \subseteq \mathbb{R}^2$ is called (locally) computable in time $t : \mathbb{N} \rightarrow \mathbb{N}$, if there exists function $f : (\{0, 1\}^*)^3 \rightarrow \{0, 1\}^*$ computable in time $t$ such that

$$f(0^n, \text{bin}(i), \text{bin}(j)) = \begin{cases} 
1 & \text{if } B((\frac{i}{2^n+2}, \frac{j}{2^n+2}), \frac{1}{2^n+2}) \cap K \neq \emptyset \\
0 & \text{if } \overline{B}((\frac{i}{2^n+2}, \frac{j}{2^n+2}), \frac{1}{2^n+1}) \cap K = \emptyset \\
0 \text{ or } 1 & \text{otherwise}
\end{cases}$$
Complexity of Compact Subsets in Euclidean Space

**Definition 110** A compact set $K \subseteq \mathbb{R}^2$ is called (locally) computable in time $t : \mathbb{N} \to \mathbb{N}$, if there exists function $f : (\{0, 1\}^\ast)^3 \to \{0, 1\}^\ast$ computable in time $t$ such that

$$f(0^n, \text{bin}(i), \text{bin}(j)) = \begin{cases} 1 & \text{if } B((\frac{i}{2^{n+2}}, \frac{j}{2^{n+2}}), \frac{1}{2^{n+2}}) \cap K \neq \emptyset \\ 0 & \text{if } \overline{B}((\frac{i}{2^{n+2}}, \frac{j}{2^{n+2}}), \frac{1}{2^{n+1}}) \cap K = \emptyset \\ 0 \text{ or } 1 & \text{otherwise} \end{cases}$$

**Remark 111** *The (local) complexity of a set measures the time to set an arbitrary pixel (in order to get a good approximation of the set).*
Complexity of the Distance Function of Compact Subsets

Theorem 112 (Braverman 2004) Let $K \subseteq \mathbb{R}^n$ be a compact subset.

1. If the distance function $d_K : \mathbb{R}^n \to \mathbb{R}$ of $K$ is polynomial-time computable, then $K$ is polynomial-time computable.

2. If $n = 1$, then the converse holds.

3. If $n > 1$, then the converse holds, if and only if $P = NP$. 
Complexity of the Distance Function of Compact Subsets

Theorem 112 (Braverman 2004) Let $K \subseteq \mathbb{R}^n$ be a compact subset.

1. If the distance function $d_K : \mathbb{R}^n \to \mathbb{R}$ of $K$ is polynomial-time computable, then $K$ is polynomial-time computable.

2. If $n = 1$, then the converse holds.

3. If $n > 1$, then the converse holds, if and only if $P = \text{NP}$.

Remark 113 The (local) time complexity is a more practical complexity measure. Given a screen of print area of fixed resolution, say with $k$ pixels, it requires $O(k \cdot t(n))$ time to print a set $K \subseteq \mathbb{R}^2$ with (local) time complexity $t$ and a zoom factor $2^n$. 
Definition 114 Let \( f : \mathbb{C} \to \mathbb{C} \) be a polynomial function of degree \( \geq 2 \).

- A point \( w \in \mathbb{C} \) is called a periodic point of \( f \), if there exists a \( p \in \mathbb{N} \) such that \( f^p(w) = w \).
- A periodic point \( w \in \mathbb{C} \) is called repelling, if \( |(f^p)'(w)| > 1 \).
- The Julia set \( J(f) \) of \( f \) is the closure of the set of repelling periodic points of \( f \).
- A point \( z \in \mathbb{C} \) is called critical, if \( f'(z) = 0 \).
- The function \( f \) is called hyperbolic, if the closure of the orbits \( \bigcup_z \text{critical} \bigcup_{n=0}^\infty f^n(z) \) is disjoint to \( J(f) \).
Julia Sets

Definition 114 Let $f : \mathbb{C} \rightarrow \mathbb{C}$ be a polynomial function of degree $\geq 2$.

- A point $w \in \mathbb{C}$ is called a periodic point of $f$, if there exists a $p \in \mathbb{N}$ such that $f^p(w) = w$.

- A periodic point $w \in \mathbb{C}$ is called repelling, if $|f^p(w)| > 1$.

- The Julia set $J(f)$ of $f$ is the closure of the set of repelling periodic points of $f$.

- A point $z \in \mathbb{C}$ is called critical, if $f'(z) = 0$.

- The function $f$ is called hyperbolic, if the closure of the orbits
  $\bigcup_{z \text{ critical}} \bigcup_{n=0}^{\infty} f^n(z)$ is disjoint to $J(f)$.

Theorem 115 (Zhong 1998) The Julia sets $J(f)$ of hyperbolic polynomials are recursive.
Complexity of Some Julia Sets

Theorem 116 (Rettinger and Weihrauch 2003) Let $c \in \mathbb{C}$ be such that $|c| < \frac{1}{4}$. Then the Julia set $J$ of the function $z \mapsto z^2 + c$ can be computed locally in time $O(n^2 \cdot M(n)) \subseteq O(n^3 \cdot \log n \cdot \log \log n)$. 
Complexity of Some Julia Sets

**Theorem 116 (Rettinger and Weihrauch 2003)** Let $c \in \mathbb{C}$ be such that $|c| < \frac{1}{4}$. Then the Julia set $J$ of the function $z \mapsto z^2 + c$ can be computed locally in time $O(n^2 \cdot M(n)) \subseteq O(n^3 \cdot \log n \cdot \log \log n)$. 
Complexity of Some Julia Sets

**Theorem 116 (Rettinger and Weihrauch 2003)** Let $c \in \mathbb{C}$ be such that $|c| < \frac{1}{4}$. Then the Julia set $J$ of the function $z \mapsto z^2 + c$ can be computed locally in time $O(n^2 \cdot M(n)) \subseteq O(n^3 \cdot \log n \cdot \log \log n)$.

**Remark 117** In 2004 this result has independently been generalized to hyperbolic polynomial and rational functions $f$ by Braverman and Rettinger with the time bound $O(n \cdot M(n)) \subseteq O(n^2 \cdot \log n \cdot \log \log n)$. 
Further Reading on Complexity of Sets


Turing Degrees on Metric Spaces

Let $X$ be a computable metric space with Cauchy representation $\delta$. It is natural to define the \textit{Turing degree} of $x \in X$ by

$$\deg_T(x) := \min\{\deg_T(p) : p \in \delta^{-1}\{x\}\}$$
Turing Degrees on Metric Spaces

Let \( X \) be a computable metric space with Cauchy representation \( \delta \). It is natural to define the Turing degree of \( x \in X \) by

\[
\deg_T(x) := \min\{\deg_T(p) : p \in \delta^{-1}\{x\}\}
\]

**Proposition 118** For \( x \in \mathbb{R} =: X \) the Turing degree \( \deg_T(x) \) is well-defined and coincides with the Turing degree of the binary expansion of \( x \).
Turing Degrees on Metric Spaces

Let $X$ be a computable metric space with Cauchy representation $\delta$. It is natural to define the Turing degree of $x \in X$ by

$$\deg_T(x) := \min \{\deg_T(p) : p \in \delta^{-1}\{x\}\}$$

**Proposition 118** For $x \in \mathbb{R} =: X$ the Turing degree $\deg_T(x)$ is well-defined and coincides with the Turing degree of the binary expansion of $x$.

**Theorem 119 (Miller 2004)** There exists some $f \in C[0,1] =: X$ such that $\deg_T(f)$ is not well-defined (i.e. there is no Cauchy name with a minimal Turing degree for $f$).
Definition 120 Let $X$ and $Y$ be computable metric spaces with points $x \in X, y \in Y$. Then $x$ is called \textit{representation reducible} to $y$, in symbols $x \leq_r y$, if and only if there exists a computable function $f : \subseteq Y \rightarrow X$ such that $f(y) = x$. 
Definition 120 Let $X$ and $Y$ be computable metric spaces with points $x \in X, y \in Y$. Then $x$ is called *representation reducible* to $y$, in symbols $x \leq_r y$, if and only if there exists a computable function $f : \subseteq Y \rightarrow X$ such that $f(y) = x$.

Proposition 121 *Every continuous degree contains a real-analytic function.*
Continuous Degrees (Miller 2004)

Definition 120  Let $X$ and $Y$ be computable metric spaces with points $x \in X, y \in Y$. Then $x$ is called representation reducible to $y$, in symbols $x \leq_r y$, if and only if there exists a computable function $f : \subseteq Y \rightarrow X$ such that $f(y) = x$.

Proposition 121  Every continuous degree contains a real-analytic function.

Theorem 122  The Turing degrees can be embedded into the continuous degrees and the continuous degrees can be embedded into the enumeration degrees.
Borel Hierarchy and Arithmetical Hierarchy

- $\Sigma^0_1(X)$ is the set of open subsets of $X$,
- $\Pi^0_1(X)$ is the set of closed subsets of $X$,
- $\Sigma^0_2(X)$ is the set of $F_\sigma$ subsets of $X$,
- $\Pi^0_2(X)$ is the set of $G_\delta$ subsets of $X$, etc.
- $\Delta^0_k(X) := \Sigma^0_k(X) \cap \Pi^0_k(X)$.

By $\Sigma^0_k, \Pi^0_k, \Delta^0_k$ we denote the classes of the arithmetical hierarchy.
Borel Hierarchy and Arithmetical Hierarchy

- \( \Sigma^0_1(X) \) is the set of open subsets of \( X \),
- \( \Pi^0_1(X) \) is the set of closed subsets of \( X \),
- \( \Sigma^0_2(X) \) is the set of \( F_\sigma \) subsets of \( X \),
- \( \Pi^0_2(X) \) is the set of \( G_\delta \) subsets of \( X \), etc.
- \( \Delta^0_k(X) := \Sigma^0_k(X) \cap \Pi^0_k(X) \).

By \( \Sigma^0_k, \Pi^0_k, \Delta^0_k \) we denote the classes of the arithmetical hierarchy.
Definition 123 Let $X, Y$ be separable metric spaces. A function $f : X \rightarrow Y$ is called $\Sigma^0_k$–measurable, if $f^{-1}(U) \in \Sigma^0_k(X)$ for any $U \in \Sigma^0_1(Y)$ and effectively $\Sigma^0_k$–measurable or $\Sigma^0_k$–computable, if the map $U \mapsto f^{-1}(U)$ is computable (w.r.t. suitable representations).
Borel Measurable Operations

Definition 123 Let $X, Y$ be separable metric spaces. A function $f : X \to Y$ is called $\Sigma^0_k$–measurable, if $f^{-1}(U) \in \Sigma^0_k(X)$ for any $U \in \Sigma^0_1(Y)$ and effectively $\Sigma^0_k$–measurable or $\Sigma^0_k$–computable, if the map $U \mapsto f^{-1}(U)$ is computable (w.r.t. suitable representations).

Theorem 124 Let $X, Y$ be computable metric spaces, $k \geq 1$ and let $f : X \to Y$ be a total function. Then the following are equivalent:

1. $f$ is (effectively) $\Sigma^0_k$–measurable,

2. $f$ admits an (effectively) $\Sigma^0_k$–measurable realizer $F : \subseteq \mathbb{N}^\mathbb{N} \to \mathbb{N}^\mathbb{N}$.
Reducibility of Functions

Definition 125 Let $X, Y, U, V$ be computable metric spaces and consider functions $f : \subseteq X \to Y$ and $g : \subseteq U \to V$. We say that

- $f$ is **reducible** to $g$, for short $f \leq_t g$, if there are continuous functions $A : \subseteq X \times V \to Y$ and $B : \subseteq X \to U$ such that
  \[
  f(x) = A(x, g \circ B(x))
  \]
  for all $x \in \text{dom}(f)$,

- $f$ is **computably reducible** to $g$, for short $f \leq_c g$, if there are computable $A, B$ as above.

- The corresponding equivalences are denoted by $\cong_t$ and $\cong_c$. 

Reducibility of Functions

Definition 125  Let $X, Y, U, V$ be computable metric spaces and consider functions $f : \subseteq X \to Y$ and $g : \subseteq U \to V$. We say that

- $f$ is reducible to $g$, for short $f \leq_t g$, if there are continuous functions $A : \subseteq X \times V \to Y$ and $B : \subseteq X \to U$ such that
  $$f(x) = A(x, g \circ B(x))$$
  for all $x \in \text{dom}(f)$,

- $f$ is computably reducible to $g$, for short $f \leq_c g$, if there are computable $A, B$ as above.

- The corresponding equivalences are denoted by $\approx_t$ and $\approx_c$.

Proposition 126  There exist complete functions $C_k$ in the classes of $\Sigma^0_{k+1}$-measurable functions on Baire space. This notion of completeness can be transferred to other computable metric spaces.
Theorem 127. Let $X,Y$ be computable Banach spaces and let $f : \subseteq X \to Y$ be a closed linear and unbounded operator. Let $(e_n)_{n \in \mathbb{N}}$ be a computable sequence in $\text{dom}(f)$ whose linear span is dense in $X$ and let $(f(e_n))_{n \in \mathbb{N}}$ be computable in $Y$. Then $C_1 \leq_c f$. 

Borel-Measurability for Linear Operators
Borel-Measurability for Linear Operators

**Theorem 127** Let $X, Y$ be computable Banach spaces and let $f : \subseteq X \to Y$ be a closed linear and unbounded operator. Let $(e_n)_{n \in \mathbb{N}}$ be a computable sequence in $\text{dom}(f)$ whose linear span is dense in $X$ and let $(f(e_n))_{n \in \mathbb{N}}$ be computable in $Y$. Then $C_1 \leq_c f$.

**Proposition 128** The operator of differentiation

$$d : \subseteq C[0, 1] \to C[0, 1], f \mapsto f'$$

is $\Sigma^0_2$–complete.
Definition 129 A numbering $F$ of the computable functions $f : \mathbb{R}^n \to \mathbb{R}$ can be defined by $F_i := \delta(\varphi_i)$ for all $i \in \mathbb{N}$, where $\delta$ is some suitable representation of $C(\mathbb{R}^n)$ and $\varphi$ is some Gödel numbering of the computable functions $f : \mathbb{N} \to \mathbb{N}$. Let $I^n := \text{dom}(F)$. 
Index Sets of Differential Equations

Definition 129  A numbering $F$ of the computable functions $f : \mathbb{R}^n \to \mathbb{R}$ can be defined by $F_i := \delta(\varphi_i)$ for all $i \in \mathbb{N}$, where $\delta$ is some suitable representation of $C(\mathbb{R}^n)$ and $\varphi$ is some Gödel numbering of the computable functions $f : \mathbb{N} \to \mathbb{N}$. Let $I^n := \text{dom}(F)$.

Theorem 130 (Cenzer and Remmel 2004)  The set of those $i \in I^2$ such that the differential equation $y'(x) = F_i(x, y(x))$ with initial condition $y(0) = 0$ has a computable solution $y$ on some interval $[-\delta, \delta]$ for $\delta > 0$, is $\Sigma^0_3$–complete.
Index Sets of Differential Equations

Definition 129 A numbering $F$ of the computable functions $f : \mathbb{R}^n \to \mathbb{R}$ can be defined by $F_i := \delta(\varphi_i)$ for all $i \in \mathbb{N}$, where $\delta$ is some suitable representation of $C(\mathbb{R}^n)$ and $\varphi$ is some Gödel numbering of the computable functions $f : \mathbb{N} \to \mathbb{N}$. Let $I^n := \text{dom}(F)$.

Theorem 130 (Cenzer and Remmel 2004) The set of those $i \in I^2$ such that the differential equation $y'(x) = F_i(x, y(x))$ with initial condition $y(0) = 0$ has a computable solution $y$ on some interval $[-\delta, \delta]$ for $\delta > 0$, is $\Sigma^0_3$–complete.

Theorem 131 (Cenzer and Remmel 2004) The set of those $i \in I^3$ such that the wave equation

$$u_{xx} + u_{yy} + u_{zz} - u_{tt} = 0$$

with initial condition $u_t(x, y, z, 0) = 0$ and $u(x, y, z, 0) = F_i(x, y, z)$ has a computable solution is $\Sigma^0_3$–complete.
Further Reading on Degrees of Non-Computability


CCA Webpage http://cca-net.de

Computability & Complexity in Analysis
NETWORK

CCA 2006
International Conference

- Introduction
- Members
- Mailing List
- Bibliography
- Publications
- Events
- Links

Gainesville, Florida, USA
November 1-5, 2006

http://cca-net.de