Computability of system properties

Pieter Collins

Centrum voor Wiskunde en Informatica, Amsterdam

Pieter.Collins@cwi.nl

DyToComp, 5-10 June 2006

Acknowledgement This work was partially supported by the Nederlandse Wetenschappelijk Organisatie (NWO) through VIDI project number 639.032.408.

Motivation

- What is it possible to compute about a dynamical system?

Motivation

- What is it possible to compute about a dynamical system?
- What is it *impossible* to compute??

Motivation

- What is it possible to compute about a dynamical system?
- What is it *impossible* to compute??
- What should we use as the *semantics* of a valid computation?

Various approaches

- Brouwer's intuitionalist logic. [W.P. van Stigt, *Brouwer's Intuitionism* (1990)]
- Markov's constructive logic and analysis.
- Bishop's constructive analysis. [Bishop & Bridges, *Constructive analysis* (1985)]

Various approaches

- Brouwer's intuitionalist logic. [W.P. van Stigt, *Brouwer's Intuitionism* (1990)]
- Markov's constructive logic and analysis.
- Bishop's constructive analysis. [Bishop & Bridges, *Constructive analysis* (1985)]
- Scott's domain theory.

Various approaches

- Brouwer's intuitionalist logic. [W.P. van Stigt, *Brouwer's Intuitionism* (1990)]
- Markov's constructive logic and analysis.
- Bishop's constructive analysis. [Bishop & Bridges, *Constructive analysis* (1985)]
- Scott's domain theory.
- Blum, Cucker, Shub and Smale's real-RAM model. [Blum, Cucker, Shub, Smale, *Complexity and real computation* (1998)]

Various approaches

- Brouwer's intuitionalist logic. [W.P. van Stigt, *Brouwer's Intuitionism* (1990)]
- Markov's constructive logic and analysis.
- Bishop's constructive analysis. [Bishop & Bridges, *Constructive analysis* (1985)]
- Scott's domain theory.
- Blum, Cucker, Shub and Smale's real-RAM model. *Unrealistic!* [Blum, Cucker, Shub, Smale, *Complexity and real computation* (1998)]

Various approaches

- Brouwer's intuitionalist logic. [W.P. van Stigt, *Brouwer's Intuitionism* (1990)]
- Markov's constructive logic and analysis.
- Bishop's constructive analysis. [Bishop & Bridges, *Constructive analysis* (1985)]
- Scott's domain theory.
- Blum, Cucker, Shub and Smale's real-RAM model. *Unrealistic!* [Blum, Cucker, Shub, Smale, *Complexity and real computation* (1998)]
- Ko's oracle machines. [Ko, *Complexity theory of real functions* (1991).]
- Weihrauch's computable analysis. [Klaus Weihrauch, *Computable analysis - An introduction* (2000).]

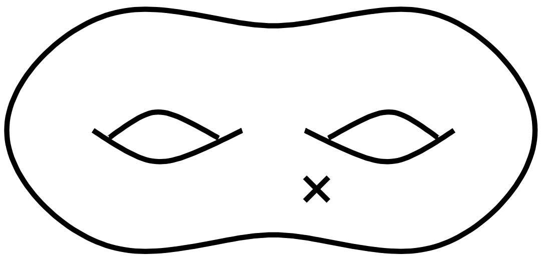
General topology (and philosophy)

A question...

What do the following have in common?

A question...

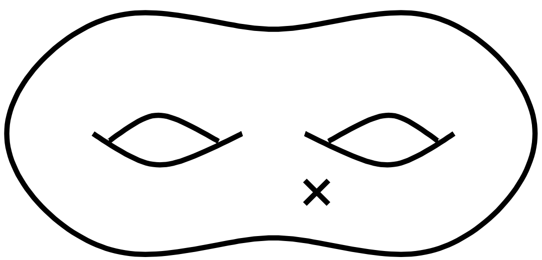
What do the following have in common?



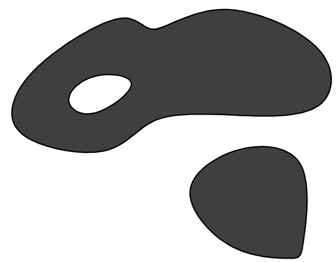
M

A question...

What do the following have in common?



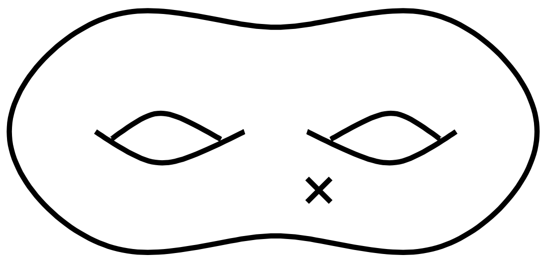
M



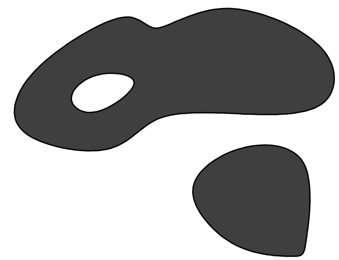
$\mathcal{K}(\mathbb{R}^n)$

A question...

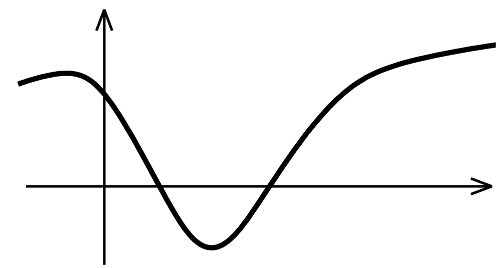
What do the following have in common?



M



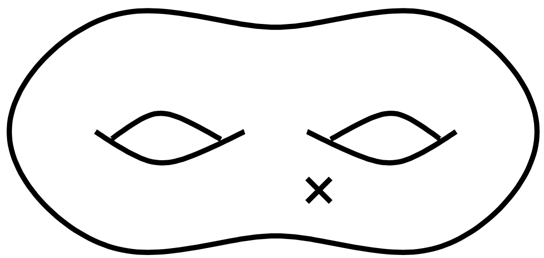
$\mathcal{K}(\mathbb{R}^n)$



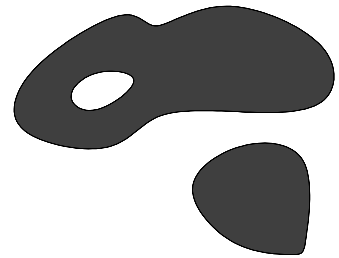
$\mathrm{C}(\mathbb{R} \to \mathbb{R})$

A question...

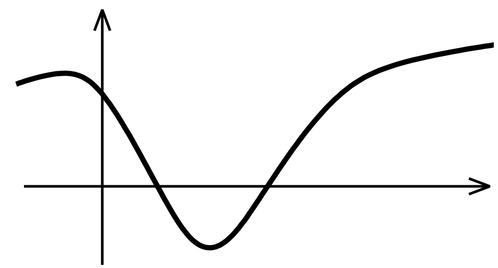
What do the following have in common?



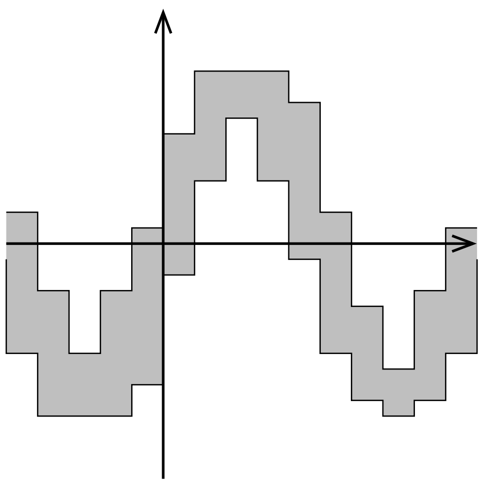
M



$\mathcal{K}(\mathbb{R}^n)$



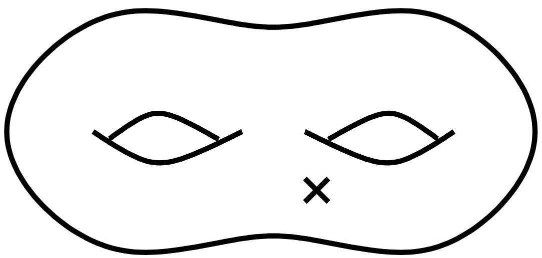
$\mathrm{C}(\mathbb{R} \to \mathbb{R})$



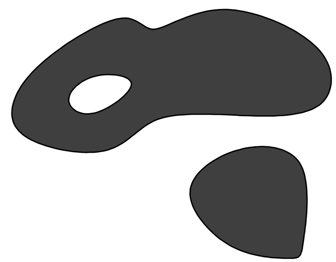
$\mathrm{USC}(\mathbb{R} \rightrightarrows \mathbb{R})$

A question...

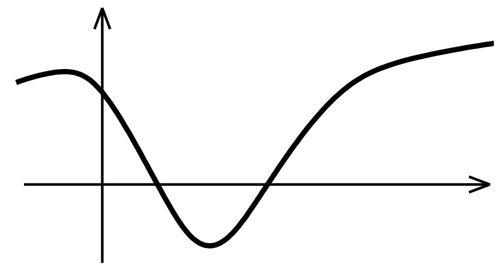
What do the following have in common?



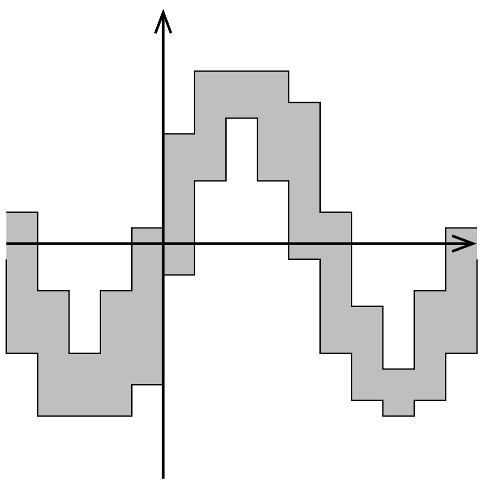
M



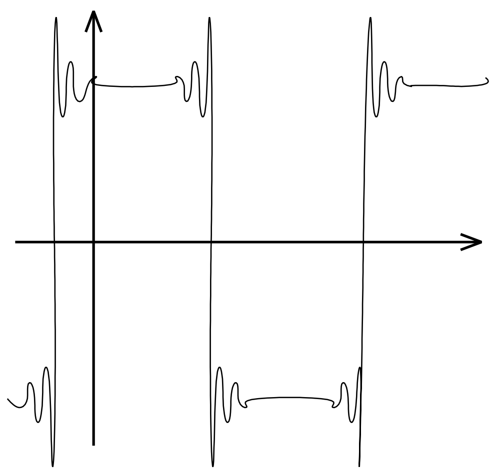
$\mathcal{K}(\mathbb{R}^n)$



$\mathrm{C}(\mathbb{R} \to \mathbb{R})$



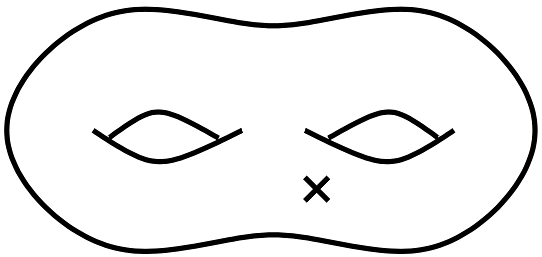
$\mathrm{USC}(\mathbb{R} \rightrightarrows \mathbb{R})$



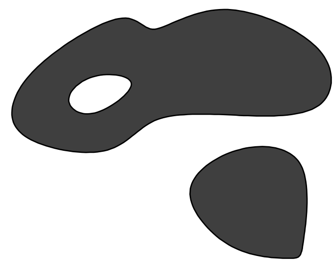
$L^2(\mathbb{R})$

A question...

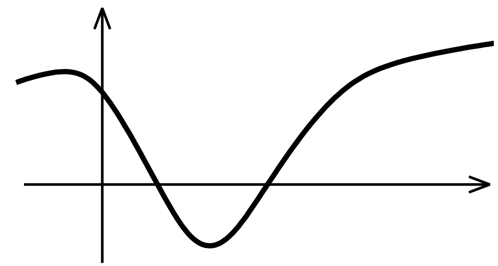
What do the following have in common?



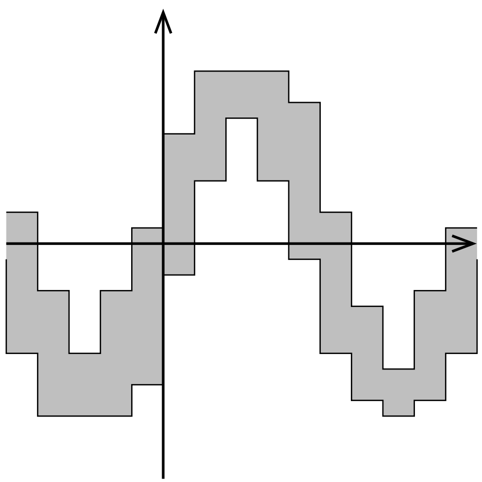
M



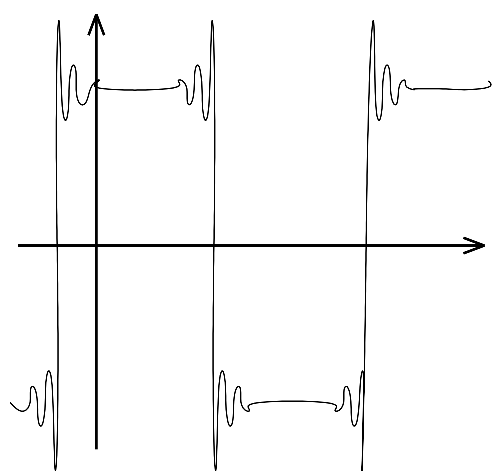
$\mathcal{K}(\mathbb{R}^n)$



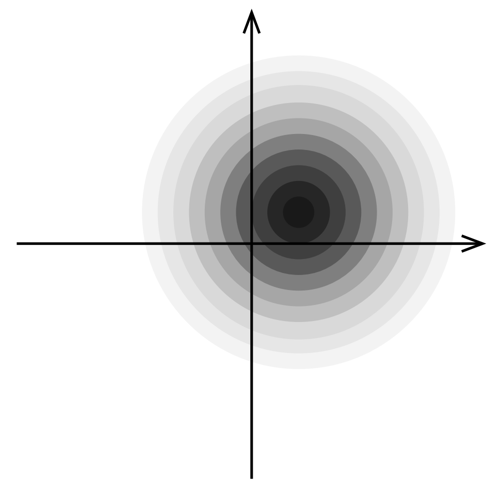
$\mathrm{C}(\mathbb{R} \to \mathbb{R})$



$\mathrm{USC}(\mathbb{R} \rightrightarrows \mathbb{R})$



$L^2(\mathbb{R})$



$\mathcal{B}(\mathbb{R}^2)$

Answer

Answer

- *They are all second-countable Kolmogorov (T_0) spaces!!!*

Answer

- *They are all second-countable Kolmogorov (T_0) spaces!!!*
- Why is this important??

Answer

- *They are all second-countable Kolmogorov (T_0) spaces!!!*
- Why is this important??
  - An elements x of a Kolmogorov space is *uniquely specified* by giving all basic open sets containing x.

Answer

- *They are all second-countable Kolmogorov (T_0) spaces!!!*
- Why is this important??
  - An elements x of a Kolmogorov space is *uniquely specified* by giving all basic open sets containing x.
  - As there are countably many basic open sets, we can *list* all basic open sets containing x.

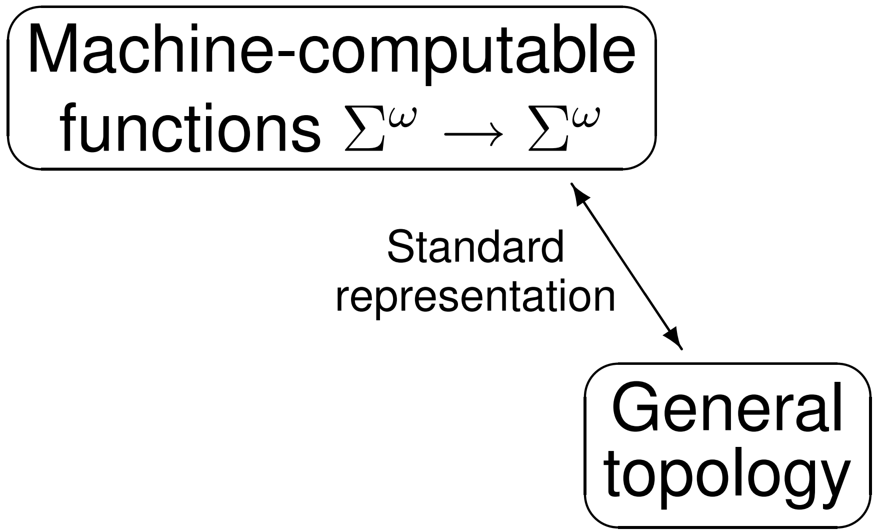
Answer

- *They are all second-countable Kolmogorov (T_0) spaces!!!*
- Why is this important??
  - An elements x of a Kolmogorov space is *uniquely specified* by giving all basic open sets containing x.
  - As there are countably many basic open sets, we can *list* all basic open sets containing x.
  - By naming the basic open sets as words in Σ^*, points of X can be named by *sequences* in Σ^ω.

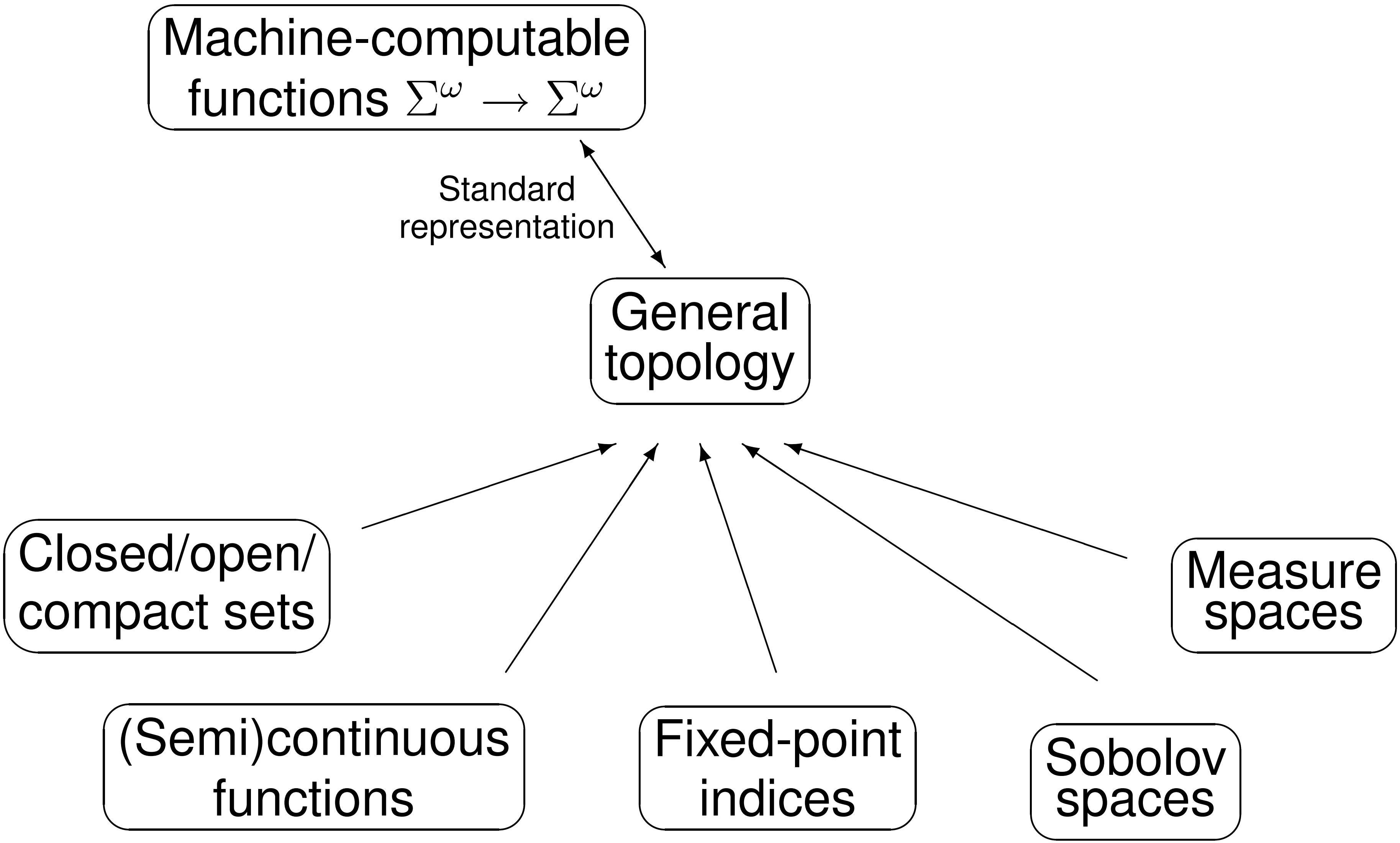
Computability theory

General
topology

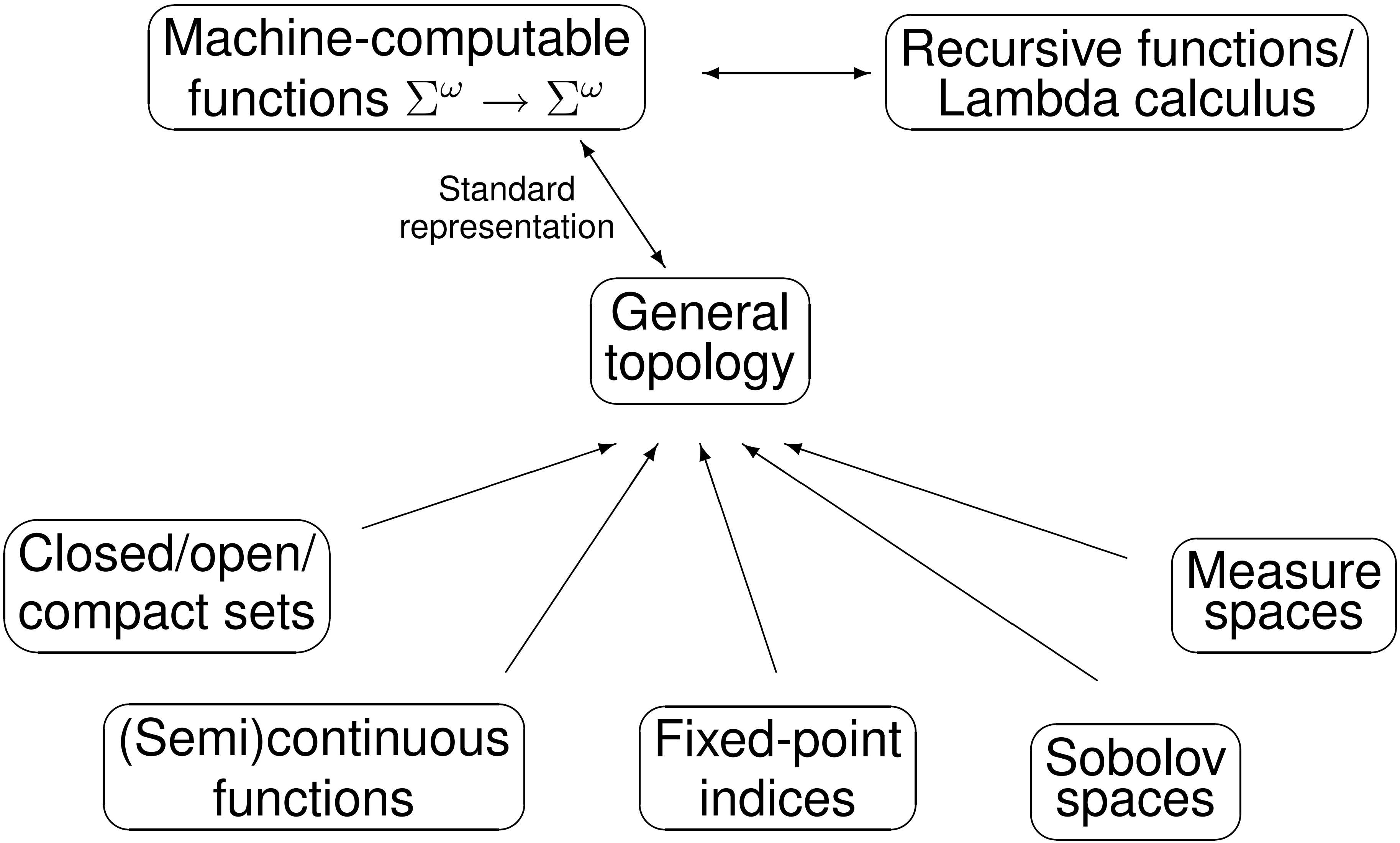
Computability theory



Computability theory



Computability theory

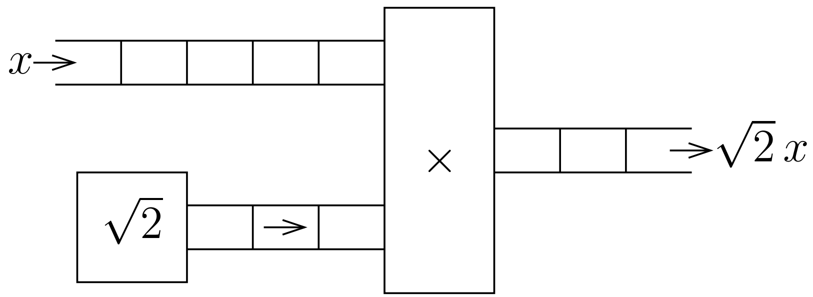


Computable analysis

[*Klaus Weihrauch,* Computable analysis - An introduction *(2000).*]

Type-two Turing machines

- Turing machine with *input*, *output* and *work* tapes; tape alphabet Σ.
- Computation is performed on *sequences* (elements of Σ^ω).



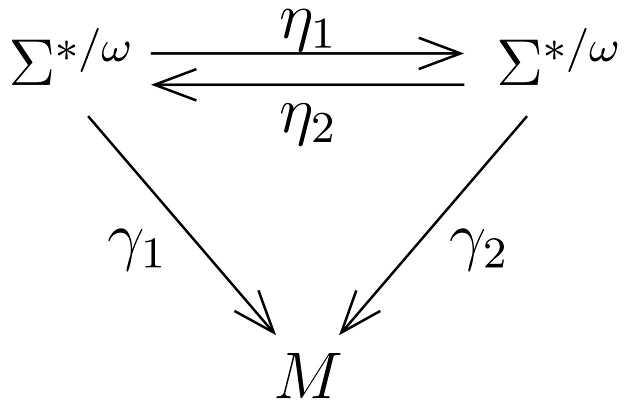
- A type-two machine computes a partial function

$$\eta :\subset \Sigma^\omega \times \cdots \times \Sigma^\omega \to \Sigma^\omega.$$

- **Theorem** If $\eta :\subset \Sigma^\omega \times \cdots \times \Sigma^\omega \to \Sigma^\omega$ is a type-two computable function, then η is continuous.

Naming systems

- A *notation* of a countable set M is a partial surjective function $\nu :\subset \Sigma^* \to M$.
- A *representation* of a set M is a partial surjective function $\delta :\subset \Sigma^\omega \to M$.
- Naming systems γ_1 and γ_2 are *equivalent* if each can be computably converted to the other.



Computability induced by naming systems

- Let $\delta_i : \Sigma^\omega \to M_i$ be representations.
- A function $f : M_1 \times \cdots \times M_k \to M_0$ is $(\delta_1, \ldots, \delta_n; \delta_0)$-*computable* if there is a Turing-computable function η with

$$\begin{array}{ccccccccc}
\Sigma^\omega & \times & \Sigma^\omega & \times \cdots \times & \Sigma^\omega & \xrightarrow{\quad\eta\quad} & \Sigma^\omega \\
\downarrow{\scriptstyle\delta_1} & & \downarrow{\scriptstyle\delta_2} & & \downarrow{\scriptstyle\delta_k} & & \downarrow{\scriptstyle\delta_0} \\
M_1 & \times & M_2 & \times \cdots \times & M_k & \xrightarrow{\quad f\quad} & M_0
\end{array}$$

Computable topological spaces

- A *computable topological space* is a tuple (X, τ, σ, ν), where
  - (X, τ) is a Kolmogorov space,
  - σ is a countable sub-base of τ, and
  - ν is a notation for σ.
- The *standard representation* of a computable topological space is the function $\delta :\subset \Sigma^\omega \to X$ defined by

$$\delta\langle w_1 w_2 \ldots\rangle = x :\iff \{\nu(w_1), \nu(w_2), \ldots\} = \{J \in \sigma \mid x \in J\}.$$

- The standard representation encodes a list of *all* elements of σ which contain x.

Fundamental theorem of computable analysis

- **Theorem** If δ_i is the standard representation of $(X_i, \tau_i, \sigma_i, \nu_i)$ for $i = 0, \ldots, k$, then any $(\delta_1, \ldots, \delta_k; \delta_0)$-computable function $f : X_1 \times \cdots \times X_k \rightarrow X_0$ is $(\tau_1, \ldots, \tau_k; \tau_0)$-continuous

Representations of real numbers

Standard representation of real numbers

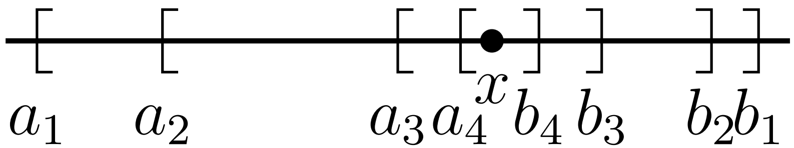
- The topology τ of $\mathbb{R}$ is generated by rational open intervals

$$\sigma = \{(a, b) \in \mathbb{Q}^2 \mid a < b\}.$$

- The standard representation ρ of $\mathbb{R}$ encodes a list of all rational open intervals (a, b) containing $x \in \mathbb{R}$.
- Under the standard representation, addition, subtraction, multiplication and division are computable.

Interval representation of real numbers

- Alternatively, represent a real number by a nested sequence of closed bounded rational intervals $\overline{I}_n = [a_n, b_n]$ with $\overline{I}_{n+1} \subset \overline{I}_n$ and $\bigcap_{n=0}^{\infty} \overline{I}_n = \{x\}$.



- The interval representation is equivalent to the standard representation.
- Note that it is not possible to decide if two numbers are equal!!

Decimal representation of real numbers

- Arithmetical operations are *not* computable for the representation by decimals!!

Decimal representation of real numbers

- Arithmetical operations are *not* computable for the representation by decimals!!
- Consider the computation of $x = 4/3 + 2/3$ in decimal representation.

Decimal representation of real numbers

- Arithmetical operations are *not* computable for the representation by decimals!!
- Consider the computation of $x = 4/3 + 2/3$ in decimal representation.
- Given finitely many digits, we have

$$x = 1 \cdot 3333... + 0 \cdot 6666... = ?{\cdot}????...$$

Decimal representation of real numbers

- Arithmetical operations are *not* computable for the representation by decimals!!
- Consider the computation of $x = 4/3 + 2/3$ in decimal representation.
- Given finitely many digits, we have

$$x = 1 \cdot 3333... + 0 \cdot 6666... = ?\cdot????...$$

- All we can deduce is that

$$1.9999 \leqslant 1 \cdot 3333 + 0 \cdot 6666 \leqslant x \leqslant 1 \cdot 3334 + 0 \cdot 6667 = 2.0001$$

Decimal representation of real numbers

- Arithmetical operations are *not* computable for the representation by decimals!!
- Consider the computation of $x = 4/3 + 2/3$ in decimal representation.
- Given finitely many digits, we have

$$x = 1 \cdot 3333... + 0 \cdot 6666... = ?\cdot????...$$

- All we can deduce is that

$$1.9999 \leqslant 1 \cdot 3333 + 0 \cdot 6666 \leqslant x \leqslant 1 \cdot 3334 + 0 \cdot 6667 = 2.0001$$

- Therefore it is *impossible* to output even the *first digit* of the answer!!

Signed-digit representation of real numbers

- The *signed-digit* representation of real numbers is given by

$$x = \sum_{i=-\infty}^{n} x_i 2^i, \text{ where } x_i \in \{-1, 0, 1\}.$$

Signed-digit representation of real numbers

- The *signed-digit* representation of real numbers is given by

$$x = \sum_{i=-\infty}^{n} x_i 2^i, \text{ where } x_i \in \{-1, 0, 1\}.$$

- The signed-digit representation is equivalent to the standard representation.

Representations of points and sets

Computable state spaces

- A *computable state space* is computable topological space (X, τ, β, ν), where
  - (X, τ) is a locally-compact second-countable Hausdorff space, and
  - β is a base for τ consisting of pre-compact open sets.
- A set $I \in \beta$ is called a *basic open set*, and $\overline{I}$ is a *basic compact set*.
- A *denotable (compact) set* is a finite union of basic compact sets, $C = \bigcup_{i=1}^{n} \overline{I}_i$.
- Assume that the basic sets are “nice” to work with.

Representation of points

- The standard representation ρ of (X, τ, β, ν) encodes a list of *all* elements of β containing x
- The standard representation is equivalant to the representation by sequences $(I_0, I_1, \ldots) \in \beta^\omega$ such that
  (i) $\overline{I}_{k+1} \subset \overline{I}_k$ for all k, and
  (ii) $\bigcap_{k=0}^{\infty} \overline{I}_k = \{x\}$.
- Equivalently, we can take $\overline{I}_{k+1} \subset I_k$ in (i).

Lower representation of open sets

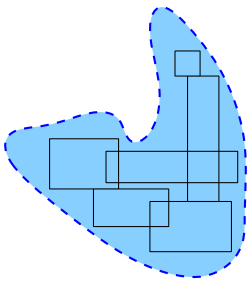
- A $\theta_<$-name of $U \in \mathcal{O}(X)$ encodes a list of all basic compact subsets of U:

$$\{I \in \beta \mid \overline{I} \subset U\}.$$

Lower representation of open sets

- A $\theta_<$-name of $U \in \mathcal{O}(X)$ encodes a list of all basic compact subsets of U:

$$\{I \in \beta \mid \overline{I} \subset U\}.$$

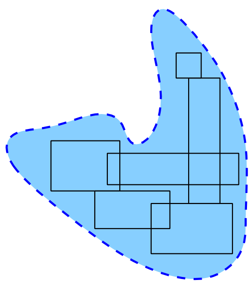


- Representation $\theta_<$ of $\mathcal{O}$.

Lower representation of open sets

- A $\theta_<$-name of $U \in \mathcal{O}(X)$ encodes a list of all basic compact subsets of U:

$$\{I \in \beta \mid \overline{I} \subset U\}.$$

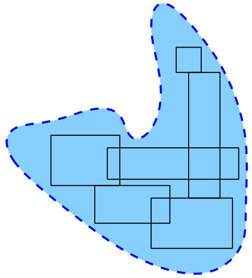


- Representation $\theta_<$ of $\mathcal{O}$.
- A $\theta_<$ name is equivalent to an increasing sequence of denotable sets (inner approximation).

Lower representation of open sets

- A $\theta_<$-name of $U \in \mathcal{O}(X)$ encodes a list of all basic compact subsets of U:

$$\{I \in \beta \mid \overline{I} \subset U\}.$$

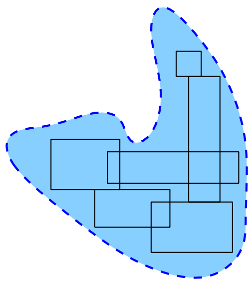


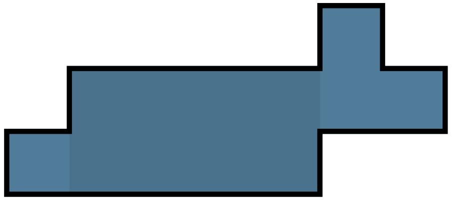
- Representation $\theta_<$ of $\mathcal{O}$.
- Inner approximation.
- A $\theta_<$ name is equivalent to an increasing sequence of denotable sets (inner approximation).

Lower representation of open sets

- A $\theta_<$-name of $U \in \mathcal{O}(X)$ encodes a list of all basic compact subsets of U:

$$\{I \in \beta \mid \overline{I} \subset U\}.$$



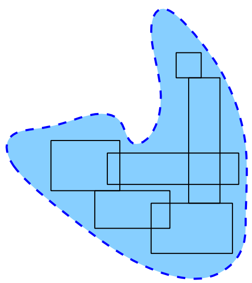


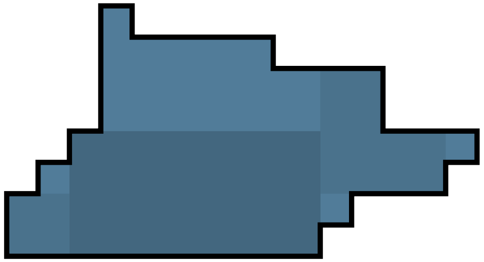
- Representation $\theta_<$ of $\mathcal{O}$.
- Inner approximation.
- A $\theta_<$ name is equivalent to an increasing sequence of denotable sets (inner approximation).

Lower representation of open sets

- A $\theta_<$-name of $U \in \mathcal{O}(X)$ encodes a list of all basic compact subsets of U:

$$\{I \in \beta \mid \overline{I} \subset U\}.$$



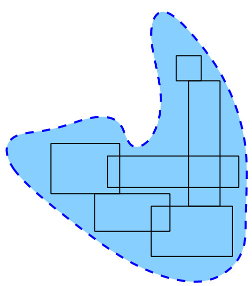


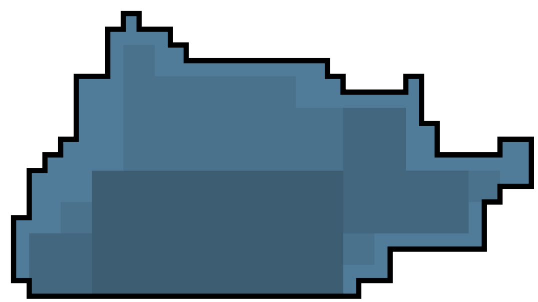
- Representation $\theta_<$ of $\mathcal{O}$.
- Inner approximation.
- A $\theta_<$ name is equivalent to an increasing sequence of denotable sets (inner approximation).

Lower representation of open sets

- A $\theta_<$-name of $U \in \mathcal{O}(X)$ encodes a list of all basic compact subsets of U:

$$\{I \in \beta \mid \overline{I} \subset U\}.$$





- Representation $\theta_<$ of $\mathcal{O}$.
- Inner approximation.
- A $\theta_<$ name is equivalent to an increasing sequence of denotable sets (inner approximation).

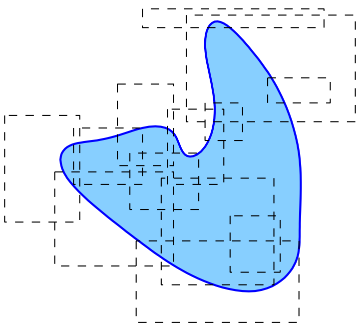
Lower representation of closed sets

- A $\psi_<$-name of $A \in \mathcal{A}(X)$ encodes a list of all basic open sets intersecting A:
$$\{J \in \beta \mid A \cap J \neq \emptyset\}.$$

Lower representation of closed sets

- A $\psi_<$-name of $A \in \mathcal{A}(X)$ encodes a list of all basic open sets intersecting A:

$$\{J \in \beta \mid A \cap J \neq \emptyset\}.$$

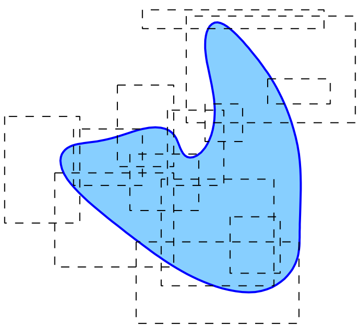


- Representation $\psi_<$ of $\mathcal{A}$.

Lower representation of closed sets

- A $\psi_<$-name of $A \in \mathcal{A}(X)$ encodes a list of all basic open sets intersecting A:

$$\{J \in \beta \mid A \cap J \neq \emptyset\}.$$

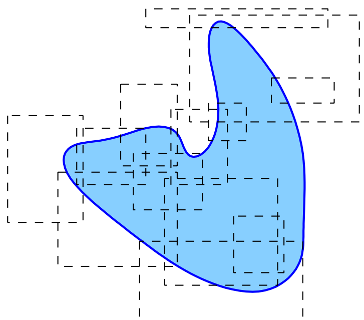


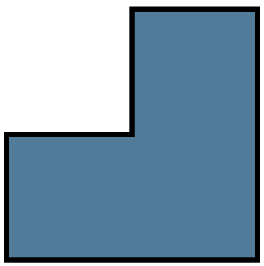
- Representation $\psi_<$ of $\mathcal{A}$.
- A $\psi_<$ name is equivalent to an "almost increasing" sequence of denotable sets (lower approximation).

Lower representation of closed sets

- A $\psi_<$-name of $A \in \mathcal{A}(X)$ encodes a list of all basic open sets intersecting A:

$$\{J \in \beta \mid A \cap J \neq \emptyset\}.$$



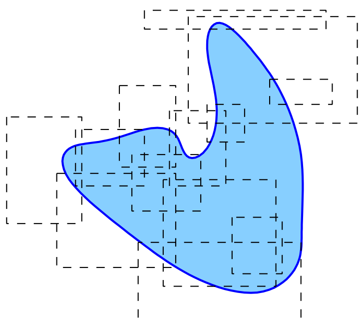


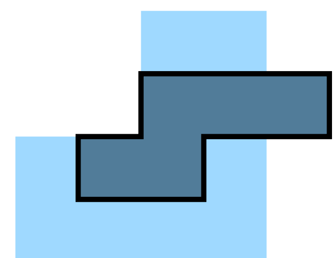
- Representation $\psi_<$ of $\mathcal{A}$.
- Lower approximation.
- A $\psi_<$ name is equivalent to an “almost increasing” sequence of denotable sets (lower approximation).

Lower representation of closed sets

- A $\psi_<$-name of $A \in \mathcal{A}(X)$ encodes a list of all basic open sets intersecting A:

$$\{J \in \beta \mid A \cap J \neq \emptyset\}.$$



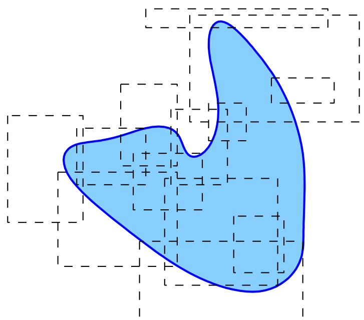


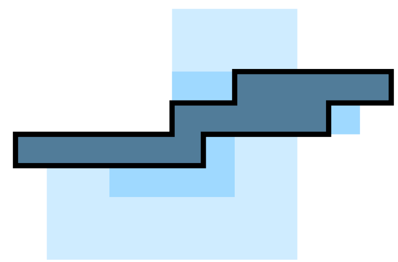
- Representation $\psi_<$ of $\mathcal{A}$.
- Lower approximation.
- A $\psi_<$ name is equivalent to an "almost increasing" sequence of denotable sets (lower approximation).

Lower representation of closed sets

- A $\psi_<$-name of $A \in \mathcal{A}(X)$ encodes a list of all basic open sets intersecting A:

$$\{J \in \beta \mid A \cap J \neq \emptyset\}.$$



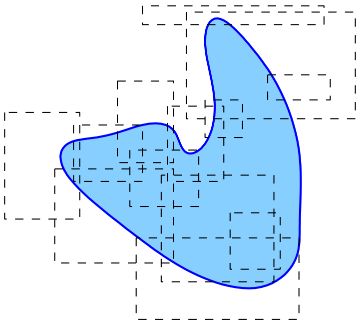


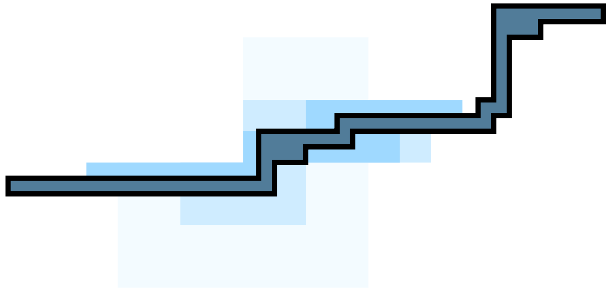
- Representation $\psi_<$ of $\mathcal{A}$.
- Lower approximation.
- A $\psi_<$ name is equivalent to an "almost increasing" sequence of denotable sets (lower approximation).

Lower representation of closed sets

- A $\psi_<$-name of $A \in \mathcal{A}(X)$ encodes a list of all basic open sets intersecting A:

$$\{J \in \beta \mid A \cap J \neq \emptyset\}.$$





- Representation $\psi_<$ of $\mathcal{A}$.
- Lower approximation.
- A $\psi_<$ name is equivalent to an “almost increasing” sequence of denotable sets (lower approximation).

Upper representation of closed sets

- A $\psi_>$-name of A encodes a list of all basic compact sets disjoint from A:

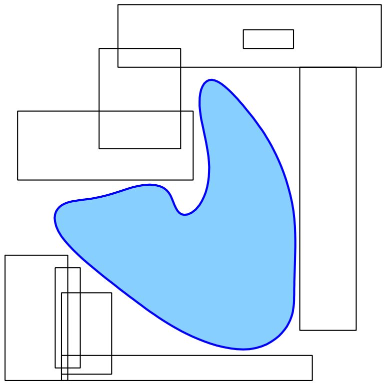
$$\{I \in \beta \mid A \cap \overline{I} = \emptyset\}.$$

Upper representation of closed sets

- A $\psi_>$-name of A encodes a list of all basic compact sets disjoint from A:

$$\{I \in \beta \mid A \cap \overline{I} = \emptyset\}.$$

- Representation $\psi_>$ of $\mathcal{A}$.



Upper representation of compact sets

- A $\kappa_>$-name of $C \in \mathcal{K}(X)$ is encodes a list of all basic open covers of C:

$$\{(J_1, \ldots, J_k) \in \beta^* \mid C \subset J_1 \cup \cdots \cup J_k\}.$$

Upper representation of compact sets

- A $\kappa_>$-name of $C \in \mathcal{K}(X)$ is encodes a list of all basic open covers of C:

$$\{(J_1, \ldots, J_k) \in \beta^* \mid C \subset J_1 \cup \cdots \cup J_k\}.$$

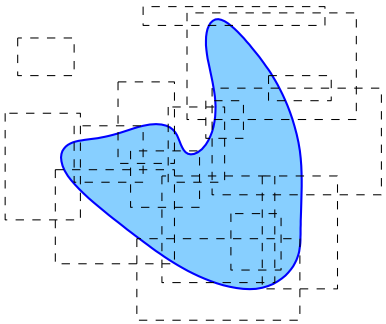
- A $\kappa_>$-name is equivalent to a $\psi_>$-name and a bounding box.

Upper representation of compact sets

- A $\kappa_>$-name of $C \in \mathcal{K}(X)$ is encodes a list of all basic open covers of C:

$$\{(J_1, \ldots, J_k) \in \beta^* \mid C \subset J_1 \cup \cdots \cup J_k\}.$$

- A $\kappa_>$-name is equivalent to a $\psi_>$-name and a bounding box.



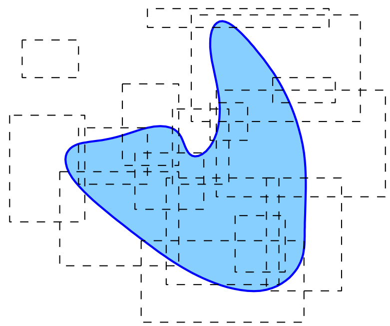
- Representation $\kappa_>$ of $\mathcal{K}$.

Upper representation of compact sets

- A $\kappa_>$-name of $C \in \mathcal{K}(X)$ is encodes a list of all basic open covers of C:

$$\{(J_1, \ldots, J_k) \in \beta^* \mid C \subset J_1 \cup \cdots \cup J_k\}.$$

- A $\kappa_>$-name is equivalent to a $\psi_>$-name and a bounding box.



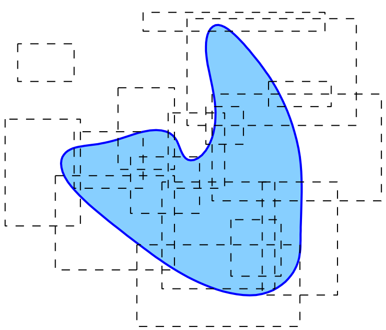
- Representation $\kappa_>$ of $\mathcal{K}$.
- A $\kappa_>$ name is equivalent to a decreasing sequence of denotable sets (outer approximation).

Upper representation of compact sets

- A $\kappa_>$-name of $C \in \mathcal{K}(X)$ is encodes a list of all basic open covers of C:

$$\{(J_1, \ldots, J_k) \in \beta^* \mid C \subset J_1 \cup \cdots \cup J_k\}.$$

- A $\kappa_>$-name is equivalent to a $\psi_>$-name and a bounding box.



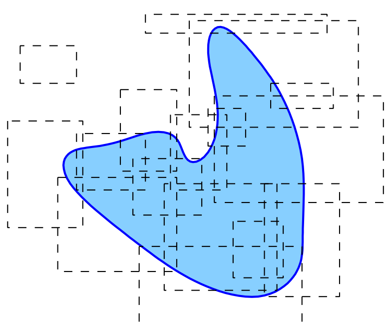
- Representation $\kappa_>$ of $\mathcal{K}$.
- Outer approximation.
- A $\kappa_>$ name is equivalent to a decreasing sequence of denotable sets (outer approximation).

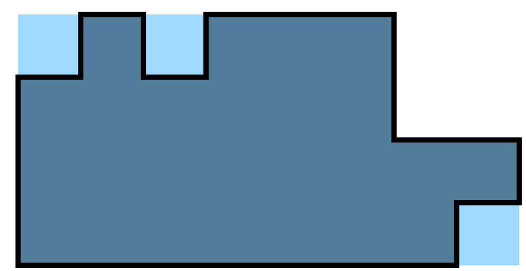
Upper representation of compact sets

- A $\kappa_>$-name of $C \in \mathcal{K}(X)$ is encodes a list of all basic open covers of C:

$$\{(J_1, \ldots, J_k) \in \beta^* \mid C \subset J_1 \cup \cdots \cup J_k\}.$$

- A $\kappa_>$-name is equivalent to a $\psi_>$-name and a bounding box.





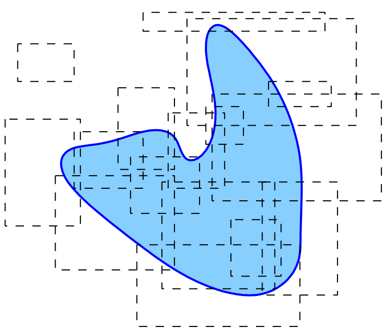
- Representation $\kappa_>$ of $\mathcal{K}$.
- Outer approximation.
- A $\kappa_>$ name is equivalent to a decreasing sequence of denotable sets (outer approximation).

Upper representation of compact sets

- A $\kappa_>$-name of $C \in \mathcal{K}(X)$ is encodes a list of all basic open covers of C:

$$\{(J_1, \dots, J_k) \in \beta^* \mid C \subset J_1 \cup \cdots \cup J_k\}.$$

- A $\kappa_>$-name is equivalent to a $\psi_>$-name and a bounding box.



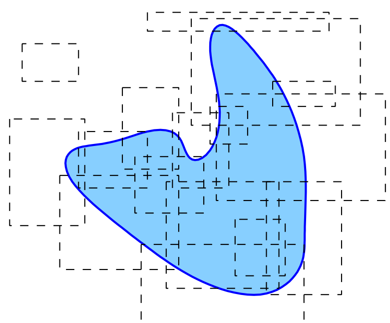
- Representation $\kappa_>$ of $\mathcal{K}$.
- Outer approximation.
- A $\kappa_>$ name is equivalent to a decreasing sequence of denotable sets (outer approximation).

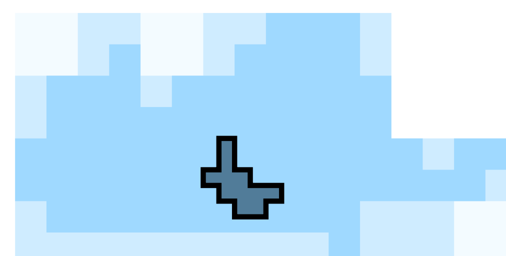
Upper representation of compact sets

- A $\kappa_>$-name of $C \in \mathcal{K}(X)$ is encodes a list of all basic open covers of C:

$$\{(J_1, \ldots, J_k) \in \beta^* \mid C \subset J_1 \cup \cdots \cup J_k\}.$$

- A $\kappa_>$-name is equivalent to a $\psi_>$-name and a bounding box.





- Representation $\kappa_>$ of $\mathcal{K}$.
- Outer approximation.
- A $\kappa_>$ name is equivalent to a decreasing sequence of denotable sets (outer approximation).

Representation of compact sets

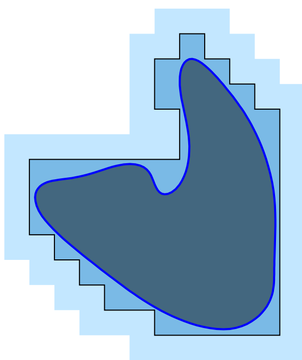
- A κ-name of C is a combination of a lower $\psi_<$-name and an upper $\kappa_>$-name.

Representation of compact sets

- A κ-name of C is a combination of a lower $\psi_<$-name and an upper $\kappa_>$-name.
- κ is the standard representation of the *Vietoris topology* on $\mathcal{K}(X)$.

Representation of compact sets

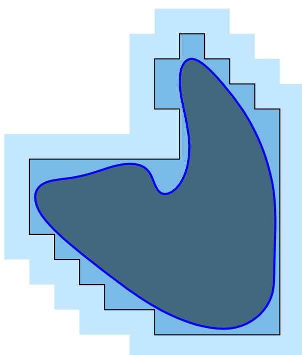
- A κ-name of C is a combination of a lower $\psi_<$-name and an upper $\kappa_>$-name.
- κ is the standard representation of the *Vietoris topology* on $\mathcal{K}(X)$.



- Representation κ of $\mathcal{K}$.

Representation of compact sets

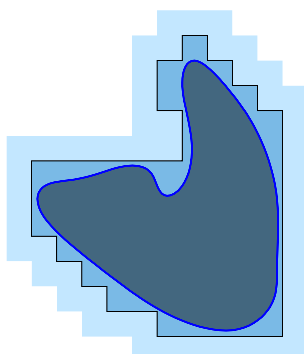
- A κ-name of C is a combination of a lower $\psi_<$-name and an upper $\kappa_>$-name.
- κ is the standard representation of the *Vietoris topology* on $\mathcal{K}(X)$.

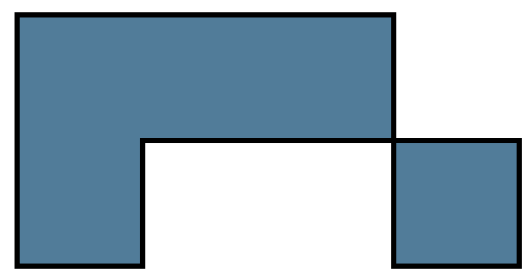


- Representation κ of $\mathcal{K}$.
- κ is equivalent to arbitrarily accurate approximation in the Hausdorff metric d_H.

Representation of compact sets

- A κ-name of C is a combination of a lower $\psi_<$-name and an upper $\kappa_>$-name.
- κ is the standard representation of the *Vietoris topology* on $\mathcal{K}(X)$.

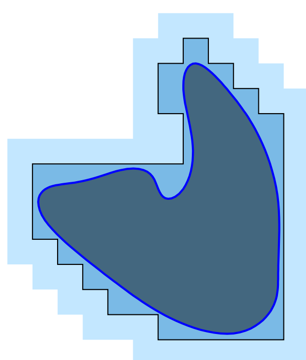


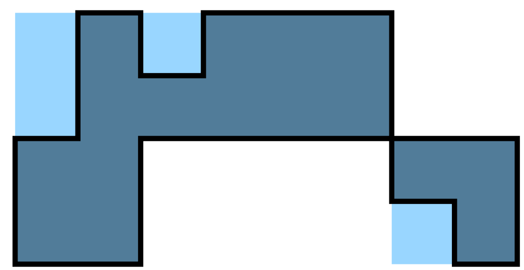


- Representation κ of $\mathcal{K}$.
- Approximation in d_H.
- κ is equivalent to arbitrarily accurate approximation in the Hausdorff metric d_H.

Representation of compact sets

- A κ-name of C is a combination of a lower $\psi_<$-name and an upper $\kappa_>$-name.
- κ is the standard representation of the *Vietoris topology* on $\mathcal{K}(X)$.

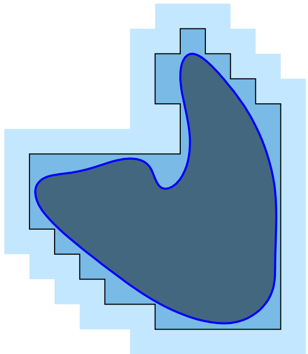


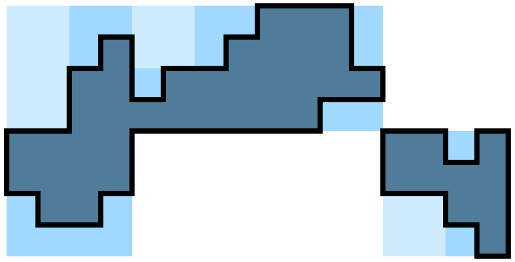


- Representation κ of $\mathcal{K}$.
- Approximation in d_H.
- κ is equivalent to arbitrarily accurate approximation in the Hausdorff metric d_H.

Representation of compact sets

- A κ-name of C is a combination of a lower $\psi_<$-name and an upper $\kappa_>$-name.
- κ is the standard representation of the *Vietoris topology* on $\mathcal{K}(X)$.

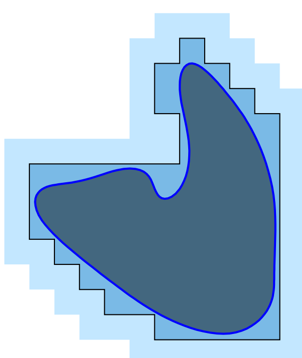


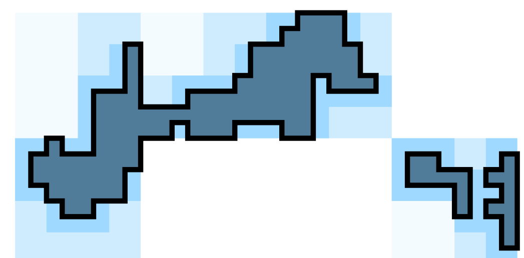


- Representation κ of $\mathcal{K}$.
- Approximation in d_H.
- κ is equivalent to arbitrarily accurate approximation in the Hausdorff metric d_H.

Representation of compact sets

- A κ-name of C is a combination of a lower $\psi_<$-name and an upper $\kappa_>$-name.
- κ is the standard representation of the *Vietoris topology* on $\mathcal{K}(X)$.





- Representation κ of $\mathcal{K}$.
- Approximation in d_H.
- κ is equivalent to arbitrarily accurate approximation in the Hausdorff metric d_H.

Computable operators

- Union $A, B \mapsto A \cup B$ is $(\psi_<, \psi_<; \psi_<)$-computable and $(\psi_>, \psi_>; \psi_>)$-computable.
- Intersection $A, B \mapsto A \cap B$ is $(\psi_>, \psi_>; \psi_>)$-computable; $(A, C) \mapsto A \cap C$ is $(\psi_>, \kappa_>; \kappa_>)$-computable.
- Intersection $A, B \mapsto A \cap B$ is *not* $(\psi_<, \psi_<; \psi_<)$-computable.
- Closed intersection $(A, U) \mapsto \mathrm{cl}(A \cap U)$ is $(\psi_<, \theta_<; \psi_<)$-computable.

Representations of maps

Representation of continuous functions

- The *compact-open topology* τ^{co} on $C(X \to Y)$ is the topology of uniform convergence on compact sets.
- A δ^{co}-name of f encodes a list of pairs $(\overline{I}, J)$ such that the image of $\overline{I}$ is a subset of J:

$$\{(I, J) \in \beta_X \times \beta_Y \mid f(\overline{I}) \subset J\}.$$

Evaluation representation

- The *evaluation representation* $\delta^{\to}$ of $C(X \to Y)$ encodes a function f by a function $\mathcal{F} : \overline{\beta}_X \to \overline{\beta}_Y$ such that
  (i) $f(\overline{I}) \subset \mathcal{F}(\overline{I})$
  (ii) $\overline{I}_1 \subset \overline{I}_2 \implies \mathcal{F}(\overline{I}_1) \subset \mathcal{F}(\overline{I}_2)$, and
  (iii) If $\overline{I}_{i+1} \subset \overline{I}_i$ and $\bigcap_{i=0}^{\infty} \overline{I}_i = \{x\}$, then $\bigcap_{i=0}^{\infty} \mathcal{F}(\overline{I}_i) = \{f(x)\}$
- May alternatively require $f(\overline{I}) \subset \mathrm{int}(\mathcal{F}(\overline{I}))$ in (i).
- The evaluation representation is equivalent to the compact-open representation.

Computable operators on continuous functions

- Evaluation $(f, x) \mapsto f(x)$ is $(\delta^{\mathrm{co}}, \rho; \rho)$-computable.
- Image $(f, A) \mapsto \mathrm{cl}(f(A))$ is $(\delta^{\mathrm{co}}, \psi_<; \psi_<)$-computable;
  $(f, C) \mapsto f(C)$ is $(\delta^{\mathrm{co}}, \kappa_>; \kappa_>)$-computable.
- Preimage $(f, U) \mapsto f^{-1}(U)$ is $(\delta^{\mathrm{co}}, \theta_<; \theta_<)$-computable;
  $(f, A) \mapsto f^{-1}(A)$ is $(\delta^{\mathrm{co}}, \psi_>; \psi_>)$-computable.

- $(f, A) \mapsto \mathrm{cl}(f(A))$ is *not* $(\delta^{\mathrm{co}}, \psi_>, \psi_>)$-computable.
- $(f, A) \mapsto f^{-1}(A)$ is *not* $(\delta^{\mathrm{co}}, \psi_<, \psi_<)$-computable.

Representation of differentiable functions

- The $\gamma^{(r)}$-representation of $C^{(r)}(X \to Y)$ consists of a δ^{co}-representation of f and each of its first r derivative.
- Operations such as set-image are more efficient using the $\gamma^{(1)}$-representation than the $\gamma^{(0)} = \delta^{\mathrm{co}}$ representation.

Representations of multivalued functions

- The lower representation $\mu_<^{\mathcal{A}}$ of closed-valued lower-semicontinuous functions $\mathrm{LSC}^{\mathcal{A}}(X \rightrightarrows Y)$ encodes all pairs $(\overline{I}, J)$ such that $\overline{I} \subset F^{-1}(J)$.
- The upper representation $\mu_>^{\mathcal{K}}$ of compact-valued upper-semicontinuous functions $\mathrm{USC}^{\mathcal{K}}(X \rightrightarrows Y)$ encodes all tuples $(\overline{I}, J_1, \ldots, J_k)$ such that $F(\overline{I}) \subset \bigcup_{i=1}^{k} J_i$.
- $\mu_<^{\mathcal{A}}$ and $\mu_>^{\mathcal{K}}$ are each equivalent to δ^{co} if F is a single-valued continuous function.

Implementation of computable analysis

Tools for computable analysis

- Set-based analysis
  — GAIO [Dellnitz, Junge]
- Taylor integration schemes
  — AWA[Lohner]
  — CAPD [Mrozek, Wilczak, Żelawski, Zgliczyński]
  — Cosy [Berz, Makino]
- Reachability analysis
  — d/dt [Asarin, Dang, Maler]
  — Hy(per)Tech [Henzinger et al.]
  — Ellipsoidal calculus [Kurzhanski]
- — Ariadne [Balluchi, Casagrande, C., Murrieri, Villa]

Ariadne–Numerical types

- `double` Machine double-precision floating-point.
  Fixed precision, inexact arithmetic, FAST!!
- `MPFloat` Multiple-precision floating-point.
  Arbitrary precision, inexact arithmetic.
- `Dyadic` Arbitrary precision, exact addition, subtraction, multiplication and division by 2.
- `Rational` Arbitrary precision, exact arithmetic.

  [Gnu Multiple-Precision Library (GMP), MPFR]

- `Interval<real_type>` Interval arithmetic.

Ariadne–Basic set types

- Basic set types include
  - `Simplex`
  - `Cuboid`
  - `Parallelotope`
  - `Zonotope`
  - `Polytope`
  - `Ellipsoid`

Ariadne–Operations on basic sets

- Operations on basic sets
  - `contains`
  - `disjoint`
  - `interiors_intersect`
  - `subset`
  - `bounding_box`
  - `regular_intersection`
  - `minkowski_sum`

Ariadne–Denotable set types

- *Denotable sets* are finite unions of basic sets.
  - `ListSet<basic_set_type>`
  - `GridMaskSet`, `GridCellListSet`
  - `PartitionTreeSet`
  - `SimplicialComplex`

Ariadne–Maps

- Maps are specified using the evaluation representation on cuboids.
  - `map.apply(cuboid)`
- Evaluation on other sets for performed using C^1 methods.
  - `c1map.derivative(cuboid)`
  - `c1map.apply(parallelotope)`
- Also implement vector fields and control systems.

Computability in dynamical systems

Computability of reachable sets

- The *reachable set* from X_0 under f is

$$\mathrm{Reach}(f, X_0) := \{x \mid \exists\, x_0, x_1, \ldots, x_n \text{ s.t. } x_0 \in X_0,\ f(x_i) = x_{i+1} \text{ and } x_n = x\}.$$

$$= \bigcup_{n=0}^{\infty} f^n(X_0)$$

Computability of reachable sets

- The *reachable set* from X_0 under f is

  $$\mathrm{Reach}(f, X_0) := \{x \mid \exists\, x_0, x_1, \ldots, x_n \text{ s.t. } x_0 \in X_0,\ f(x_i) = x_{i+1} \text{ and } x_n = x\}.$$
  $$= \bigcup_{n=0}^{\infty} f^n(X_0)$$

- **Theorem**

  (i) The operator $(f, A) \mapsto \mathrm{clReach}(f, A)$ is $(\delta^{\mathrm{co}}, \psi_<; \psi_<)$-computable.

Computability of reachable sets

- The *reachable set* from X_0 under f is

$$\mathrm{Reach}(f, X_0) := \{x \mid \exists\, x_0, x_1, \ldots, x_n \text{ s.t. } x_0 \in X_0,\ f(x_i) = x_{i+1} \text{ and } x_n = x\}.$$

$$= \bigcup_{n=0}^{\infty} f^n(X_0)$$

- **Theorem**
  (i) The operator $(f, A) \mapsto \mathrm{clReach}(f, A)$ is $(\delta^{\mathrm{co}}, \psi_<; \psi_<)$-computable.
  (ii) $\mathrm{clReach}$ is not upper-semicontinuous, so is not $(\delta^{\mathrm{co}}, \kappa_>; \psi_>)$-computable.

Computability of reachable sets

- The *reachable set* from X_0 under f is

$$\mathrm{Reach}(f, X_0) := \{x \mid \exists\, x_0, x_1, \ldots, x_n \text{ s.t. } x_0 \in X_0,\ f(x_i) = x_{i+1} \text{ and } x_n = x\}.$$
$$= \bigcup_{n=0}^{\infty} f^n(X_0)$$

- **Theorem**

  (i) The operator $(f, A) \mapsto \mathrm{clReach}(f, A)$ is $(\delta^{\mathrm{co}}, \psi_<; \psi_<)$-computable.

  (ii) $\mathrm{clReach}$ is not upper-semicontinuous, so is not $(\delta^{\mathrm{co}}, \kappa_>; \psi_>)$-computable.

- Look for an upper-semicontinuous over-approximation to Reach.

Computability of chain reachability

- An ϵ-*chain* is a sequence $(x_0, x_1, \ldots, x_n)$ such that there exist y_i with $y_{i+1} = f(x_i)$ and $d(x_i, y_i) < \epsilon$ for all i.

Computability of chain reachability

- An ϵ-*chain* is a sequence $(x_0, x_1, \ldots, x_n)$ such that there exist y_i with $y_{i+1} = f(x_i)$ and $d(x_i, y_i) < \epsilon$ for all i.
- The *chain-reachable set* is

$$\mathrm{ChainReach}(f, X_0) := \{x \mid \forall \epsilon > 0,\ \exists\ \epsilon\text{-chain } x_0, \ldots, x_n \text{ with } x_0 \in X_0 \text{ and } x_n = x\}$$

Computability of chain reachability

- An *ϵ-chain* is a sequence $(x_0, x_1, \dots, x_n)$ such that there exist y_i with $y_{i+1} = f(x_i)$ and $d(x_i, y_i) < \epsilon$ for all i.
- The *chain-reachable set* is

$$\mathrm{ChainReach}(f, X_0) := \{x \mid \forall\epsilon > 0,\ \exists\ \epsilon\text{-chain } x_0, \dots, x_n \text{ with } x_0 \in X_0 \text{ and } x_n = x\}$$

- **Theorem** If $\mathrm{ChainReach}(f, C)$ is compact, then
  (i) $\mathrm{ChainReach}(f, C)$ is $(\delta^{\mathrm{co}}, \kappa_>; \kappa_>)$-computable,

Computability of chain reachability

- An *ϵ-chain* is a sequence $(x_0, x_1, \ldots, x_n)$ such that there exist y_i with $y_{i+1} = f(x_i)$ and $d(x_i, y_i) < \epsilon$ for all i.
- The *chain-reachable set* is

$$\mathrm{ChainReach}(f, X_0) := \{x \mid \forall \epsilon > 0,\ \exists\ \epsilon\text{-chain } x_0, \ldots, x_n \text{ with } x_0 \in X_0 \text{ and } x_n = x\}$$

- **Theorem** If $\mathrm{ChainReach}(f, C)$ is compact, then
  (i) $\mathrm{ChainReach}(f, C)$ is $(\delta^{\mathrm{co}}, \kappa_>; \kappa_>)$-computable, and
  (ii) $\mathrm{ChainReach}(f, C) = \limsup_{f' \to f,\ C' \to C} \mathrm{Reach}(f', C')$.

Solutions of differential equations

- Denote by $\Phi_t^f(x)$ the solution of the differential equation $\dot{x} = f(x)$ at time t.
- **Theorem** If f is Lipschitz, then the map $(f, t) \mapsto \Phi_t^f$ is $(\delta^{\mathrm{co}}, \rho; \delta^{\mathrm{co}})$-computable.
- Use Taylor methods to perform the compuation.

Solutions of differential inclusions

- If F is Lipschitz and lower-semicontinuous with closed valued, then $(F, A, t) \mapsto \Phi_t^F(A)$ is $(\mu_<^{\mathcal{A}}, \psi_<, \rho; \psi_<)$-computable.
- If F is upper-semicontinuous with compact values, and solutions of $\dot{x} \in F(x)$ remain bounded, then $(F, C, t) \mapsto \Phi_t^F(C)$ is $(\mu_>^{\mathcal{K}}, \kappa_>, \rho; \kappa_>)$-computable.

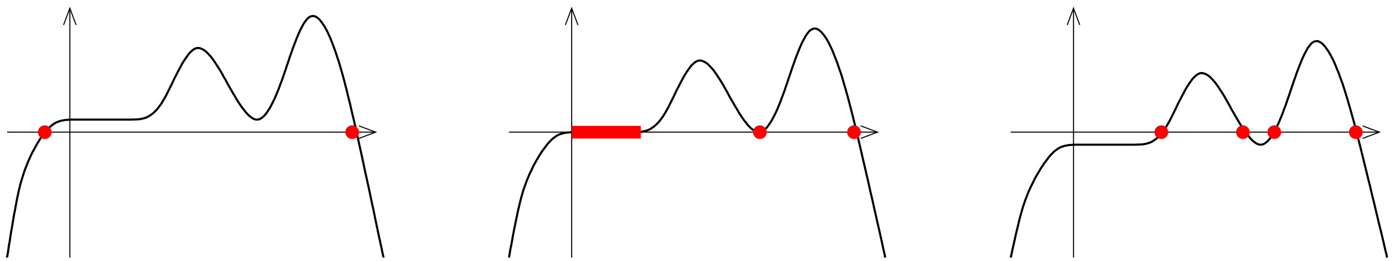
Computation of fixed point sets

- **Theorem**

  (i) $f \mapsto \mathrm{Fix}(f)$ is $(\delta^{\mathrm{co}}, \psi_{>})$-computable.

  (ii) $f \mapsto \mathrm{Fix}(f)$ is *not* $(\delta^{\mathrm{co}}, \psi_{<})$-computable.

- Arbitrarily perturbations can destroy fixed-points.



- Intervals of fixed-points are especially problematic...

The fixed-point index

- The Lefschetz fixed-point index is a computable partial function
$$\mathrm{Ind} :\subset \mathcal{O}(X) \times C(X \to X) \to \mathbb{Z}$$
defined on pairs (U, f) such that $\mathrm{Fix}(f) \cap \partial U = \emptyset$.
- **Theorem** If $\mathrm{Ind}(U, f) \neq 0$, then $\mathrm{Fix}(f) \cap U \neq \emptyset$.
- **Theorem (Jiang)** If X is a manifold, U is connected, and $\mathrm{Ind}(f, U) = 0$, there then all fixed points of f in U can be removed by a homotopy supported in U, unless U is a surface of negative Euler characteristic.
- The fixed-point index can be used to compute a ψ_{fx}*-name* of *robust* fixed-points of f.
- The representation ψ_{fx} is *incompatible* with both $\psi_<$ and $\psi_>$.

Fixed points of multivalued maps

- The fixed-point index is also defined for upper-semicontinuous *multivalued* maps with *acyclic* values.
- Index theory is a way of getting *lower* bounds on sets from *outer* approximations of *upper*-semicontinuous acyclic-valued maps.

Computability of forward invariant sets

- The *forward invariance kernel* of f is defined
$$\mathrm{Inv}(f, B) = \{x \mid \exists \text{ forward orbit of } f \text{ in } B \text{ through } x\}.$$
- **Theorem** $(f, C) \mapsto \mathrm{Inv}(f, C)$ is $(\delta^{\mathrm{co}}, \kappa_>; \kappa_>)$-computable, but not $(\delta^{\mathrm{co}}, \kappa; \psi_<)$-computable.
- The *robust forward invariance kernel* of U is
$$\mathrm{RobustInv}(F, U) := \bigcup\{C \in \mathcal{K} \mid C \subset U \text{ and } F(C) \subset \mathrm{int}(C)\}.$$
- **Theorem** $\mathrm{RobustInv}(F, U)$ is $(\delta^{\mathrm{co}}, \theta_<; \theta_<)$-computable, and $\mathrm{RobustInv}(f, U) = \liminf \mathrm{Inv}(f, U)$.

Open problems

Computability of invariant sets

- The Conley index can compute invariant sets of upper-semicontinuous multivalued maps with acyclic values.
- **Question** How does the Conley index compare with the direct computation of robust forward invariant sets?

Computability of topological entropy

- The topological entropy of cellular automata is uncomputable. [Hurd, Kari & Culik, 1992]

- h_{top} is lower-semicontinuous for $C(M^n)$ for $n = 1$, and for $\text{Diff}^{1+\epsilon}(M^n)$ for $n = 2$. [Katok, 1980]
- h_{top} is not continuous on $C^{1+\epsilon}(M^n)$ for $n \geqslant 2$. [Misiurewicz, 1971]
- h_{top} is upper-semicontinuous on $C^{\infty}(M)$. [Yomdin, 1987]

- Is h_{top} semicomputable when it is semicontinuous?
- What are the best semicomputable approximations to h_{top}?

Computability of measures

- What is an appropriate representation of a probability measure?
  — theory of valuations [Edelat]
- Is it (im)possible to compute invariant measures for a dynamical system on a compact space?
- Is it possible to compute the measure of the set of stochastic parameters of an interval map?

Conclusions

Conclusions

- Computable analysis provides a framework for formal study of computability in an approximative setting.
- Defines semantics for fundamental operations to be implemented by tool developers.
- Computability of many important dynamical systems concepts still unknown.

That's all, folks!