Minicourse: Lecture 2
Applying Topology to Spaces of Countable Structures

Russell Miller

Queens College & CUNY Graduate Center

DDC Program, Part I: Virtual Semester

Mathematical Sciences Research Institute
Berkeley, CA (remotely)
Autumn 2020

Plan of the Minicourse

Week 1: Specific example: subrings of $\mathbb{Q}$.
Online discussion: Thursday, Sept. 24, 11:00 PDT.

Week 2: Computability and continuity.
Online discussion: Thursday, Oct. 1, 11:00 PDT.

Week 3: Classifications of spaces of structures.
Online discussion: Thursday, Oct. 8, 11:00 PDT.

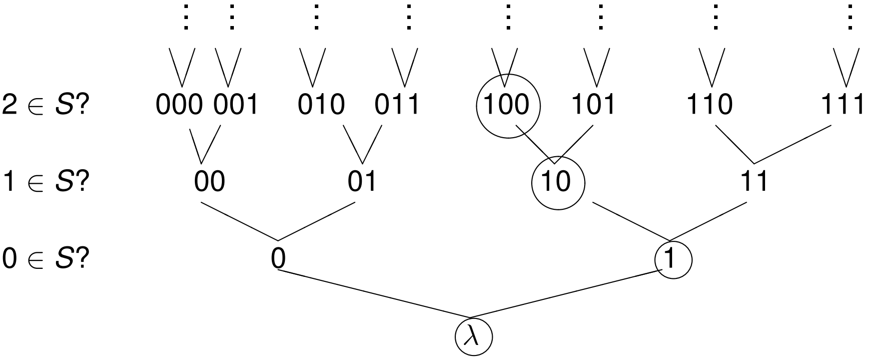
Week 4: The space of algebraic fields.
Online discussion: Thursday, Oct. 15, 11:00 PDT.

Week 5: Other related questions.
Online discussion: Thursday, Oct. 22, 11:00 PDT.

(Also watch Caleb Springer's MSRI Junior Seminar: Oct. 20, 09:00.)

Cantor space $2^{\mathbb{N}}$

$2^{\mathbb{N}}$ means $\{f : \mathbb{N} \to \{0, 1\}\}$, the set of all binary-valued functions on $\mathbb{N}$. We think of a point in $2^{\mathbb{N}}$ as a (countable) infinite binary sequence 10001111010010 . . .. This point also names the subset $S = \{0, 4, 5, 6, 7, 9, 12, \ldots\} \subseteq \mathbb{N}$, by giving its characteristic function. Finally, each point is a path through the complete binary tree $2^{<\mathbb{N}}$:



Topology of Cantor space

Recall: basic open sets are of the form $\mathcal{U}_\sigma = \{f : \mathbb{N} \to \{0, 1\} : \sigma \subset f\}$, meaning that σ is an initial segment of f. The sets $\mathcal{U}_{Y,N}$ defined last week form a slightly different basis for the same topology. The intuition is that membership of f in an open set is always confirmed by a finite amount of information about f.

For f to belong to a closed set may require infinitely much information. For example, the set

$$\mathcal{V} = \{f \in 2^{\mathbb{N}} : f \text{ contains six consecutive zeros somewhere}\}$$

is an open set. If $f \in \mathcal{V}$, some finite $\sigma \subset f$ gives a reason why. To see that $f \notin \mathcal{V}$ would require looking at the entire infinite sequence f.

Doesn’t “Cantor space” mean something else?

In real analysis, one meets the “middle thirds set” in the unit interval. This is also often called the *Cantor set*. It is constructed by starting with the unit interval and repeatedly removing the open middle third of each remaining interval. Shouldn’t we have chosen a different name?

Doesn't "Cantor space" mean something else?

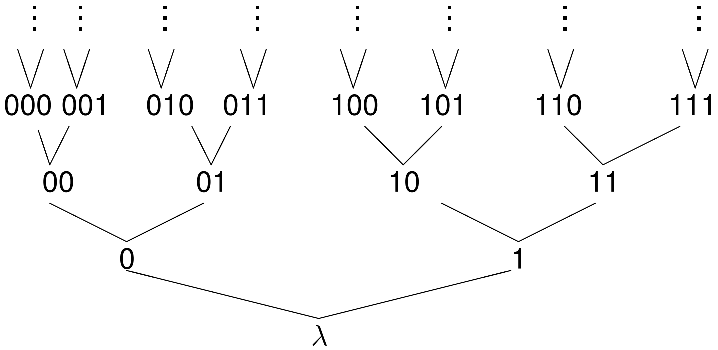
In real analysis, one meets the "middle thirds set" in the unit interval. This is also often called the *Cantor set*. It is constructed by starting with the unit interval and repeatedly removing the open middle third of each remaining interval. Shouldn't we have chosen a different name?

No: these two spaces are homeomorphic!

Doesn't "Cantor space" mean something else?

In real analysis, one meets the "middle thirds set" in the unit interval. This is also often called the *Cantor set*. It is constructed by starting with the unit interval and repeatedly removing the open middle third of each remaining interval. Shouldn't we have chosen a different name?

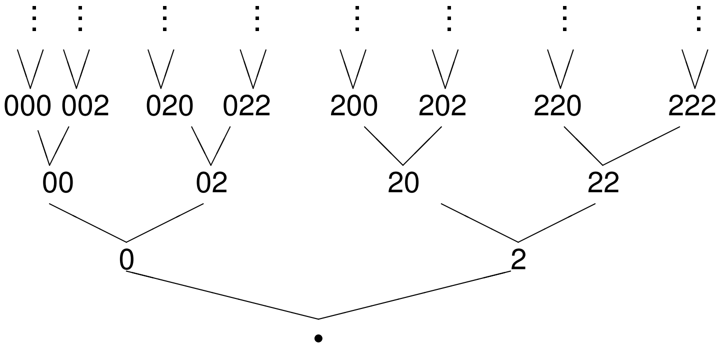
No: these two spaces are homeomorphic!



Doesn't "Cantor space" mean something else?

In real analysis, one meets the "middle thirds set" in the unit interval. This is also often called the *Cantor set*. It is constructed by starting with the unit interval and repeatedly removing the open middle third of each remaining interval. Shouldn't we have chosen a different name?

No: these two spaces are homeomorphic!



Oracle Turing machines

Since many of our spaces of structures will be homeomorphic to Cantor space, or to a quotient of it, we want a notion of computability for functions $F : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$.

Ordinary Turing machines only compute (partial) functions $\varphi : \mathbb{N} \to \mathbb{N}$.

Oracle Turing machines

Since many of our spaces of structures will be homeomorphic to Cantor space, or to a quotient of it, we want a notion of computability for functions $F : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$.

Ordinary Turing machines only compute (partial) functions $\varphi : \mathbb{N} \to \mathbb{N}$.

An *oracle Turing machine* still computes partial functions $: \mathbb{N} \to \mathbb{N}$, but can be endowed with one (or several) *oracles*. An oracle is a countable infinite read-only tape on which is written the characteristic function of a set $A \subseteq \mathbb{N}$. This set A is the *oracle*.

The program for an oracle Turing machine is still a finite set of instructions. The oracle is not part of the program. The program uses the usual Turing-machine instructions, plus new instructions:

- Move one cell left or right on the oracle tape.
- Read the current oracle-tape cell. If it's 0, do this. If it's 1, do that.

Oracle programs and Turing reducibility

Earlier, we considered membership in $HTP(R)$. For certain subrings $R \subseteq \mathbb{Q}$, such as semilocal subrings, we could compute, for each $f \in \mathbb{Z}[\vec{X}]$, some g such that

$$f \in \mathrm{HTP}(R) \iff g \in \mathrm{HTP}(\mathbb{Q}).$$

So $\mathrm{HTP}(R) \leq_T \mathrm{HTP}(\mathbb{Q})$: there is an oracle Turing program that, if it runs with an oracle for $\mathrm{HTP}(\mathbb{Q})$, decides if its input f lies in $\mathrm{HTP}(R)$.

More broadly, for any oracle $A \subseteq \mathbb{N}$, the *A-computable functions* on $\mathbb{N}$ are those computed by an oracle Turing program Φ using oracle A. A set $B \subseteq \mathbb{N}$ is *A-computable*, or *Turing-reducible to A*, if χ_B is A-computable. (We write $B \leq_T A$.)

But this is only about (partial) functions from $\mathbb{N}$ to $\mathbb{N}$.....

A computable function on $2^{\mathbb{N}}$

An oracle Turing program, with one or more oracles, computes a partial function $: \mathbb{N} \to \mathbb{N}$. Here is a program with two oracles:

- Given the input $n \in \mathbb{N}$, move right to the n-th cell on each of the two oracle tapes.
- If the first oracle has a 1 there, print 1 as the output and halt.
- If the second oracle has a 1 there, print 1 as the output and halt.
- Otherwise, print 0 as the output and halt.

A computable function on $2^{\mathbb{N}}$

An oracle Turing program, with one or more oracles, computes a partial function $: \mathbb{N} \to \mathbb{N}$. Here is a program with two oracles:

- Given the input $n \in \mathbb{N}$, move right to the n-th cell on each of the two oracle tapes.
- If the first oracle has a 1 there, print 1 as the output and halt.
- If the second oracle has a 1 there, print 1 as the output and halt.
- Otherwise, print 0 as the output and halt.

For any two oracle sets A and B, this decides membership in $A \cup B$. We have $\Phi^{A \oplus B} = \chi_{A \cup B}$.

So we regard Φ as computing the function $F : 2^{\mathbb{N}} \times 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ with $F(A, B) = A \cup B$.

Computable functions on $2^{\mathbb{N}}$

Definition

A function $F : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is *computable* if there exists an oracle Turing program Φ such that, for every $X \in 2^{\mathbb{N}}$, the program Φ with the oracle X will compute the characteristic function $\chi_{F(X)}$ of $F(X)$:

$$\Phi^X = \chi_{F(X)} \qquad \text{(that is: } (\forall n)\ \Phi^X(n) = \chi_{F(X)}(n)\text{).}$$

For $F : (2^{\mathbb{N}})^k \to 2^{\mathbb{N}}$, just use an oracle Turing program designed for k oracle tapes.

The first example above showed that the (binary) union function $F : 2^{\mathbb{N}} \times 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ with $F(A, B) = A \cup B$ is computable.

Possibilities and impossibilities

The basic set-theoretic operations on subsets of $\mathbb{N}$ are computable.

We can collapse countably many oracles into one, using a *pairing function* that maps $(m, n) \in \mathbb{N}^2$ to a code $\langle m, n\rangle \in \mathbb{N}$ bijectively. Given oracles $A_0, A_1, \ldots$, we produce $\oplus_m A_m = \{\langle m, n\rangle \in \mathbb{N} : n \in A_m\}$.

Possibilities and impossibilities

The basic set-theoretic operations on subsets of $\mathbb{N}$ are computable.

We can collapse countably many oracles into one, using a *pairing function* that maps $(m, n) \in \mathbb{N}^2$ to a code $\langle m, n\rangle \in \mathbb{N}$ bijectively. Given oracles $A_0, A_1, \ldots$, we produce $\oplus_m A_m = \{\langle m, n\rangle \in \mathbb{N} : n \in A_m\}$.

However, the projection map $\oplus_m A_m \mapsto \{n \in \mathbb{N} : (\exists m) n \in A_m\}$ is not computable. If Φ computed it, then with all $A_m = \emptyset$ in the oracle, $\Phi^{\oplus_m \emptyset}(5)$ would eventually halt and output 0, having checked only finitely many elements $\langle m, 5\rangle$ in the oracle. Suppose it did not check whether $\langle 70, 5\rangle$ was in the oracle. Then we could create another oracle $B = \{\langle 70, 5\rangle\}$ on which $\Phi^B(5) = 0$, even though 5 lies in the projection of B.

Which functions $2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are computable?

Which functions $2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are computable?

Proposition

Every computable function F is continuous (for the Cantor topology).

Which functions $2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are computable?

Proposition

Every computable function F is continuous (for the Cantor topology).

Let Φ compute F. We show that $F^{-1}(\mathcal{U}_{\emptyset,\{k\}})$ is open.
If $F(X) \in \mathcal{U}_{\emptyset,\{k\}}$, then $\Phi^X(k)$ halts and outputs 0 after finitely many steps. During those steps, it examined only finitely much of the oracle X. Let $\sigma \subset X$ be the initial segment it examined, so $X \in \mathcal{U}_\sigma$.

Now whenever $W \in \mathcal{U}_\sigma$, $\Phi^W(k)$ will follow exactly the same steps as $\Phi^X(k)$ did, so will output 0. Thus $\mathcal{U}_\sigma \subseteq F^{-1}(\mathcal{U}_{\emptyset,\{k\}})$.

Which functions $2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are computable?

Proposition

Every computable function F is continuous (for the Cantor topology).

Let Φ compute F. We show that $F^{-1}(\mathcal{U}_{\emptyset,\{k\}})$ is open.
If $F(X) \in \mathcal{U}_{\emptyset,\{k\}}$, then $\Phi^X(k)$ halts and outputs 0 after finitely many steps. During those steps, it examined only finitely much of the oracle X. Let $\sigma \subset X$ be the initial segment it examined, so $X \in \mathcal{U}_\sigma$.

Now whenever $W \in \mathcal{U}_\sigma$, $\Phi^W(k)$ will follow exactly the same steps as $\Phi^X(k)$ did, so will output 0. Thus $\mathcal{U}_\sigma \subseteq F^{-1}(\mathcal{U}_{\emptyset,\{k\}})$.

The point is that Φ used only finitely much information from X to decide whether $k \in F(X)$; and basic open sets are defined by "containing some particular (fixed finite piece of) information."

Which functions $2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are continuous?

Plenty of noncomputable functions are continuous. For example, consider the constant function $F(X) = C$. If Φ computes this F, then $\Phi^{\emptyset} = C$. But $\Phi^{\emptyset}$ is computable! So even constant functions are mostly noncomputable.

Similarly, for a fixed noncomputable C, the continuous unary function $F(X) = C \cup X$ is not computable. It would require C itself as an oracle, along with the input X.

Which functions $2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are continuous?

Plenty of noncomputable functions are continuous. For example, consider the constant function $F(X) = C$. If Φ computes this F, then $\Phi^{\emptyset} = C$. But $\Phi^{\emptyset}$ is computable! So even constant functions are mostly noncomputable.

Similarly, for a fixed noncomputable C, the continuous unary function $F(X) = C \cup X$ is not computable. It would require C itself as an oracle, along with the input X.

Definition: *relative computability*

For any fixed $C \subseteq \mathbb{N}$, a function $F : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ is *C-computable* if there there exists an oracle Turing program Φ such that, for every $X \in 2^{\mathbb{N}}$, the program Φ with the oracle $C \oplus X$ will compute the characteristic function of $F(X)$:

$$\Phi^{C \oplus X} = \chi_{F(X)}.$$

Answer: which functions $2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are continuous!

Theorem

The continuous functions $F : 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are precisely the *relatively computable* functions: those such that there exists $C \subseteq \mathbb{N}$ for which F is C-computable.

C-computable functions are continuous, by the same proof as before.

For an arbitrary continuous F, use a coding to define two fixed oracles:

$$C = \{\langle \sigma, k \rangle : F(\mathcal{U}_\sigma) \subseteq \mathcal{U}_{\emptyset,\{k\}}\} \subseteq \mathbb{N}$$
$$D = \{\langle \tau, k \rangle : F(\mathcal{U}_\tau) \subseteq \mathcal{U}_{\{k\},\emptyset}\} \subseteq \mathbb{N}.$$

Then $k \in F(X)$ iff $X \in F^{-1}(\mathcal{U}_{\{k\},\emptyset})$ iff $(\exists \tau \subset X)\ \langle \tau, k \rangle \in D$;
while $k \notin F(X)$ iff $X \in F^{-1}(\mathcal{U}_{\emptyset,\{k\}})$ iff $(\exists \sigma \subset X)\ \langle \sigma, k \rangle \in C$.
So our Φ just searches in the $(C \oplus D \oplus X)$-oracle for one or the other.

Open sets

Certain open sets $\mathcal{V} \subseteq 2^{\mathbb{N}}$ are considered *effectively open*: these are of the form $\cup_{\sigma\in S}\mathcal{U}_\sigma$, where S is a computably enumerable set of finite binary strings.
The set $\mathcal{A}_f = \{W \subseteq \mathbb{P} : f \text{ has a solution in } \mathbb{Z}[W^{-1}]\}$ is an example.

Lemma

$\mathcal{V}$ is effectively open iff there exists an oracle Turing program Φ s.t.

$$X \in \mathcal{V} \iff \Phi^X(0) \text{ halts.}$$

Φ simply enumerates the strings in S (which is c.e.), and halts if it finds any $\sigma \in S$ with $\sigma \subset X$. For the converse, let $S = \{\sigma : \Phi^\sigma(0) \text{ halts}\}$.

Every open set $\mathcal{V}$ is "relatively effectively open," and vice versa.
Indeed, let $S = \{\sigma \in 2^{<\mathbb{N}} : \mathcal{U}_\sigma \subseteq \mathcal{V}\}$: then $\mathcal{V} = \{X : \Phi^{S\oplus X}(0) \text{ halts}\}$.

Other spaces

For subspaces, some additional functions become continuous. With the subspace $\{X \in 2^{\mathbb{N}} : X \text{ is infinite}\}$, it's safe for Φ to search for the n-th smallest element in its oracle X. Similarly, with the subspace of *generic* sets, it's safe to search for an initial segment that lies in a given open dense set.

Sometimes we take quotients of $2^{\mathbb{N}}$, under the quotient topology. If $\sim$ is an equivalence relation on $2^{\mathbb{N}}$ (or a subspace), an open set in $2^{\mathbb{N}}/\sim$ is the closure under $\sim$ of an open set in $2^{\mathbb{N}}$. A computable function Φ from $2^{\mathbb{N}}/\sim$ to $2^{\mathbb{N}}$ should run on every $X \in 2^{\mathbb{N}}$, and should satisfy

$$X \sim Y \implies \Phi^X = \Phi^Y.$$

This can be hazardous. There is a natural equivalence relation E_0, with $A \; E_0 \; B$ iff A and B have finite symmetric difference as sets. But the quotient topology on $2^{\mathbb{N}}/E_0$ is the indiscrete topology!

Measurable functions

Consider the functions

$$F(X) = \{n \in \mathbb{N} : X \text{ contains some multiple of } n\}.$$
$$G(X) = \{n \in \mathbb{N} : X \text{ contains infinitely many multiples of } n\}.$$

These are not continuous, but they are Borel-measurable. For F, there is an oracle Turing program Φ with $\chi_{F(X)} = \Phi^{X'}$. (Here X', the *jump of* X, is the Halting Problem for X-computable partial functions.)
For F, there is s Ψ with $\chi_{G(X)} = \Phi^{X''}$, using the *second jump* X'' of X.

Measurable functions

Consider the functions

$$F(X) = \{n \in \mathbb{N} : X \text{ contains some multiple of } n\}.$$
$$G(X) = \{n \in \mathbb{N} : X \text{ contains infinitely many multiples of } n\}.$$

These are not continuous, but they are Borel-measurable. For F, there is an oracle Turing program Φ with $\chi_{F(X)} = \Phi^{X'}$. (Here X', the *jump of* X, is the Halting Problem for X-computable partial functions.)
For F, there is s Ψ with $\chi_{G(X)} = \Phi^{X''}$, using the *second jump* X'' of X.

Borel-measurable functions $2^{\mathbb{N}} \to 2^{\mathbb{N}}$ are all *relatively α-jump computable*. This means that each such function can be computed if you are allowed some fixed oracle C and the oracle Turing program is given C and the α-th jump of X as oracles. (α can be any countable ordinal!)

Other *represented spaces*

- *Baire space* $\mathbb{N}^{\mathbb{N}}$ is the set $\{f : \mathbb{N} \to \mathbb{N}\}$, with a similar topology to Cantor space. Open sets are determined by finite information; other results mirror those for $2^{\mathbb{N}}$. But $\mathbb{N}^{\mathbb{N}}$ is not compact: the initial segments of length 1 partition the space into ∞-many open sets.

Other *represented spaces*

- *Baire space* $\mathbb{N}^\mathbb{N}$ is the set $\{f : \mathbb{N} \to \mathbb{N}\}$, with a similar topology to Cantor space. Open sets are determined by finite information; other results mirror those for $2^\mathbb{N}$. But $\mathbb{N}^\mathbb{N}$ is not compact: the initial segments of length 1 partition the space into ∞-many open sets.
- The usual topology on $\mathbb{R}$: think of Baire space as $\mathbb{Q}^\mathbb{N}$. Take the subspace of *fast-converging Cauchy sequences* $q_0 q_1 \ldots$, with $|q_k - \lim_n q_n| < 2^{-k}$ for all k, and mod out by the relation of having the same limit. This is the usual presentation of $\mathbb{R}$ used in computable analysis. The continuous functions $\mathbb{R} \to \mathbb{R}$ again coincide with the relatively computable functions.
- From $\mathbb{R}$ we get $[0, 1]$, $\mathbb{C}$, and various other spaces.

Other *represented spaces*

- *Baire space* $\mathbb{N}^\mathbb{N}$ is the set $\{f : \mathbb{N} \to \mathbb{N}\}$, with a similar topology to Cantor space. Open sets are determined by finite information; other results mirror those for $2^\mathbb{N}$. But $\mathbb{N}^\mathbb{N}$ is not compact: the initial segments of length 1 partition the space into ∞-many open sets.
- The usual topology on $\mathbb{R}$: think of Baire space as $\mathbb{Q}^\mathbb{N}$. Take the subspace of *fast-converging Cauchy sequences* $q_0 q_1 \ldots$, with $|q_k - \lim_n q_n| < 2^{-k}$ for all k, and mod out by the relation of having the same limit. This is the usual presentation of $\mathbb{R}$ used in computable analysis. The continuous functions $\mathbb{R} \to \mathbb{R}$ again coincide with the relatively computable functions.
- From $\mathbb{R}$ we get $[0, 1]$, $\mathbb{C}$, and various other spaces.
- The *Scott topology* on $2^\mathbb{N}$ has basis sets $\mathcal{U}_Y = \{X \subseteq \mathbb{N} : Y \subseteq X\}$. Open sets are given by finite positive information, and continuous functions are described by relativized *enumeration operators*. HTP itself can be seen as such an operator.

The HTP operator

The *HTP operator* is the map $\text{HTP} : \mathcal{P}(\mathbb{P}) \to \mathcal{P}(\mathbb{Z}[\vec{X}])$ given by:

$$f \in \text{HTP}(W) \iff f = 0 \text{ has a solution in } \mathbb{Z}[W^{-1}].$$

(We usually write $\text{HTP}(\mathbb{Z}[W^{-1}])$, not $\text{HTP}(W)$.)
Matiyasevich, Davis, Putnam, and Robinson showed that $HTP(\mathbb{Z})$ is noncomputable, indeed just as hard as the Halting Problem $\emptyset'$.
Therefore, the HTP operator is not computable. ($\Phi^{\emptyset}$ would have to compute $\text{HTP}(\emptyset)$, i.e., $\text{HTP}(\mathbb{Z})$.)

The HTP operator

The *HTP operator* is the map HTP : $\mathcal{P}(\mathbb{P}) \to \mathcal{P}(\mathbb{Z}[\vec{X}])$ given by:

$$f \in \text{HTP}(W) \iff f = 0 \text{ has a solution in } \mathbb{Z}[W^{-1}].$$

(We usually write $\text{HTP}(\mathbb{Z}[W^{-1}])$, not $\text{HTP}(W)$.)
Matiyasevich, Davis, Putnam, and Robinson showed that $HTP(\mathbb{Z})$ is noncomputable, indeed just as hard as the Halting Problem $\emptyset'$. Therefore, the HTP operator is not computable. ($\Phi^{\emptyset}$ would have to compute $\text{HTP}(\emptyset)$, i.e., $\text{HTP}(\mathbb{Z})$.)

The existence of nonempty boundary sets $\mathcal{B}_f$ shows that HTP is not continuous. $\mathbb{Z} = \mathbb{Z}[\emptyset^{-1}]$ contains no nontrivial solution to $X^2 + Y^2 = 1$, but no finite initial segment $\sigma = 000 \cdots 0$ of $\emptyset$ is sufficient to guarantee this. Therefore, HTP is not relatively computable: no single fixed oracle set (such as $\text{HTP}(\mathbb{Z})$ or $\text{HTP}(\mathbb{Q})$) allows one to compute $HTP(\mathbb{Z}[W^{-1}])$ from W uniformly for all $W \subseteq \mathbb{P}$.

How much oracle is needed?

The HTP operator can be computed with only a single jump, and with no additional oracle. The question, given a set $W \subseteq \mathbb{P}$ and a polynomial f, is whether a program (with a W-oracle) that searches through $\mathbb{Z}[W^{-1}]$ for a solution to $f = 0$ will ever find one (and halt), or whether it will search forever. The jump W', the Halting Problem for W-computable partial functions, includes the answer to this question.

The restriction of HTP to $\{\text{HTP-generic } W \subseteq \mathbb{P}\}$ is continuous, and is computable relative to a fixed oracle HTP($\mathbb{Q}$). We saw this in Lecture 1. In Baire category, this is a large subset of $2^{\mathbb{P}}$. In Lebesgue measure, we don't know whether it is large or small.

Enumeration operators

It can be quite productive to consider HTP using the Scott topologies on $\mathcal{P}(\mathbb{P})$ and $\mathcal{P}(\mathbb{Z}[\vec{X}])$. Here, the operator is given a list of the primes in W (not necessarily in order), and must output a list of the polynomials in HTP($\mathbb{Z}[W^{-1}]$) (in any order it likes). Only positive information goes in, and only positive information comes out.

The HTP operator is continuous w.r.t. the Scott topologies, and requires no additional oracle.

Enumeration operators

It can be quite productive to consider HTP using the Scott topologies on $\mathcal{P}(\mathbb{P})$ and $\mathcal{P}(\mathbb{Z}[\vec{X}])$. Here, the operator is given a list of the primes in W (not necessarily in order), and must output a list of the polynomials in HTP($\mathbb{Z}[W^{-1}]$) (in any order it likes). Only positive information goes in, and only positive information comes out.

The HTP operator is continuous w.r.t. the Scott topologies, and requires no additional oracle. This is the key to several recent results.

Theorem (M., using results of Jockusch and of Kurtz)

Almost all subrings $R \subseteq \mathbb{Q}$ have the property that R' is R-computably enumerable, but not diophantine in the ring R. This fails only on a meager set of measure 0. (So the MDPR result for $\mathbb{Z}$ is anomalous.)

Theorem (Kramer-M.)

There exist subrings R and S of $\mathbb{Q}$ such that $R <_T S$, yet HTP(S) $<_T$ HTP(R), with strict Turing reducibility for both.