GOSTS Descriptive Set Theory

Computability and the Lightface Hierarchies

James Holland

2020-11-11

Computation on ω

The usual notion of computability applies to $f : \omega \to \omega$ (i.e. to reals) rather than functions $f : \mathcal{N} \to \mathcal{N}$.

Definition

- f is a *partial function* from A to B ($f : A \rightharpoonup B$ iff $f : \text{dom}(f) \to B$ and $\text{dom}(f) \subseteq A$.
- A function $f : \omega \rightharpoonup \omega$ is *computable* iff there's some program $e \in \omega$ such that the function it computes, $[\![e]\!] = f$.
- A subset $A \subseteq \omega$ is *computable* iff the characteristic function $\chi_A : \omega \to 2$ is computable, defined by

$$\chi_A(n) = \begin{cases} 1 & \text{if } n \in A \\ 0 & \text{if } n \notin A. \end{cases}$$

How do we generalize this?

Computation on ω

We have the following *arithmetical hierarchy* of complexity of subsets of ω.

Definition

- $A \subseteq \omega$ is $\Sigma_1^{0,\omega}$ iff $A = \exists^\omega B$ for some computable $B \subseteq \omega \times \omega$.
- $A \subseteq \omega$ is $\Sigma_{n+1}^{0,\omega}$ iff $A = \exists^\omega B$ for some $B \in \Pi_n^{0,\omega}$;
- $\Pi_n^{0,\omega} = \neg\Sigma_n^{0,\omega}$.
- $\Delta_n^{0,\omega} = \Sigma_n^{0,\omega} \cap \Pi_n^{0,\omega}$.

$$\text{Computable sets} \to \begin{array}{ccccccc} & & \Sigma_1^{0,\omega} & & & \Sigma_2^{0,\omega} & \\ & \subsetneq & & \subsetneq \quad \subsetneq & & & \subsetneq \\ \Delta_1^{0,\omega} & & & \Delta_2^{0,\omega} & & & \cdots \\ & \subsetneq & & \subsetneq \quad \subsetneq & & & \subsetneq \\ & & \Pi_1^{0,\omega} & & & \Pi_2^{0,\omega} & \end{array}$$

(We can also go beyond this, but don't worry about it.)

Computation on ω

Theorem

A set $A \subseteq \omega$ is computable iff $A, \neg A \in \Sigma_1^{0,\omega}$, i.e. $A \in \Delta_1^{0,\omega}$.

Proof.

We have computable $R, P \subseteq \omega \times \omega$ where

$$\begin{aligned} x \in A &\quad \text{iff} \quad \exists y\ R(x,y) \\ x \notin A &\quad \text{iff} \quad \exists y\ P(x,y) \end{aligned}$$

So to see whether $x \in A$ or not, the program to compute χ_A will be:

- Take n, and see whether $R(x,n)$ or $P(x,n)$.
- If $R(x,n)$, then $x \in A$ so return 1.
- If $P(x,n)$, then $x \notin A$ so return 0.
- If neither holds, consider $n+1$ and repeat.

Since $x \in A$ or $x \notin A$, we always will get an n where $R(x,n)$ or $P(x,n)$ so this program will always (eventually) terminate for any $x \in \omega$. ⊣

Computation on ω

As one would expect, we have the following closure properties: for $A, B \in \Sigma_1^{0,\omega}$,

- $A \cap B$, $A \cup B$ are $\Sigma_1^{0,\omega}$.
- $\forall^{<\omega} A = \{\langle n, m\rangle : \exists x < m \operatorname{code}(n, x) \in A\}$ is $\Sigma_1^{0,\omega}$ for every x (i.e. bounded quantification)
- $\exists^\omega A = \{n : \exists m \;\operatorname{code}(n, m) \in A\}$ is $\Sigma_1^{0,\omega}$.
- $f^{-1}{}''A$ is $\Sigma_1^{0,\omega}$ for $f : \omega \rightharpoonup \omega$ computable.

Proof.

$x \in f^{-1}{}''A$ iff $\exists y(\langle x, y\rangle \in f \wedge y \in A)$. If $f : \omega \rightharpoonup \omega$ is computable, then the graph of f is $\Sigma_1^{0,\omega}$ (asking whether there's a code of a computation starting with x and ending with y).

$$\underbrace{\exists y(\underbrace{\langle x, y\rangle \in f}_{\Sigma_1^{0,\omega}} \wedge \underbrace{y \in A}_{\Sigma_1^{0,\omega}})}_{\Sigma_1^{0,\omega}}$$

⊣

Computation on ω

Really we just want to generalize $\Sigma_1^{0,\omega}$.

Theorem

For $A \subseteq \omega$, the following are equivalent:

1. $A = \exists^\omega B$ *for some computable* B.
2. $A = \operatorname{dom} f$ *for some computable* $f : \omega \rightharpoonup \omega$;
3. $A = \operatorname{im} f$ *for some computable* $f : \omega \to \omega$;
4. $A = \operatorname{im} f$ *for some computable* $f : \omega \rightharpoonup \omega$;
5. $|A| < \aleph_0$ *or* $A = \operatorname{im} f$ *for some injective computable* $f : \omega \to \omega$.

We are most interested in (3), because if $\{n\} \subseteq \omega$ is open for every $n \in \omega$, then we get the following.

Corollary

$A \subseteq \omega$ is $\Sigma_1^{0,\omega}$ iff there is some computable $f : \omega \to \omega$ where $A = \bigcup_{n\in\omega} N_{f(n)}$ where the neighborhood $N_x = \{x\}$.

Computation on $\mathcal{N}$

For $\underset{\sim}{\mathcal{N}}$ and other polish spaces, we don't start with the definition of a computable function but instead with computable subsets.

Definition

A set $X \subseteq \mathcal{N}$ is Σ_1^0 iff there is some computable $f : \omega \to {}^{<\omega}\omega$ where

$$X = \bigcup_{n \in \omega} \mathcal{N}_{f(n)}.$$

This works with ω in two main ways:

- The previous result with $\{f(n)\}$ in place of $\mathcal{N}_{f(n)}$;
- All Σ_1^0 sets are "simple", i.e. open.

For ω as a polish space, all subsets are open, so the arithmetical hierarchy refines this, giving a kind of computable topology. So that's the motivating idea behind other polish spaces.

Computation on $\mathcal{N}$

Another way to motivate this definition is that to determine whether $x \in \mathcal{N}$ is in our set $A \subseteq \mathcal{N}$, there needs to exist an $n \in \omega$ such that $f(n) \lhd x$. In other words, whereas $\Sigma_1^{0,\omega}$-predicates over ω satisfy

$$P(x) \quad \text{iff} \quad \exists y \in \omega\ (x = f(y))$$

Σ_1^0-predicates over $\mathcal{N}$ satisfy

$$P(x) \quad \text{iff} \quad \exists y \in \omega\ (x \in \mathcal{N}_{f(y)}).$$

Due to the many equivalences of $\Sigma_1^{0,\omega}$-sets, we get the following.

Result

For $X \subseteq \mathcal{N}$, X is Σ_1^0 iff there is a computable $R \subseteq {}^{<\omega}\omega$ where

$$x \in X \quad \textit{iff} \quad \exists n < \omega\ R(x \restriction n).$$

Computation on $\mathcal{N}$

Due to the many equivalences of $\Sigma^{0,\omega}_1$-sets, we get the following.

Result

For $X \subseteq \mathcal{N}$, X is Σ^0_1 iff there is a computable $R \subseteq {}^{<\omega}\omega$ where

$$x \in X \quad \textit{iff} \quad \exists n < \omega\ R(x \upharpoonright n).$$

Proof.

- If $X \in \Sigma^0_1$ then $X = \bigcup_{\tau \in R} \mathcal{N}_\tau$ for some $R \in \Sigma^{0,\omega}_1$. Thus

$$\begin{aligned} x \in X &\leftrightarrow \exists \tau \in R\ (x \in \mathcal{N}_\tau) \\ &\leftrightarrow \exists n < \omega\ (x \upharpoonright n \in R). \end{aligned}$$

- For the converse, if $x \upharpoonright n \in R$ then $\mathcal{N}_{x \upharpoonright n} \subseteq \bigcup_{\tau \in R} \mathcal{N}_\tau = X$ shows X is Σ^0_1. ⊣

Computation on $\mathcal{N}$

We can also relativize with parameters.

Definition

X is $\Sigma_1^0(A)$ for $A \subseteq \mathcal{N}$ iff $X = \bigcup_{n\in\omega} \mathcal{N}_{f(n)}$ for some $\vec{a} \in A^{<\omega}$ and $\vec{a}$-computable $f : \omega \to {}^{<\omega}\omega$.

This tells us that we really have refined the open sets. (Note we'll often write $\Sigma_1^0(a)$ for $\Sigma_1^0(\{a\})$ if $a \in \mathcal{N}$.)

Corollary

$\utilde{\Sigma}_1^0 = \bigcup_{x\in\mathcal{N}} \Sigma_1^0(x) = \Sigma_1^0(\mathcal{N})$.

Proof.

- Clearly anything $\Sigma_1^0(a)$ is open as the union of cones.
- If $X \subseteq \mathcal{N}$ is $\utilde{\Sigma}_1^0$, then $X = \bigcup_{\tau\in A} \mathcal{N}_\tau$ for some $A \subseteq {}^{<\omega}\omega$
- Any enumeration of A is itself A-computable so X is $\utilde{\Sigma}_1^0(A)$. ⊣

Computation on $\mathcal{N}$

With Σ^0_1-sets more-or-less solidified, we have that computable sets are precisely the Δ^0_1-sets, i.e. sets which are Σ^0_1 and their complements are Σ^0_1.

Definition

$X \subseteq \mathcal{N}$ is *computable* iff $X, \neg X \in \Sigma^0_1$.

Since all Σ^0_1-sets are open, this means all computable sets are clopen.

If we are going to say what it means for a *function* $f : \mathcal{N} \to \mathcal{N}$ to be computable, we need to be careful.

Computation on $\mathcal{N}$

Consider the following theorem from computation on ω:

Theorem

$f : \omega \rightharpoonup \omega$ is computable iff the graph of f is $\Sigma^{0,\omega}_1$.

The obvious generalization of this is that $f : \mathcal{N} \to \mathcal{N}$ is computable iff the graph of f is Σ^0_1.

- This will never be true: no function regarded as a relation is open.
- Nevertheless, there are several equivalent notions that *do* work.
- The most *practical* definition is the following.

Theorem

$f : \mathcal{N} \to \mathcal{N}$ is computable iff there's an $e \in \omega$ where $f(x) = [\![e]\!]^x$.

Here $[\![e]\!]^x$ is the function calculated by the program e with x as an oracle.

In other words, f is computable iff $f(x) \in \mathcal{N}$ is computable from $x \in \mathcal{N}$ and the way it's computed is uniform across all x.

Computation on $\mathcal{N}$

- We can similarly relativize this notion.
- f is A-computable iff $f(x) = [\![e]\!]^{\vec{a},x}$ for some $\vec{a} \in A^{<\omega}$.
- This idea is useful because it then breaks down continuous functions just like the relativizations $\Sigma^0_1(A)$ break down $\undertilde{\boldsymbol{\Sigma}}^0_1$-sets.

Result

$f : \mathcal{N} \to \mathcal{N}$ is continuous iff f is a-computable for some $a \in \mathcal{N}$.

Proof.

- If f is continuous, we can consider $a \in \mathcal{N}$ coding the set

$$\{\langle \tau, \sigma \rangle : f^{-1}{}''\mathcal{N}_\sigma \subseteq \mathcal{N}_\tau\} \subseteq {}^{<\omega}\omega \times {}^{<\omega}\omega.$$

  So f is a-computable: we can calculate $f(x)(n)$ just by finding a sufficiently large σ with a $\tau \lhd x$ and returning $\sigma(n)$.
- If f is a-computable, to calculate $f(x) \restriction n$, we only use finitely many values of x in our computation, meaning some $m \in \omega$ has $\mathcal{N}_{x \restriction m} \subseteq f^{-1}{}''\mathcal{N}_{f(x) \restriction n}$ so f is continuous. $\dashv$

Computation on other Polish Spaces

- This definition doesn't really generalize to other polish spaces $\underset{\sim}{\mathcal{M}}$ since there isn't inherently a notion of $x \in \mathcal{M}$ being computable like for $x \in \mathcal{N}$.
- Additionally, while every polish space has a countable basis $\{\mathcal{M}_n : n \in \omega\}$, there's no reason to think that the $\Sigma_1^{0,\mathcal{M}}$-sets coming from this would make a whole lot of sense: $\mathcal{M}_n \cap \mathcal{M}_m$ will be $\bigcup_{i \in A} \mathcal{M}_i$ for maybe a very complicated $A \subseteq \omega$.
- This motivates *recursive presentations* of polish spaces mentioned in the literature which just imposes some restrictions on $n \mapsto \mathcal{M}_n$ so that arguments work similarly to $\mathcal{N}$ or ω.

Definition

Let $\underset{\sim}{\mathcal{M}}, \underset{\sim}{\mathcal{W}}$ be polish spaces. $f : \mathcal{M} \to \mathcal{W}$ is *computable* iff the *neighborhood graph* $\mathrm{NG}_f \subseteq \mathcal{M} \times \omega$ is $\Sigma_1^{0,\mathcal{M}\times\omega}$:

$$\mathrm{NG}_f = \{\langle x, \sigma\rangle \in \mathcal{M} \times \omega : f(x) \in \mathcal{W}_\sigma\}.$$

This also meshes with computable $f : \omega \to \omega$ being Σ_1^0 as relations simply because $\mathrm{NG}_f = f$ in this case.

Computation on other Polish Spaces

In the end, we get $\Sigma_1^{0,\mathcal{M}}(A)$-sets are closed under

- Finite intersections, finite unions, bounded quantification
- A-computable preimages.
- Existential quantification over ω (i.e. projection from $\mathcal{M} \times \omega$ to $\mathcal{M}$)

The A-computable functions over $\mathcal{N}$ will include all the functions you'd expect:

- A-computable $f : \omega \to \omega$ (i.e. A-computable constants in $\mathcal{N}$)
- Projections ($f(x, \vec{y}) = x$)
- Simple codings like $\langle x, y \rangle \mapsto x * y$
- Compositions of other A-computable functions

The Hyperarithmetical Hierachy

- So how do we define the rest of the hierarchy?
- There are two ways to proceed: define
  1. Σ^0_n for $n < \omega$; or
  2. Σ^0_α for $\alpha < \omega_1^{\mathrm{CK}}$.
- I'll make a short mention of Σ^0_α for $\alpha < \omega_1^{\mathrm{CK}}$ and then just prove things about Σ^0_n for $n < \omega$.
- The basic properties generalize easily enough but require more technical work.
- The motivating idea is that Σ^0_1-sets are the "computable unions" of basic open sets: sets of the form $\bigcup_{n\in\omega} \mathcal{N}_{x(n)}$ for some computable $x \in \mathcal{N}$. From here, we can take complements and unions as with the borel hierarchy.
- This means creating a hierarchy of *codes* that tell us how a set is built up from basic open sets, unions, and complements.

The Hyperarithmetical Hierarchy

Definition

The set $\mathrm{BC} \subseteq \mathcal{N}$ of *Borel codes* is defined recursively in a hierarchy. Through coding, we think of $x \in \mathrm{BC}$ as an element of $\omega \times {}^{\omega}\mathcal{N}$, writing

$$x = \langle x(0) \in \omega, x_i \in \mathcal{N} : i \in \omega \rangle.$$

- x is $\mathrm{c}\Sigma^0_1$ iff $x(0) \notin \{0, 1\}$;
- x is $\mathrm{c}\Pi^0_\alpha$ iff $x(0) = 0$ and $x_0 \in \mathrm{c}\Sigma^0_\alpha$;
- x is $\mathrm{c}\Sigma^0_\alpha$ for $\alpha > 1$ iff $x(0) = 1$ and $x_i \in \bigcup_{\xi<\alpha} \mathrm{c}\Sigma^0_\xi \cup \mathrm{c}\Pi^0_\xi$ for each $i < \omega$.

We set $\mathrm{BC} = \bigcup_{\alpha<\omega_1} \mathrm{c}\Sigma^0_\alpha$.

We can then interpret these Borel codes iteratively.

Definition

For $x = \langle x(0), x_i : i \in \omega \rangle \in \mathrm{BC}$,

- If $x(0) \notin \{0, 1\}$, $B_x = \bigcup_{x_0(n)=1} \mathcal{N}_n$;
- If $x(0) = 0$, $B_x = \mathcal{N} \setminus B_{x_0}$;
- If $x(0) = 1$, then $B_x = \bigcup_{i<\omega} B_{x_i}$.

The Hyperarithmetical Hierarchy

The main result is that these indeed give us the Borel hierarchy (proof by induction, Appendix Corollary B1 • 3 in the notes).

Result

For $\alpha < \omega_1$, $\utilde{\Sigma}^0_\alpha = \{B_x : x \in \mathrm{c}\Sigma^0_\alpha\}$ *and similarly for* Π *and* Δ.

This then easily allows us to define the lightface Σ^0_α just by considering those codes which are computable.

Definition

For $X \subseteq \mathcal{N}$, $\alpha < \omega_1$

- X is Σ^0_α iff X has a computable borel code $x \in \mathrm{c}\Sigma^0_\alpha \subseteq \mathcal{N}$;
- X is Π^0_α iff X has a computable borel code $x \in \mathrm{c}\Pi^0_\alpha \subseteq \mathcal{N}$;
- X is Δ^0_α iff X is Σ^0_α and Π^0_α.

- This hierarchy is sometimes called the "lightface Borel hierarchy", sometimes the "hyperarithmetical hierarchy".
- While defined for $\alpha < \omega_1$, the hierarchy only has length ω_1^{CK}, the supremum of ordertypes $\alpha < \omega_1$ with computable relations $\langle \omega, R\rangle \cong \langle \alpha, \in\rangle$.

The Hyperarithmetical Hierarchy

- There are also other ways to define ω_1^{CK}, namely with Kleene's $\mathcal{O}$, which more-or-less a way of trying to computably name ordinals:
- ω_1^{CK} is the set of ordinals with names in $\mathcal{O}$.
- A third way to define ω_1^{CK} is with *admissible sets*:
- ω_1^{CK} is the least ordinal $> \omega$ with L_α admissible.
- "Admissible" here just means a model of KP (Kripke–Platek) set theory.
- We can generalize the arithmetical hierarchy (about to be defined) to the hyperarithmetical hierarchy in terms of iterated jumps.
- But taking this strategy would require defining Kleene's $\mathcal{O}$, which again requires more coding and technical details to work with.
- So let's just work with Σ_n^0 for $n < \omega$, which has more direct connections to definability anyway.

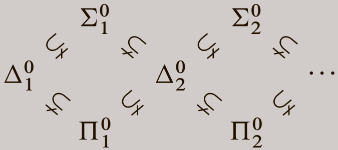
The Arithmetical Hierarchy

Definition

For $X \subseteq \mathcal{N}$,

- X is Σ^0_1 iff X is the computable union of basic open sets.
- X is Π^0_n iff $\mathcal{N} \setminus X$ is Σ^0_n.
- X is Σ^0_{n+1} iff $X = \exists^\omega Y$ for some $Y \in \Pi^0_n$ regarding $Y \subseteq \mathcal{N} \times \omega$.
- X is Δ^0_n iff X is Σ^0_n and Π^0_n.

This defines the *arithmetical hierarchy*, and these pointclasses are the *arithmetical pointclasses* with sets in them *arithmetical*.



This same definition works for ω as well as $\mathcal{C}$, $\mathcal{N}^2$, $\mathcal{N}^3$, etc.

The Arithmetical Hierarchy

To further motivate why this gives the computable analogue of topology, we have the following.

Result

$X \subseteq \mathcal{N}$ is Π^0_1 iff $X = [T]$ for some computable *tree $T \subseteq {}^{<\omega}\omega$.*

Proof.

$x \in [T]$ iff $\forall n < \omega\ (x \restriction n \in T)$. The negation of this is $\exists n < \omega\ (x \restriction n \in T)$ which is therefore a Σ^0_1-relation since membership in T is computable. Hence X is Π^0_1.

If X is Π^0_1, then there's some computable R where $x \in X$ iff $\forall n \in \omega\ R(x \restriction n)$. So take the tree

$$T = \{\tau \in R : \forall \sigma \lhd \tau\ R(\sigma)\}.$$

T is computable since R is and $[T] = X$. ⊣

We could attempt to characterize the rest of the hierarchy in this sort of way, but it's much better to phrase things in terms of relations.

The Arithmetical Hierarchy

Result

$X \subseteq \mathcal{N}$ is Σ^0_1 iff there's a computable $R \subseteq {}^{<\omega}\omega$ where

$$x \in X \quad \textit{iff} \quad \exists n < \omega\ R(x \restriction n).$$

What do the sets in the rest of the hierarchy look like? Note that

$$\Sigma^0_{n+1} = \exists^\omega \Pi^0_n = \exists^\omega \neg \Sigma^0_n.$$

Using the above result, we have (for example) for any Σ^0_3-set X,

$$\begin{aligned} x \in X \quad & \text{iff} \quad \exists m_2 \in \omega\ \neg P_2(x, m_2) \text{ for some } P_2 \in \Sigma^0_2,\ P_2 \subseteq \mathcal{N} \times \omega \\ & \text{iff} \quad \exists m_2 \in \omega\ \neg \exists m_1 \in \omega\ \neg P_1(x, m_1, m_2) \\ & \qquad\qquad \text{for some } P_1 \in \Sigma^0_1,\ P_1 \subseteq \mathcal{N} \times \omega^2 \\ & \text{iff} \quad \exists m_2 \in \omega\ \forall m_1 \in \omega\ \exists n \in \omega\ P_0(x \restriction n, m_1, m_2) \\ & \qquad\qquad \text{for some computable } P_0 \subseteq {}^{<\omega}\omega \times \omega^2. \end{aligned}$$

The analogy between the arithmetical hierarchy and the Lévy hierarchy of formulas (quantifying over ω) should then be clear.

The Arithmetical Hierarchy

Showing the usual inclusions is actually quite annoying, mostly amounting to translating the topological ideas in a way that ensures everything is computable when playing with the basic open sets. So let's cut to the chase:

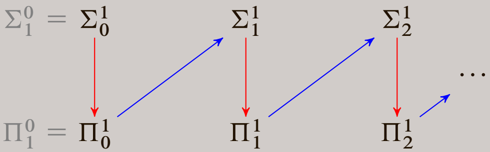
- $\Delta^0_n \subseteq \Sigma^0_n \subseteq \Delta^0_{n+1}$ and similarly for Π;
- All arithmetical pointclasses are closed under
  - finite unions, finite intersections;
  - bounded quantification; and
  - computable preimages.
- Σ^0_n is closed under $\exists^\omega$;
- Π^0_n is closed under $\forall^\omega$;
- Δ^0_n is closed under $\neg$.

We can show the inclusions are proper in just the same way as before.

- There's a Σ^0_n-universal set showing $\Delta^0_n \subsetneq \Sigma^0_n$;
- There's a Π^0_n-universal set showing $\Sigma^0_n \subsetneq \Delta^0_{n+1}$.

The Analytical Hierarchy

- The analytical hierarchy is the lightface version of the projective hierarchy.
- It's generated in precisely the same way as the projective hierarchy, but it starts with the Π^0_1 sets instead of all closed sets.



Projections are in blue
Complements are in red

The Analytical Hierarchy

- Both the arithmetical and analytical hierarchies use projections:
  - the arithmetical hierarchy projects from products with ω,
  - but the analytical hierarchy projects from products with $\mathcal{N}$.
- Projection from products with $\mathcal{N}$ will correspond to existential quantification over $\mathcal{N}$: $\mathfrak{p}Y$ for $Y \subseteq X \times \mathcal{N}$ will be written $\exists^{\mathcal{N}} Y$.

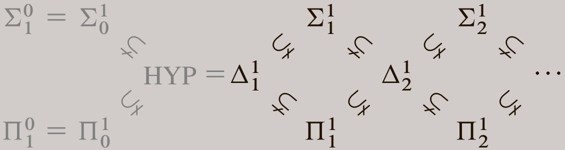
Definition

For $X \subseteq \mathcal{N}$ and $n < \omega$,

- X is Σ^1_0 iff X is Σ^0_1.
- X is Π^1_n iff $\mathcal{N} \setminus X$ is Σ^1_n.
- X is Σ^1_{n+1} iff $X = \exists^{\mathcal{N}} Y$ for some $Y \in \Pi^1_n$ regarding $Y \subseteq \mathcal{N} \times \mathcal{N}$.
- X is Δ^1_n iff X is both Σ^1_n and Π^1_n.

These are the *analytical* pointclasses and sets in them *analytical* (which is different from being analytic i.e. $\utilde{\Sigma}^1_1$).

The Analytical Hierarchy



Here HYP is the hyperarithmetical sets, analogous to $\mathcal{B} = \underset{\sim}{\mathbf{\Delta}}^1_1$.

We can also relativize this hierarchy to $A \subseteq \mathcal{N}$ by starting with $\Sigma^0_1(A)$-sets. This then easily yields the following by induction.

Corollary

For $n < \omega$, $\underset{\sim}{\mathbf{\Sigma}}^1_n = \bigcup_{a \in \mathcal{N}} \Sigma^1_n(a) = \Sigma^1_n(\mathcal{N})$.

The Analytical Hierarchy

Just as with $\underset{\sim}{\boldsymbol{\Sigma}}^1_1$, we have many different characterizations of Σ^1_1-sets. The following are all equivalent (18 C • 6 in the notes)

- $X = \{x \in \mathcal{N} : \exists y \in \mathcal{N}\ (x * y \in Y)\}$ for some $Y \in \Pi^0_1$.
- $X = \exists^{\mathcal{N}} Y$ for some $Y \in \Pi^{0,\mathcal{N}\times\mathcal{N}}_1$.
- $X = \exists^{\mathcal{M}} Y$ for some polish $\underset{\sim}{\mathcal{M}}$ with recursive presentation yielding $Y \in \Pi^{0,\mathcal{N}\times\mathcal{M}}_1$.
- $X = \exists^{\mathcal{N}} Y$ for some arithmetical $Y \subseteq \mathcal{N} \times \mathcal{N}$.
- $X = \exists^{\mathcal{N}} Y$ for some hyperarithmetical $Y \subseteq \mathcal{N} \times \mathcal{N}$.
- $X = \operatorname{im} f$ for some computable $f : \mathcal{N} \to \mathcal{N}$.

Proving these relies on some of the closure properties of Σ^1_1-sets, which should be expected from the closure properties of $\underset{\sim}{\boldsymbol{\Sigma}}^1_1$-sets.

The Analytical Hierarchy

Result

For $0 < n < \omega$,

- Σ^1_n *is closed under* $\cap$, $\cup$, $\exists^\omega$, $\forall^\omega$, *computable preimages, and* $\exists^{\mathcal{N}}$;
- Π^1_n *is closed under* $\cap$, $\cup$, $\exists^\omega$, $\forall^\omega$, *computable preimages, and* $\forall^{\mathcal{N}}$;
- Δ^1_n *is closed under* $\cap$, $\cup$, $\exists^\omega$, $\forall^\omega$, *computable preimages, and* $\neg$.

Proof.

- For $n = 1$, X is Σ^1_1 iff $X = \exists^{\mathcal{N}} Y$ for some $Y \in \Pi^0_1$.
- So there's some computable R where

$$x \in X \quad \text{iff} \quad \exists^{\mathcal{N}} y \forall^\omega n \; R(x \restriction n, y \restriction n)$$

- So if $X_0, X_1 \in \Sigma^1_1$, then $x \in X_0 \cap X_1$ iff

$$\begin{aligned} &\text{iff} \quad \exists^{\mathcal{N}} y_0 \exists^{\mathcal{N}} y_1 \; \forall^\omega n \; (R_0(x \restriction n, y_0 \restriction n) \wedge R_1(x \restriction n, y_1 \restriction n)) \\ &\text{iff} \quad \exists^{\mathcal{N}} y \; \forall^\omega n \; (y = y_0 * y_1 \wedge R(x \restriction n, y_0 \restriction n) \wedge R(x \restriction n, y_1 \restriction n)). \end{aligned}$$

The Analytical Hierarchy

Result

Σ^1_1 *is closed under* $\cap$, $\cup$, $\exists^\omega$, $\forall^\omega$, *and* $\exists^\mathcal{N}$;

Proof.

- For $\exists^\omega$, if X is $\Sigma^{1,\mathcal{N}\times\omega}_1$, $X = \exists^\mathcal{N} Y$ so that

$$\begin{aligned} x \in \exists^\omega X \quad &\text{iff} \quad \exists^\omega m\, \exists^\mathcal{N} y\, \forall^\omega n\ R_0(x\restriction n, y\restriction n, m) \\ &\text{iff} \quad \exists^\mathcal{N} y\, \forall^\omega n\ (y = \langle y_0\rangle^\frown y' \wedge R_0(x\restriction n, y'\restriction n, y_0)). \end{aligned}$$

- Dealing with $\forall^\omega$ is slightly trickier, but basically relies on the same idea by identifying in a computable way $\mathcal{N}$ with ${}^\omega\mathcal{N}$: write $y \in \mathcal{N}$ as $\langle y_i : i \in \omega\rangle \in {}^\omega\mathcal{N}$: thus $x \in \forall^\omega X$ iff

$$\begin{aligned} &\text{iff} \quad \forall^\omega m\, \exists^\mathcal{N} y\, \forall^\omega n\ R(x\restriction n, y\restriction n, m) \\ &\text{iff} \quad \exists^\mathcal{N} y\, \forall^\omega n\ (n = \text{code}(n_0, n_1) \wedge R(x\restriction n_0, y_{n_1}\restriction n_0, n_1)). \end{aligned}$$

⊣

The Analytical Hierarchy

The inclusions also aren't too difficult to show just by adding unnecessary quantifiers and a bit of coding.

Result (18 C • 5)

$\Delta^1_n \subseteq \Sigma^1_n \subseteq \Delta^1_{n+1}$ *and similarly for* Π.

To show that the inclusions are strict, we as usual use universal sets: for each $n < \omega$,

- There's a Σ^1_n-universal set, showing $\Delta^1_n \subsetneq \Sigma^1_n$;
- There's a Π^1_n-universal set, showing $\Sigma^1_n \subsetneq \Delta^1_{n+1}$.

The Lévy Hierarchy

Definition

For $\mathscr{L}$ a language with some sort of “special” relation symobl $<$, a formula φ is

- $\Sigma_0 = \Pi_0$ iff all quantifiers of φ are bounded;
- Σ_{n+1} iff φ is of the form $\exists x\ \psi$ for some Π_n-formula ψ;
- Π_n iff φ is of the form $\neg\psi$ for some Σ_n-formula ψ.

This defines the Lévy hierarchy of formulas and subsequently the hierarchy of Σ_n-definable sets of any given model. Recall the following result about computability over ω.

Theorem (Appendix A3 b • 9)

A set $X \subseteq \omega$ is $\Sigma_n^{0,\omega}$ iff X is Σ_n-definable over $\mathbf{N} = \langle \omega, 0, 1, +, \cdot \rangle$.

This is nice because it explicitly identifies the arithmetical pointclasses with the Lévy hierarchy.

Second-Order Logic

For first-order logic, definable sets of a model **M** are elements of $\mathcal{P}(M)$:

$$\varphi(\mathbf{M}) = \{x \in M : \mathbf{M} \models \varphi(x)\} \subseteq M$$

If we want to talk about definable subsets of $\mathcal{N}$, we need to either talk about some sort of first-order structure with universe $\mathcal{N}$, or else a higher-order logic.

Question

Why higher-order logic?

- In the end, we want to identify Σ_n-definable sets with Σ_n^0-sets.
- (And with parameters, $\Sigma_n(A)$-definable sets with $\Sigma_n^0(A)$-sets.)
- In this way, Σ_n-definable sets with parameters will be precisely the $\utilde{\Sigma}_n^0$-sets.

Let's consider $\{x\}$ for some $x \in \mathcal{N}$.

- This will be $\utilde{\Pi}_1^0$.
- But $\{x\}$ would be $\Sigma_0(\mathcal{N})$-definable with parameters for *any* first-order model with universe $\mathcal{N}$.
- So this first-order logic approach would need to do away with equality.

Second-Order Logic

Higher-order logic makes more sense because it allows for more quantifiers corresponding to projection over different spaces.

Definition

The *variables* of second-order logic are

- v_n for $n < \omega$ (individual variables), and
- P^i_n for $i, n < \omega$ (predicate variables).

Formulas are defined exactly the same as with first order logic except that predicate variables can take inputs.

Definition

φ is a second-order logic formula iff

- φ is "$x = y$";
- φ is "$R(\vec{x})$" for some relation symbol R and terms $\vec{x}$;
- φ is "$P^i_n(\vec{x})$" for $\vec{x}$ a sequence of i terms;
- φ is "$\neg\psi$" or "$\psi \wedge \theta$" for formulas ψ, θ;
- φ is "$\forall X\ \psi$" for ψ a formula and X either kind of variable.

Second-Order Logic

For any first-order model **M**, we can expand to a second-order model $\underset{\sim}{\mathcal{M}}$ by interpretting, for $\vec{x} \in M^{<\omega}$ and $P_0 \in \mathcal{P}(M^{<\omega})$,

$$\underset{\sim}{\mathcal{M}} \models \text{“}P_0(\vec{x})\text{”} \quad \text{iff} \quad \vec{x} \in P_0$$

$$\underset{\sim}{\mathcal{M}} \models \text{“}\forall P_n^i\, \varphi(P_n^i, \vec{x})\text{”} \quad \text{iff} \quad \underset{\sim}{\mathcal{M}} \models \text{“}\varphi(P_0, \vec{x})\text{” for every } P_0 \in \mathcal{P}(M^{<\omega}).$$

This yields $\underset{\sim}{\mathcal{N}}$ as the second-order expansion of $\mathbf{N} = \langle \omega, 0, 1, +, \cdot \rangle$.

We will identify elements of $\mathcal{P}(\omega^{<\omega})$ instead as elements of $\mathcal{N} = {}^{\omega}\omega$. We could also just say that the variable $P \in \mathcal{P}(\omega^2)$ defines a function, or something to this effect.

Question

What happens with the Lévy hierarchy in second-order logic?

We need to expand to allow for multiple kinds of quantifiers.

Second-Order Logic

Definition

Let $\mathcal{L}$ be a second-order language with special relation symbol '$<$'. Let φ be a formula. A quantifier is *bounded* iff it's of the form "$\exists x < y$" or "$\forall x < y$" for individual variables x, y.

- φ is Σ^0_0 iff all quantifiers in φ are bounded.
- φ is Σ^0_{n+1} iff φ is of the form "$\exists m \psi$" for individual variable m and ψ a Π^0_n-formula.
- φ is Π^0_n iff φ is of the form "$\neg\psi$" for ψ a Σ^0_n-formula.

- We can also define Σ^1_n-formulas.
- Note that not every formula can be placed in the hierarchy: $\exists P^1_0 \, \varphi(x, P^1_0)$ can't, for example.
- For $\mathcal{N}$, we'll write $\exists^\omega m$ or $\exists^\mathcal{N} x$ to distinguish variables.
- This gets equality right: $x_0 = x_1$ is Π^0_1 definable: $\forall^\omega n \; (x_0(n) = x_1(n))$.

Second-Order Logic

For $A \subseteq \mathcal{N}$, define

$$\mathrm{SOL}_{\Sigma^0_n}(A) = \{\varphi(\utilde{\mathcal{N}}) : \varphi \text{ is } \Sigma^0_n \text{ with parameters in } A\}.$$

This gives another hierarchy:

$$\begin{array}{ccccccccc} & & \mathrm{SOL}_{\Sigma^0_1} & & & & \mathrm{SOL}_{\Sigma^0_2} & & \\ & \rotatebox{45}{$\subsetneq$} & & \rotatebox{-45}{$\subsetneq$} & & \rotatebox{45}{$\subsetneq$} & & \rotatebox{-45}{$\subsetneq$} & \\ \mathrm{SOL}_{\Delta^0_1} & & & & \mathrm{SOL}_{\Delta^0_2} & & & & \cdots \\ & \rotatebox{-45}{$\subsetneq$} & & \rotatebox{45}{$\subsetneq$} & & \rotatebox{-45}{$\subsetneq$} & & \rotatebox{45}{$\subsetneq$} & \\ & & \mathrm{SOL}_{\Pi^0_1} & & & & \mathrm{SOL}_{\Pi^0_2} & & \end{array}$$

We can show all sorts of nice properties about these pointclasses, but really, this is just the arithmetical hierarchy:

Theorem (18 D • 8)

For each $0 < n < \omega$, $\Sigma^0_n(A) = \mathrm{SOL}_{\Sigma^0_n}(A) \cap \mathcal{P}(\mathcal{N})$.

In particular, $\utilde{\boldsymbol{\Sigma}}^0_n$-sets are those Σ^0_n-definable with parameters in $\mathcal{N}$.

Second-Order Logic

We can do the same for the analytical hierarchy.

Definition

Let $\mathcal{L}$ be a second-order language with special relation symbol '$<$'. Let φ be a formula. A quantifier is *bounded* iff it's of the form "$\exists x < y$" or "$\forall x < y$" for individual variables x, y.

- φ is Σ^1_0 iff φ is Σ^0_1.
- φ is Σ^1_{n+1} iff φ is of the form "$\exists x\, \psi$" for predicate variable x and ψ a Π^1_n-formula.
- φ is Π^1_n iff φ is of the form "$\neg\psi$" for ψ a Σ^0_n-formula.

- Note that not every formula can be placed in the hierarchy: $\exists m\, \exists P^1_0\, \varphi(x, P^1_0)$ can't, for example.
- But for $\mathcal{N}$, every formula *is equivalent* to one in the hierarchy because we can code things.
- This gives another notion of being $\Sigma^1_n(A)$-definable.
- We get another hierarchy of $\mathrm{SOL}_{\Sigma^1_n}(A)$.

Second-Order Logic

This gives another hierarchy:

$$\begin{array}{ccccccccc}
 & & \mathrm{SOL}_{\Sigma^0_1} & & & & \mathrm{SOL}_{\Sigma^0_2} & & \\
 & \rotatebox{45}{$\subsetneq$} & & \rotatebox{-45}{$\subsetneq$} & & \rotatebox{45}{$\subsetneq$} & & \rotatebox{-45}{$\subsetneq$} & \\
\mathrm{SOL}_{\Delta^0_1} & & & & \mathrm{SOL}_{\Delta^0_2} & & & & \cdots \\
 & \rotatebox{-45}{$\subsetneq$} & & \rotatebox{45}{$\subsetneq$} & & \rotatebox{-45}{$\subsetneq$} & & \rotatebox{45}{$\subsetneq$} & \\
 & & \mathrm{SOL}_{\Pi^0_1} & & & & \mathrm{SOL}_{\Pi^0_2} & &
\end{array}$$

And again, this turns out to be exactly what we want.

Theorem (18 D • 11)

For $A \subseteq \mathcal{N}$, $0 < n < \omega$, $\Sigma^1_n(A) = \mathrm{SOL}_{\Sigma^1_n}(A)$.

We also have other notions of definability that give these hierarchies.

HF and HC

- Write HF for the set of hereditarily finite sets (i.e. $H_{\aleph_0}$)
- Write HC for the set of hereditarily countable sets (i.e. $H_{\aleph_1}$)

We can pretty easily code $\langle \mathrm{HF}, \in \rangle$ into $\mathbf{N}$ in a Σ_0-definable way.

Proof.

The easiest way to conceptually see this is that any element of HF can be writen down with parantheses and commas: $\emptyset = \{\}$, and $\{\emptyset, 2\} = \{\{\}, \{\{\}, \{\{\}\}\}\}$, for example. We can then identify HF with finite strings, and look at whether something's an element according to whether it's a substring and then counting how many unpaired parantheses exist to the left and right. ⊣

We can also get that $\mathbf{N}$ is simply definable over $\mathbf{HF} = \langle \mathrm{HF}, \in \rangle$.

HF and HC

Those two ideas together tell us that something Σ_n-definable over **HF** is Σ_n-definable over **N** and vice versa just by relativizing quantifiers over the coded versions of one in the other.

Result (18 D • 16)

For $0 < n < \omega$, $X \subseteq \mathcal{N}$ is Σ^0_n iff X is Σ_n-definable over **HF**.

If we expand to second order logic, nothing goes wrong.

Result (18 D • 17)

For $0 < n < \omega$, $X \subseteq \mathcal{N}$ is Σ^0_n iff X is Σ^0_n-definable over the second-order version of **HF**.

And indeed, this generalizes to the analytical hierarchy.

Result (18 D • 18)

For $0 < n < \omega$, $X \subseteq \mathcal{N}$ is Σ^1_n iff X is Σ^1_n-definable over the second-order version of **HF**.

HF and HC

Question

So where does **HC** $= \langle \mathrm{HC}, \in \rangle$ *come into the picture?*

- Assuming some absoluteness (in particular, Π^1_1-absoluteness), we can show Σ^1_2-sets are Σ_1-definable over **HC**.
- This forms the base case of an induction where we get the following result.

Theorem (19 A • 13)

For $0 < n < \omega$, $X \subseteq \mathcal{N}$ *is* Σ^1_{n+1} *iff* X *is* Σ_n*-definable over* **HC**.

- The basic idea behind the base case is just to code transitive structures in HC in a simple way:

$$\langle M, \in \rangle \cong \langle \omega, E \rangle \text{ for some } E \subseteq \omega^2, \text{ i.e. } E \in \mathcal{N} \subseteq \mathrm{HC}.$$

- To find a real where the Π^1_1-formula φ holds, we merely need a countable, transitive model of a finite fragment of ZFC where φ holds of that real.
- Using coding, we can express the existence of such a model with that real in a Σ_1-way.