Math Logic: Model Theory & Computability
Lecture 30

We can't directly apply Cantor diagonalization to a parameterization of a class of functions on $\mathbb{N}^k$ because we need the set of codes to be the same as the set of inputs, i.e. we need $\mathbb{N} = \mathbb{N}^k$, so $k=1$. But we can encode $\mathbb{N}^k$ into $\mathbb{N}$:

Def. Let $\Gamma := \bigcup_{k=0}^{\mathbb{N}} \Gamma(\mathbb{N}^k)$, where $\Gamma(\mathbb{N}^k)$ is a class of subsets of $\mathbb{N}^k$. A *parameterization* for Γ is a subset $P \subseteq \mathbb{N} \times \mathbb{N}$ such that for each $k \in \mathbb{N}$ and $R \in \Gamma(\mathbb{N}^k)$ there is $c \in \mathbb{N}$ such that for all $\vec{a} \in \mathbb{N}^k$,

$$R(\vec{a}) \quad \text{iff} \quad P_c(\langle \vec{a} \rangle).$$

Similarly, for $\Delta := \bigcup_{k=0}^{\infty} \Delta(\mathbb{N}^k)$, where $\Delta(\mathbb{N}^k)$ is a class of functions $\mathbb{N}^k \to \mathbb{N}$, a *parameterization* for Δ is a function $F : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for each $k \in \mathbb{N}$ and $f \in \Delta(\mathbb{N}^k)$, there is $c \in \mathbb{N}$ such that for all $\vec{a} \in \mathbb{N}^k$,

$$f(\vec{a}) = F_c(\langle \vec{a} \rangle).$$

Cantor diagonalization now gives:

Cor. The class of computable (resp. primitive recursive) functions/relations does not admit a parameterization that is in the same class.

Proof. Let Γ be the class of computable subsets of powers of $\mathbb{N}$ and let $P \subseteq \mathbb{N} \times \mathbb{N}$ be a parameterization for it. Suppose P is itself computable. Then $\mathrm{AntiDiag}_P := \{ n \in \mathbb{N} : \neg P(n, \langle n \rangle) \}$ is also computable because Γ is closed under complements. Let $c \in \mathbb{N}$ be such that for all $a \in \mathbb{N}$,

$$\mathrm{AntiDiag}_P(a) \quad \text{iff} \quad P_c(\langle a \rangle).$$

But then $\mathrm{AntiDiag}_P(c)$ iff $P_c(\langle c \rangle)$ iff $P(c, \langle c \rangle)$ iff $\neg \mathrm{AntiDiag}_P(c)$. The proof for functions is similar and uses that the inverse-bit function is computable. The proofs are identical for primitive recursive. $\square$

However:

Prop. There is a computable parameterization for the class of primitive recursive functions/relations.

Proof. It is enough to prove for functions since if $F:\mathbb{N}\times\mathbb{N}\to\mathbb{N}$ is a computable parameterization for prim. rec. functions, then $\mathrm{bit}\circ F$ is the indicator function of a computable parameterization for prim. rec. relations.
To each prim. rec. $f:\mathbb{N}^k\to\mathbb{N}$ we associate a code $c_f\in\mathbb{N}$ defining c_f by induction on the definition of f:

Case 1: f is a basic function in (PR1). Then put $c_f := \langle 1, k, m\rangle$, where k is the arity of f and
- $m = 0$ if $f = S$ the successor function.
- $m = 2(i+1)$ if $f = C_i^k$ is the constant i function
- $m = 2i+1$ if $f = P_i^k$ is the projection onto the ith coordinate.

Case 2: f is obtained by composition (PR2) from $g:\mathbb{N}^\ell\to\mathbb{N}$ and $f_1,\dots,f_\ell:\mathbb{N}^k\to\mathbb{N}$.
Then put $c_f := \langle 2, k, \langle c_g, c_{f_1},\dots,c_{f_\ell}\rangle\rangle$.

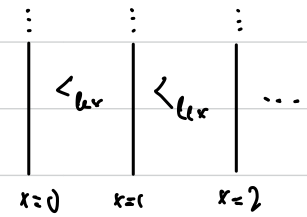
Case 3: f is obtained by prim. rec. (PR3) from $g:\mathbb{N}^k\to\mathbb{N}$ and $h:\mathbb{N}^{k+2}\to\mathbb{N}$.
Then put $c_f := \langle 3, k+1, \langle c_g, c_h\rangle\rangle$.

Define $F:\mathbb{N}\times\mathbb{N}\to\mathbb{N}$ as follows $F(c,a) := f((a)_0, (a)_1, \dots, (a)_{k-1})$ if $c = c_f$ for some prim. rec. $f:\mathbb{N}^k\to\mathbb{N}$, and $f(c,a) := 0$, otherwise.
It is clear that F is a parameterization for the class of prim. rec. functions, and proving that F is computable is done through Dedekind analysis of prim. recursion, searching for a computation-certificate for $f((a)_0, (a)_1, \dots, (a)_{k-1})$, i.e. a sequence $(\langle c_{f_1}, a_1, b_1\rangle, \dots, \langle c_{f_n}, a_n, b_n\rangle)$, where $f_1,\dots,f_n$ are the functions that

appear in the def. of f, a_i is the code of the input for f_i and b_i is the value of f_i on this input. We leave the details as a (difficult) exercise. $\boxtimes$

The computable parameterization for the class of prim. rec. functions is an example of a computable but not primitive rec. function. Another, more natural example is:

Ackermann function. Let $<_{lex}$ be the lexicographic ordering of $\mathbb{N}^2$, i.e.



$$(x_1, y_1) <_{lex} (x_2, y_2) \quad :\Longleftrightarrow \quad x_1 < x_2 \text{ or } (x_1 = x_2 \text{ and } y_1 < y_2).$$

This is a well-ordering so we can define a function $A : \mathbb{N}^2 \to \mathbb{N}$ by induction on $(\mathbb{N}^2, <_{lex})$ as follows:

$$\begin{cases} A(0, y) = y+1 \\ A(x+1, 0) = A(x, 1) \\ A(x+1, y+1) = A(x, A(x+1, y)) \end{cases}$$

Prop. For each prim. rec. $f : \mathbb{N} \to \mathbb{N}$, there is $x_f \in \mathbb{N}$ such that for all $y \in \mathbb{N}$
$$A(x_f, y) > f(y).$$

Proof. By induction on the construction of f, using inequalities for A. $\boxtimes$

Cor. A is not prim. rec.

Proof. Otherwise $A' : \mathbb{N} \to \mathbb{N}$ by $n \mapsto A(n,n)$ is prim. rec. so $\exists\, x_0 \in \mathbb{N}$ with $A(x_0, y) > A'(y)$ for all $y \in \mathbb{N}$ contradicting $A(x_0, x_0) = A'(x_0)$.
$\square$

Arithmetical hierarchy and undecidability.

In this last section, we answer the following questions:

(Q1) Computable sets / functions are arithmetical, is the converse true? If not, how much more complicated arithmetical sets can get? Can we stratify the class of arithmetical sets into a hierarchy of classes of simpler sets, starting with computable sets?

(Q2) Okay, PA is incomplete, and even ZFC is incomplete (follows from a generalization of Gödel's incompleteness theorem), but maybe the set of theorems of these theories is computer-recognizable? More precisely, given a computable theory T that's rich enough (like PA or ZFC), is the set

$$\ulcorner \mathrm{Thm}(T) \urcorner := \{ \ulcorner \varphi \urcorner : \varphi \in \mathrm{Thm}(T) \}$$
$$= \{ \ulcorner \varphi \urcorner : \varphi \text{ is provable from } T \}$$

computable?

It turns out that the answers to (Q1) and (Q2) are Yes and No, respectively. We discuss the answers without going much into proofs.

Σ^0_1 sets and arithmetical hierarchy. We saw that every computable set is of the form $\exists y\, R(\vec{x}, y)$ where R is primitive recursive. Is any set of this form computable? Let's first define this class of sets.

Notation. Let Γ be a class of subsets of finite powers of $\mathbb{N}$. Denote

$$\exists^{\mathbb{N}} \Gamma := \{ \exists_y S(\vec{x}, y) : S \in \Gamma \}, \quad \forall^{\mathbb{N}} \Gamma := \{ \forall_y S(\vec{x}, y) : S \in \Gamma \},$$
$$\neg \Gamma := \{ \neg S(\vec{x}) : S \in \Gamma \}.$$

Let $\mathcal{A}$, $\mathcal{C}$, $\mathcal{R}$ denote the classes of arithmetical, computable and prim. rec. sets, resp.

Definition. Let $\Sigma^0_1 := \exists^{\mathbb{N}}\mathcal{C}$ and supposing that Σ^0_n is defined, we define

$$\Pi^0_n := \neg\Sigma^0_n \qquad \Delta^0_n := \Sigma^0_n \cap \Pi^0_n$$
$$\Sigma^0_{n+1} := \exists^{\mathbb{N}}\Pi^0_n.$$

Thus, $\Pi^0_1 = \forall^{\mathbb{N}}\mathcal{C}$, $\Sigma^0_2 = \exists^{\mathbb{N}}\forall^{\mathbb{N}}\mathcal{C}$, $\Pi^0_2 = \forall^{\mathbb{N}}\exists^{\mathbb{N}}\mathcal{C}$, $\Sigma^0_3 = \exists^{\mathbb{N}}\forall^{\mathbb{N}}\exists^{\mathbb{N}}\mathcal{C}$, ...

$$\Sigma^0_n = \underbrace{\exists^{\mathbb{N}}\forall^{\mathbb{N}}\exists^{\mathbb{N}}\ldots}_{n}\mathcal{C},$$

$$\Pi^0_n = \underbrace{\forall^{\mathbb{N}}\exists^{\mathbb{N}}\forall^{\mathbb{N}}\ldots}_{n}\mathcal{C}.$$

Prop. (a)

$$\begin{array}{ccccccccc}
 & \subseteq\ \Sigma^0_1\ \subseteq & & \subsetneq\ \Sigma^0_2\ \subseteq & & & \subsetneq\ \Sigma^n_0\ \subseteq & & \\
\mathcal{C} \subseteq \Delta^0_1 & & \Delta^0_2 & & \cdots & \Delta^0_n & & \Delta^0_{n+1} & \cdots \\
 & \subseteq\ \Pi^0_1\ \subsetneq & & \subseteq\ \Pi^0_2\ \subsetneq & & & \subseteq\ \Pi^0_n\ \subsetneq & &
\end{array}$$

(b)
$$\mathcal{A} = \bigcup_{n=1}^{\infty}\Sigma^0_n = \bigcup_{n=1}^{\infty}\Pi^0_n = \bigcup_{n=1}^{\infty}\Delta^0_n.$$

Proof. (a) is straightforward and (b) follows from the fact that every extended σ_{arithm}-formula $\varphi(\vec{x})$ is equivalent (in all σ_{arithm}-structures) to an extended σ_{arithm}-formula of the form $\underbrace{\exists y_1 \forall y_2 \exists y_3 \ldots}_{\text{some } n} \psi(\vec{x}, y_1, y_2, \ldots, y_n)$, where ψ is quantifier free, and hence defines a prim. rec. set. $\square$

Prop. $\Sigma^0_1 := \exists^{\mathbb{N}}\mathcal{C} = \exists^{\mathbb{N}}\mathcal{R}$. Thus, in the above definition, $\mathcal{C}$ can be replaced with $\mathcal{R}$.

Proof. Because $\mathcal{R} \subseteq \mathcal{C}$, we have $\exists^{\mathbb{N}}\mathcal{R} \subseteq \exists^{\mathbb{N}}\mathcal{C}$, so it remains to show the converse. By the normal form theorem, $\mathcal{C} \subseteq \exists^{\mathbb{N}}\mathcal{R}$, so $\exists^{\mathbb{N}}\mathcal{C} \subseteq \exists^{\mathbb{N}}\exists^{\mathbb{N}}\mathcal{R}$. But we can combine two $\exists^{\mathbb{N}}$ into one: $\exists y\, \exists x\, R(\vec{a}, x, y) \iff \exists z\, R(\vec{a}, (z)_0, (z)_1)$, so $\exists^{\mathbb{N}}\mathcal{C} \subseteq \exists^{\mathbb{N}}\exists^{\mathbb{N}}\mathcal{R} = \exists^{\mathbb{N}}\mathcal{R}$. $\square$

Theorem. Let Γ be Σ^0_n or Π^0_n for some $n \geq 1$. Then Γ admits a universal set, i.e. a parameterization $\mathcal{U} \subseteq \mathbb{N} \times \mathbb{N}$ of Γ with $\mathcal{U} \in \Gamma$, i.e. for each $k \geq 1$ and $S \in \Gamma(\mathbb{N}^k)$ there is $c \in \mathbb{N}$ such that for all $\vec{a} \in \mathbb{N}^k$

$$S(\vec{a}) \iff \mathcal{U}_c(\langle \vec{a} \rangle).$$

Proof. If $\mathcal{U}$ is a universal set for Σ^0_n, then $\mathcal{U}^c$ is a universal set for $\neg\Sigma^0_n = \Pi^0_n$. Similarly, if $\mathcal{U}$ is a universal set for Π^0_n, then

$$\mathcal{U}'(c,x) :\iff \exists y\, \mathcal{U}(c, \underbrace{x * \langle y \rangle}_{\text{"}\langle \vec{a}, y\rangle \text{ where } \langle \vec{a}\rangle = x})$$

is a universal set for $\exists^{\mathbb{N}} \Pi^0_n = \Sigma^0_{n+1}$.

For Σ^0_1, let P be a computable parameterization of prim. rec. sets and put $\mathcal{U}(c,x) :\iff \exists y\, P(c, x * \langle y \rangle)$. $\square$

Cor. (a) The arithmetical hierarchy is strict, i.e. $\Delta^0_n \subsetneq \Sigma^0_n, \Pi^0_n \subsetneq \Delta^0_{n+1}$ for all $n \geq 1$. In particular, $\Sigma^0_n \neq \Pi^0_n$.

(b) Δ^0_n does not admit a universal set.

Proof. (a) follows from (b), and (b) follows by Cantor diagonalization because Δ^0_n is closed under complement. $\square$

Kleene's Theorem. $\mathcal{C} = \Delta^0_1$, i.e. if a set $S \subseteq \mathbb{N}^k$ and its complement S^c are both Σ^0_1, then S is computable.

Proof. Search for a witness for S and S^c at the same time, and you're guaranteed to find it. More precisely, let $S(\vec{x}) :\iff \exists w\, A(\vec{x}, w)$, $S^c(\vec{x}) :\iff \exists w\, B(\vec{x}, w)$, where $A, B \subseteq \mathbb{N}^{k+1}$ are computable.

Then $S(\vec{x}) \iff A(\vec{x}, \mu_w(A(\vec{x}, w) \vee B(\vec{x}, w)))$ and the search μ_w is safe so S is computable. $\square$

Thus, we have proven the following picture:

Theorem.

$$\underbrace{\begin{array}{ccccccccccccc}
 & & \Sigma_1^0 & & & & \Sigma_2^0 & & & & \Sigma_n^0 & & \\
\mathcal{C} = \Delta_1^0 & \begin{matrix}\subsetneq \\ \subsetneq\end{matrix} & & \begin{matrix}\subsetneq \\ \subsetneq\end{matrix} & \Delta_2^0 & \begin{matrix}\subsetneq \\ \subsetneq\end{matrix} & & \begin{matrix}\subsetneq \\ \subsetneq\end{matrix} & \cdots \quad \Delta_n^0 & \begin{matrix}\subsetneq \\ \subsetneq\end{matrix} & & \begin{matrix}\subsetneq \\ \subsetneq\end{matrix} & \Delta_{n+1}^0 \subsetneq \cdots \\
 & & \Pi_1^0 & & & & \Pi_2^0 & & & & \Pi_n^0 & &
\end{array}}_{\text{Arithmetical sets}}$$

Undecidable theories.

Prop. Let σ be a finite signature and T be a σ-theory. If T is computable (i.e. the set $\ulcorner T \urcorner \subseteq \mathbb{N}$ of codes is computable) then $\text{Proof}_T \subseteq \mathbb{N}^2$ is computable, where

$$\begin{aligned} \text{Proof}(x,y) \;:\Leftrightarrow\;& x \text{ is a code of a proof from } T \text{ of the formula encoded by } y \\ \Leftrightarrow\;& x = \langle \ulcorner \varphi_1 \urcorner, \dots, \ulcorner \varphi_n \urcorner \rangle \text{ where } \ulcorner \varphi_n \urcorner = y \text{ and } (\varphi_1, \dots, \varphi_n) \\ & \text{is a proof of } \varphi_n \text{ from } T. \end{aligned}$$

In particular, $\text{Thm}_\sigma(T)$ is Σ_1^0 (i.e. $\ulcorner \text{Thm}_\sigma(T) \urcorner \subseteq \mathbb{N}$ is Σ_1^0).

Proof. That $\text{Proof}(x,y)$ is computable, we've already shown, and

$$y \in \ulcorner \text{Thm}_\sigma(T) \urcorner \Leftrightarrow \text{Provable}_T(y) :\Leftrightarrow \exists x \, \text{Proof}(x,y). \qquad \square$$

Def. Let σ be a finite signature. Call a σ-theory T **decidable** if $\text{Thm}_\sigma(T)$ is computable.

Cor (from Kleene's theorem). If a computable σ-theory is complete, then it is decidable.

Proof. We only need to show that $\mathrm{Provable}_T \subseteq \mathbb{N}$ is also Π^0_1, but it is because for each sentence φ, either $\mathrm{Provable}_T(\ulcorner\varphi\urcorner)$ or $\mathrm{Provable}_T(\ulcorner\neg\varphi\urcorner)$, hence

$$\mathrm{Provable}_T(y) \iff \neg\,\mathrm{Provable}(\mathrm{code}(\neg) * y). \qquad \square$$

Therefore, the following theorem of Church vastly strengthens and generalizes Gödel's incompleteness theorem:

Church's Undecidability Theorem. Let σ be a finite signature. Every σ-theory T that computably interprets PA is undecidable. E.g. PA, ZFC are undecidable.

Gödel's Incompleteness Theorem (Rosser's form). Every computable σ-theory that computably interprets PA is incomplete. E.g. PA, $PA \cup \{\neg\sigma_{PA}\}$, ZFC are incomplete.

Def. Let σ_1, σ_2 be finite signatures and T_i be a σ_i-theory, $i=1,2$.
A function $\pi : \mathrm{Formulas}(\sigma_1) \to \mathrm{Formulas}(\sigma_2)$ is called an interpretation of T_1 in T_2 if for all $\varphi, \psi \in \mathrm{Sentences}(\sigma_1)$,

(i) $T_1 \vDash \psi \implies T_2 \vDash \psi$

(ii) $T_2 \vDash \pi(\neg\varphi) \leftrightarrow \neg\pi(\varphi)$

(iii) $T_2 \vDash \pi(\varphi \wedge \psi) \leftrightarrow (\pi(\varphi) \wedge \pi(\psi))$.

We say that T_2 computably interprets T_1 if there is a computable interpretation $\pi : \mathrm{Formulas}(\sigma_1) \to \mathrm{Formulas}(\sigma_2)$, i.e. the function

$$\hat{\pi} : \mathbb{N} \to \mathbb{N}$$
$$n \mapsto \begin{cases} \ulcorner\pi(\varphi)\urcorner & \text{if } n = \ulcorner\varphi\urcorner \text{ for some } \varphi \in \mathrm{Formulas}(\sigma_1) \\ 0 & \text{o.w.} \end{cases}$$

is computable.

Example. ZFC computably interprets PA.