A Quick Introduction to MATHEMATICAL LOGIC

SAEED SALEHI

Frontiers Summer School in Mathematics

Theory of Computation, 30 August 2021

Computably Decidable (& Enumerable) Sets

Definition (Computably Decidable Set)

Set A is computably decidable where there is an algorithm $\mathcal{P}$ decides on any input x whether $x \in A$ (outputs **YES**) or $x \notin A$ (outputs **NO**). $\diamond$

$$\xrightarrow[\qquad\qquad]{\texttt{input}:\quad x\in\mathbb{M}} \boxed{\text{Algorithm}} \xrightarrow{\texttt{output}:} \begin{cases} \textbf{YES} & \text{if } x \in A \\ \textbf{NO} & \text{if } x \notin A \end{cases}$$

Algorithm: single–input, Boolean–output (**1**, **0**).

Definition (Computably Enumerable Set)

Set A is computably enumerable where there is an (input-free) algorithm $\mathcal{P}$ lists all members of A; i.e., $A = \texttt{output}(\mathcal{P})$. $\diamond$

$$\boxed{\text{Algorithm}} \xrightarrow{\texttt{output}:} \{a_0, a_1, a_2, \cdots\} = A$$

Algorithm: input–free, outputs a set.

Syllogism & Propositional, Equality, and Predicate Logics

- Propositional Logic is decidable — Truth-Tables.
- Aristotle's Syllogism is decidable — Venn Diagrams.
- Equational Logic is enumerable — rules.
- Predicate Logic is enumerable — axioms & rule.

Automated Theorem Proving

Decidable vs. Enumerable

Theorem (Decidable $\Rightarrow$ Enumerable)

Every decidable set is enumerable.

Proof.

If $\mathcal{P}[x] = \texttt{YES}$ for $x \in A$ and $\mathcal{P}[x] = \texttt{NO}$ for $x \in A^{\complement}$, then let $n := 0$; run $\mathcal{P}[n]$; if $\texttt{YES}$ then PRINT "n"; let $n := n+1$ and repeat. ■

Theorem (Decidable $\equiv$ Enumerable & Enumerable$^{\complement}$)

A set is decidable iff it and its complement are enumerable.

Proof.

If $\texttt{output}(\mathcal{P}) = A$ and $\texttt{output}(\mathcal{P}') = A^{\complement}$, then on input x, let $n := 1$; run n steps of $\mathcal{P}, \mathcal{P}'$; if $x \in \texttt{output}(\mathcal{P})$ then PRINT "$\texttt{YES}$" & STOP, and if $x \in \texttt{output}(\mathcal{P}')$ then PRINT "$\texttt{NO}$" & STOP; if not stopped yet, let $n := n+1$ and repeat. ■

Computability Theory

Theorem (Church & Turing 1936)

Predicate Logic is *not* decidable.

$$\xrightarrow{\texttt{input}:\quad \varphi} \boxed{\text{Algorithm}} \xrightarrow{\texttt{output}:} \begin{cases} \texttt{YES} & \text{if } \vdash \varphi \\ \texttt{NO} & \text{if } \nvdash \varphi \end{cases}$$

Theorem

Equational Logic is *not* decidable.

$$\xrightarrow{\texttt{input}:\quad t,u,\Sigma} \boxed{\text{Algorithm}} \xrightarrow{\texttt{output}:} \begin{cases} \texttt{YES} & \text{if } \Sigma \vdash t \approx u \\ \texttt{NO} & \text{if } \Sigma \nvdash t \approx u \end{cases}$$

Modern Computers

▶ A Good Outcome: Introducing Turing Machines
the grand grandfather of today's modern computers.

Every computable object can be "coded" by a natural number,
since Each ASCII code can be written by a string of 0's and 1's:

0	1	2	3	4	5	6	7	8	9	10	$\cdots$
λ	0	1	00	01	10	11	000	001	010	011	$\cdots$

$$(b_i \in \{0,1\}) \; b_1 b_2 \cdots b_n \mapsto (1 b_1 b_2 \cdots b_n)_2 - 1 \; (\in \mathbb{N})$$

$$(m \in \mathbb{N}) \; m \mapsto ([\![1^{-1}]\!])\mathbf{bin}(m+1)$$

https://www.ascii-code.com/

Coding

- It is customary to consider computable functions in the form $\mathbb{N}^n \to \mathbb{N}$.

• Finite Sequences of Natural Numbers can be coded in $\mathbb{N}$:
Let $\mathfrak{p}_0, \mathfrak{p}_1, \mathfrak{p}_2, \cdots$ be the sequence of all the prime numbers $(2, 3, 5, \cdots)$, and put

$$\langle m_0, \cdots, m_k \rangle \mapsto \mathfrak{p}_0^{m_0+1} \times \cdots \times \mathfrak{p}_k^{m_k+1}.$$

This coding is injective but not surjective.

- So, one can put all the programs (and all the finite sequences of ascii codes) in a bijective correspondence with $\mathbb{N}$.

Uncomputability

- Thus, one can also put all the computably decidable (and enumerable) subsets of $\mathbb{N}$ in a bijective correspondence with $\mathbb{N}$.
- But we saw that $\mathscr{P}(\mathbb{N}) \ncong \mathbb{N}$.

  So, there *exist some* subsets of $\mathbb{N}$ that are not computably decidable, or computably enumerable!
- We will see explicit subsets of $\mathbb{N}$ that are not computably decidable or computably enumerable.

Recursion Theory (functions: $\mathbb{N}^n \to \mathbb{N}$)

Recursive Functions Contain:

- Zero Constant Function $Z(x) = 0$
- Successor Function $S(x) = x + 1$
- Projection Functions $\pi_i^k(n_1, \cdots, n_k) = n_i$
- Addition Function $A(x, y) = x + y$
- Multiplication Function $M(x, y) = x \cdot y$
- Exponentiation Function $E(x, y) = x^y$
- Prime Numbering Function $\mathfrak{p}(x) = x^{\text{th}}$ **`prime number`**
- Order Recognition Function $\chi_{\leqslant}(x, y) = \begin{cases} 1 & \text{if } x \leqslant y \\ 0 & \text{if } x > y \end{cases}$

and Are Closed Under:

- The Composition of Functions
- Minimization of Functions

$$\big[\mu z.\, f(\mathbf{x}, z) = g(\mathbf{x}, z)\big](\mathbf{x}) \mathbf{=} \mathbf{min}\{z \in \mathbb{N} \mid f(\mathbf{x}, z) = g(\mathbf{x}, z)\}$$

Church (& Turing)'s Thesis

Thesis (by Experience)

Every (Intuitionally) Computable Function is Recursive.

Note that every recursive function is clearly computable.

For example, it can be shown that recursive functions are closed under primitive recursion:

$$\mathrm{PR}^{f,g}(\mathbf{x},z)\colon \begin{cases} \mathrm{PR}^{f,g}(\mathbf{x},0) = f(\mathbf{x}) \\ \mathrm{PR}^{f,g}(\mathbf{x},z+1) = g(\mathbf{x},z,\mathrm{PR}^{f,g}(\mathbf{x},z)) \end{cases}$$

There are lots of functions $\mathbb{N}^n \to \mathbb{N}$ that are not recursive (computable). (the characteristic functions of undecidable sets)