

Prerequisites from Logic and Analysis

We begin with logic.

This book does not require any prior knowledge of formal logic.

First we expand upon some points made in the introduction to the book. All of our notions of computability (for real numbers, continuous functions, and beyond) are based on the notion of a recursive function $a: \mathbb{N} \rightarrow \mathbb{N}$ or $a: \mathbb{N}^q \rightarrow \mathbb{N}$. (\mathbb{N} denotes the set of non-negative integers.) On the other hand, this book does not require a detailed knowledge of recursion theory. For reasons to be explained below, an intuitive understanding of that theory will suffice.

We continue for now to consider only functions from \mathbb{N} to \mathbb{N} . Intuitively, a recursive function is simply a “computable” function. More precisely, a recursive function is a function which is computable by a Turing machine. The weight of fifty years experience leads to the conclusion that “recursive function” is the correct definition of the intuitive notion of a computable function. The definition is as solid as the definition of a group or a vector space. By now, the theory of recursive functions is highly developed.

However, as we have said, this book does not require a detailed knowledge of that theory. We avoid the need for heavy technical prerequisites in two ways.

1. Whenever we prove that some process is computable, we actually give the algorithm which produces the computation. As we shall see, some of these algorithms are quite intricate. But each of them is built up from scratch, so that the book is self-contained.
2. To prove that certain processes are not computable, we shall find that it suffices to know one basic result from recursive function theory—the existence of a recursively enumerable nonrecursive set. This we now discuss.

Imagine a computer which has been programmed to produce the values of a function a from nonnegative integers to nonnegative integers. We set the program in motion, and have the computer list the values $a(0), a(1), a(2), \dots$ in order. This set A of values, a subset of the natural numbers, is an example of a “recursively enumerable set”. If we take a general all purpose computer—e.g. a Turing machine—and consider the class of all such programs, we obtain the class of all recursively enumerable sets.

Suppose now that we have a recursively enumerable set A . Can we find an effective procedure which, for arbitrary n , determines whether or not $n \in A$? If $n \in A$, then

clearly we have a procedure which tells us so. Namely, we simply list the values $a(0)$, $a(1)$, $a(2)$, \dots , and if $n \in A$, then the value n will eventually appear. The difficulty comes if $n \notin A$. For then it seems possible that we might have no method of ascertaining this fact. We can, of course, list the set A to an arbitrarily large finite number of its elements, and we can observe that the value n has not occurred *yet*. However, we might have no way of determining whether n will show up at some later stage.

It turns out that, in general, there is no effective procedure which lists the elements $n \notin A$. Thus, there is no effective procedure which answers, for every n , the question: is $n \in A$?

On the other hand, for some sets A , the question of membership in A can be effectively answered. This is true for most of the sets of natural numbers commonly encountered in number theory, e.g. the set of primes. Such sets are called “recursive”. By contrast, those sets A for which we can ascertain when $n \in A$ but not when $n \notin A$, are called “recursively enumerable nonrecursive”. It is a fundamental result of logic that such sets exist (Proposition A below).

We now spell out these definitions and results in a formal manner. These are the only recursion-theoretic facts which are used in this book.

A set $A \subseteq \mathbb{N}$ is called *recursively enumerable* if $A = \emptyset$ or A is the range of a recursive function a . When A is infinite, the function a can be chosen to be one-to-one.

A set $A \subseteq \mathbb{N}$ is called *recursive* if both A and its complement $\mathbb{N} - A$ are recursively enumerable.

A cornerstone of recursion theory is the following result. For a proof see e.g. Kleene [1952], Rogers [1967], Davis [1958], Cutland [1980], Soare [1987].

Proposition A. *There exists a set $A \subseteq \mathbb{N}$ which is recursively enumerable but not recursive.*

Occasionally we will have to use a slightly stronger result—the existence of a recursively inseparable pair of sets.

Proposition B. *There exists a recursively inseparable pair of sets, i.e. a pair of subsets A, B of \mathbb{N} such that:*

- a) *A and B are recursively enumerable.*
- b) *$A \cap B = \emptyset$.*
- c) *There is no recursive set C with $A \subseteq C$ and $B \subseteq \mathbb{N} - C$.*

We need one further result from logic—the existence of a recursive pairing function J from $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, together with recursive inverse functions $K: \mathbb{N} \rightarrow \mathbb{N}$ and $L: \mathbb{N} \rightarrow \mathbb{N}$ such that:

$$J[K(n), L(n)] = n.$$

In fact, a standard form for J is:

$$J(x, y) = \frac{(x + y)(x + y + 1)}{2} + x$$

A technical comment. When we use the *characteristic function* of a set, we follow the custom of analysts rather than that of logicians. Thus we define the characteristic function χ_S of a set S by

$$\chi_S(x) = \begin{cases} 1 & \text{for } x \in S, \\ 0 & \text{for } x \notin S. \end{cases}$$

Now we turn to analysis. We assume the contents of a standard undergraduate course in real variables, together with the rudiments of measure theory. We also need the following well known definitions.

A *Banach space* is a real or complex vector space with a norm $\| \cdot \|$ such that:

$$\begin{aligned} \|x + y\| &\leq \|x\| + \|y\|, \\ \|\alpha x\| &= |\alpha| \cdot \|x\| \quad \text{for all scalars } \alpha, \\ \|x\| &\geq 0 \quad \text{with equality if and only if } x = 0, \end{aligned}$$

and such that the space is complete in the metric $\|x - y\|$.

A *Hilbert space* is a Banach space in which the norm is given by an inner product (x, y) : the form (x, y) is linear in the first variable and conjugate linear in the second variable, and it is related to the norm by $\|x\| = (x, x)^{1/2}$.

The following spaces are used frequently in this book. Except for $C^\infty[a, b]$, they are all Banach spaces.

$L^p[a, b]$: for $1 \leq p < \infty$, the space of all measurable functions f on $[a, b]$ for which the L^p -norm $\|f\|_p = \left(\int_a^b |f(x)|^p dx \right)^{1/p}$ is finite.

l^p : for $1 \leq p < \infty$, the space of all real or complex sequences $\{c_n\}$ for which the l^p -norm $\|\{c_n\}\|_p = \left(\sum_{n=0}^{\infty} |c_n|^p \right)^{1/p}$ is finite.

$C[a, b]$: the space of all continuous functions f on $[a, b]$, endowed with the uniform norm $\|f\|_\infty = \sup_x \{|f(x)|\}$.

$C^n[a, b]$: the space of all n times continuously differentiable functions f on $[a, b]$; here the norm can be taken as the sum of the uniform norms of $f^{(k)}$, $0 \leq k \leq n$.

$C^\infty[a, b]$: the space of infinitely differentiable functions on $[a, b]$.

By obvious modifications, the interval $[a, b]$ can be replaced by the real line \mathbb{R} or euclidean q -space \mathbb{R}^q . For the space $C(\cdot)$, we use $C_0(\mathbb{R}^q)$, the space of continuous functions which vanish at infinity.

Finally, in the important case where $p = 2$, L^p and l^p are Hilbert spaces.