Qeios

How (not) to Compute the Halting Probability

Manizheh Jalilvand
Department of Computer and Data Sciences, Shahid Beheshti University,
P.O.Box 19839-69411, Tehran, IRAN. `m.jalilvand.sarand@gmail.com`
Behzad Nikzad
Research Center of Biosciences & Biotechnology (RCBB), University of Tabriz,
P.O.Box 51666-16471, Tabriz, IRAN. `blsnikzad@yahoo.com`
Saeed Salehi
Research Center of Biosciences & Biotechnology (RCBB), University of Tabriz,
P.O.Box 51666-16471, Tabriz, IRAN. `root@SaeedSalehi.ir`

Abstract

We study Chaitin's well-known constant Omega, and show that this number is not a probability of halting the randomly chosen input-free programs under any infinite discrete measure. We suggest some methods for defining the halting probabilities by various measures.

Keywords: Chaitin's Constant · The Omega Number · Halting Probability · Probability Measure · Kolmogorov Axioms · Random Numbers · Algorithmic Information Theory.
2020 Math Subject Classification: 60A10 · 28A05 · 68Q04 · 03D10.

"Mathematics is the science of learning how *not* to compute".
—*Heinrich Maschke* (1853–1908); see [10, p. 667].

1 Introduction

There are very many papers and some books on the so-called *Halting Probability* Ω, also known as Chaitin's Constant or Chaitin's Number. We aim to see if Ω defines a probability for the halting problem. And if so, on what measure? What is the distribution of that probability? What is the sample space? We will give a systematic understanding, with a very brief history, of this number and will suggest some measures based on which a halting probability can be defined, with all the glory of mathematical rigor. Let us observe right away that a real number cannot be called a probability if it is just between 0 and 1; there should be a measure and a space for a probability that satisfies Kolmogorov axioms (see, e.g., [12]): that $\mu(\mathbb{S})=1$ and $\mu(\bigcup_i S_i)=\sum_i \mu(S_i)$, where $\mathbb{S}$ is the sample space, $\{S_i\}$ is an arbitrary indexed family of pairwise disjoint subsets of $\mathbb{S}$, and the *partial* function $\mu\colon \mathscr{P}(\mathbb{S}) \to [0,1]$ is the probability measure (defined on the so-called *measurable* subsets of $\mathbb{S}$).

The number Ω was introduced by Chaitin [5, p. 337] in 1975, when it was denoted by ω. The symbol Ω appears in Chaitin's second *Scientific American* paper [6], where it was defined as the probability that "a completely random program will halt" (p. 80). This is sometimes called “the secret number,” “the magic number,” “the number of wisdom,” etc. [14, p. 178]; it is also claimed “to hold the mysteries of the universe” [9]. It was stated in [2] that “The first example of a random real was Chaitin’s Ω” (p. 1411); but as Barmpalias put it in [1, p. 180, fn. 9], “Before Chaitin’s discovery, the most concrete Martin-Löf random real known was a 2-quantifier definable number exhibited [by] Zvonkin and Levin” (in 1970).

Let us look at the formal definition of Ω rigorously.

Definition 1.1 (binary code, length, `ASCII` code):

Let $\boldsymbol{\Sigma}=\{\mathtt{0},\mathtt{1}\}^+$ be the set of all the (nonempty) finite strings of the symbols (binary bits) `0` and `1`. For a string $\sigma\in\boldsymbol{\Sigma}$, let $|\sigma|$ denote its length. Every program has a unique `ASCII` code* which is a binary string. This is called the binary code of the program. ✧

Example 1.2 (binary code, length, `ASCII` code):

The object `01001` is a binary string, and its length is $|\mathtt{01001}|=5$. The `ASCII` code of the symbol @ is `01000000`, and `00100000` is the `ASCII` code of the *blank space*, produced by the `space bar` on the keyboard. ✧

Remark 1.3 (the empty string):

We exclude the empty string with length 0, which is usually included in automata theory and formal languages. So, our strings have all positive lengths. ✧

Example 1.4 (binary code of a program):

Let us consider the command

```
BEEP
```

in, e.g., the `BASIC` programming language; it produces the actual “beep” sound through the sound card of the computer hardware. The binary code of this command is the concatenation of the `ASCII` codes of the capital letters `B` (which is `01000010`), `E` (which is `01000101`), `E` (the same), and `P` (which is `01010000`), building together the following finite binary string:

`01000010010001010100010101010000`. ✧

For defining Ω, Chaitin gave the main idea as, “The idea is you generate each bit of a program by tossing a coin and ask what is the probability that it halts.” [7, p. 151]. By “program,” Chaitin meant an *input-free* program, and by “bit” of a program, he meant any of the `0`’s and `1`’s in its binary (`ASCII`) code.

Example 1.5 (Programs: Input-Free, Halting, and Non-halting):

Consider the following three programs over a fixed programming language, where the variables i and n range over the natural numbers.

Program 1	Program 2	Program 3
`BEGIN`	`BEGIN`	`BEGIN`
`LET` $n:=1$	`LET` $n:=1$	`INPUT` i
`WHILE` $n>0$ `DO`	`WHILE` $n<9$ `DO`	`WHILE` $i<9$ `DO`
`begin`	`begin`	`begin`
`PRINT` n	`PRINT` n	`PRINT` i
`LET` $n:=n+1$	`LET` $n:=n+1$	`LET` $n:=i+1$
`end`	`end`	`end`
`END`	`END`	`END`

* `ASCII`: *American Standard Code for Information Interchange*, The Extended 8-bit Table Based on Windows-1252 (1986), available at `https://www.ascii-code.com/`

Program 1 and Program 2 do not take any input, while Program 3 takes some input (from the user) and then starts running. Program 1 never halts (loops forever) after it starts running, but Program 2 halts eventually (when n reaches 9). Program 3 takes the input i and halts on some values of i (when $i \geqslant 9$) and loops forever on others (when $i < 9$). ✧

As Chaitin took "programs" for "input-free programs," we will also use these terms interchangeably; so, we disregard the programs that take some inputs and consider only input-free programs. The number Ω was defined by Chaitin as follows:

> What exactly is the halting probability? I've written down an expression for it: $\Omega = \sum_{p \text{ halts}} 2^{-|p|}$. [..] If you generate a computer program at random by tossing a coin for each bit of the program, what is the chance that the program will halt? You're thinking of programs as bit strings, and you generate each bit by an independent toss of a fair coin [7, p. 150].

Actually, for an arbitrary set S of binary strings, one can define Ω_S as follows:

Definition 1.6 (Ω_S):
For a set of binary strings $S \subseteq \mathbf{\Sigma}$, let $\Omega_S = \sum_{\sigma \in S} 2^{-|\sigma|}$; see Definition 1.1. ✧

Example 1.7 (Ω_S):
We have $\Omega_{\{0\}} = \frac{1}{2}$, $\Omega_{\{0,00\}} = \frac{3}{4} = \Omega_{\{1,00\}}$, and $\Omega_{\{0,1,00\}} = \frac{5}{4}$. We also have $\Omega_{\mathscr{C}} = 1$, where $\mathscr{C} = \{1, 00, 010, 0110, 01110, 011110, \dots\}$. ✧

Definition 1.8 ($\mathbb{P}, \mathbb{H}$):
Let $\mathbb{P}$ denote the set of the binary codes of all the input-free programs over a fixed programming language. Over that fixed language, let $\mathbb{H}$ denote the set of the binary codes of all those input-free programs that halt after running (eventually stop; do not loop forever). ✧

Definition 1.9 (The Omega Number):
Let $\mathbf{\Omega}$ be the number $\Omega_{\mathbb{H}}$ (see Definitions 1.6 and 1.8). ✧

This finishes our mathematical definition of the Omega Number. Let us notice that "the precise numerical value of [$\mathbf{\Omega}$] depends on the choice [of the fixed] programming language" [4, p. 236].

2 Measuring Up the Omega Number

The number $\mathbf{\Omega}$ as defined in Definition 1.9 may not lie in the interval $[0,1]$, and so it may not be the probability of anything. As Chaitin warned,

> there's a technical detail which is very important and didn't work in the early version of algorithmic information theory. You couldn't write this: $\Omega = \sum_{p \text{ halts}} 2^{-|p|}$. It would give infinity. The technical detail is that no extension of a valid program is a valid program. Then this sum $\Omega = \sum_{p \text{ halts}} 2^{-|p|}$ turns out to be between zero and one. Otherwise it turns out to be infinity. It only took ten years until I got it right. The original 1960s version of algorithmic information theory is wrong. One of the reasons it's wrong is that you can't

> even define this number. In 1974 I redid algorithmic information theory with 'self-delimiting' programs and then I discovered the halting probability, Ω. [7, p. 150]

Definition 2.1 (Prefix-Free):
A set of binary strings is prefix-free when none of its elements is a proper prefix of another element. ✧

Example 2.2 (Prefix-Free):
The sets $\{\mathtt{0}\}$ and $E=\{\mathtt{1},\mathtt{00}\}$ are both prefix-free, but their union $\{\mathtt{0},\mathtt{1},\mathtt{00}\}$ is not, since $\mathtt{0}$ is a prefix of $\mathtt{00}$; neither is the set $\{\mathtt{0},\mathtt{00}\}$. The set $\mathscr{C}=\{\mathtt{1},\mathtt{00},\mathtt{010},\mathtt{0110},\mathtt{01110},\mathtt{011110},\ldots\}$ is also prefix-free (see Example 1.7). ✧

The Omega of every prefix-free set is non-greater than one. This is known as Kraft's Inequality [11] and will be proved in the following (Proposition 2.9).

Definition 2.3 (binary expansion in base 2):
Every natural number has a binary expansion (in base 2), which is a finite binary string that starts with $\mathtt{1}$; that is to say that every $n\in\mathbb{N}$ can be written as $n=(\!(x_kx_{k-1}\ldots x_2x_1x_0)\!)_{\mathbf{2}}=\sum_{i=0}^{k}x_i2^i$, where $x_i\in\{\mathtt{0},\mathtt{1}\}$, for $i=0,1,2,\ldots,k-1$, and $x_k=\mathtt{1}$. Every real number α in the unit interval $(0,1]$ has a binary expansion (in base 2) as $\alpha=(\!(\mathtt{0}\,\textbf{.}\,x_1x_2x_3\ldots)\!)_{\mathbf{2}}=\sum_{i=1}^{\infty}x_i2^{-i}$, where $x_i\in\{\mathtt{0},\mathtt{1}\}$, for $i=1,2,3,\ldots$ (see [12]). This expansion could be finite or infinite. ✧

Example 2.4 (binary expansion in base 2):
We have $9=(\!(\mathtt{1001})\!)_{\mathbf{2}}$, $26=(\!(\mathtt{11010})\!)_{\mathbf{2}}$, $41=(\!(\mathtt{101001})\!)_{\mathbf{2}}$, $1=(\!(\mathtt{0}\,\textbf{.}\,\mathtt{111}\ldots)\!)_{\mathbf{2}}$, and also $\frac{9}{32}=(\!(\mathtt{0}\,\textbf{.}\,\mathtt{01001})\!)_{\mathbf{2}}=(\!(\mathtt{0}\,\textbf{.}\,\mathtt{01000111}\ldots)\!)_{\mathbf{2}}$. ✧

Remark 2.5 (Uniqueness):
Every natural number has a unique binary expansion, which is a finite binary string. The infinite binary expansion of any real number in $(0,1]$ is unique. ✧

Definition 2.6 ($\mathbb{I}_\sigma,\mathfrak{L}$):
For a binary string $\sigma\in\mathbf{\Sigma}$, let $\mathbb{I}_\sigma$ be the interval $\big((\!(\mathtt{0}\,\textbf{.}\,\sigma)\!)_{\mathbf{2}},(\!(\mathtt{0}\,\textbf{.}\,\sigma\mathtt{111}\ldots)\!)_{\mathbf{2}}\big]$, which consists of all the real numbers in $(0,1]$ whose infinite binary expansions after $\mathtt{0}\,\textbf{.}$ contain σ as a prefix (cf. [12]). Denote the Lebesgue measure on the real line by $\mathfrak{L}$. ✧

Example 2.7 ($\mathbb{I}_\sigma,\mathfrak{L}$):
We have $\mathbb{I}_{\{\mathtt{0}\}}=(0,\frac{1}{2}]$, $\mathbb{I}_{\{\mathtt{1}\}}=(\frac{1}{2},1]$, $\mathbb{I}_{\{\mathtt{00}\}}=(0,\frac{1}{4}]$, and $\mathbb{I}_{\{\mathtt{01001}\}}=(\frac{9}{32},\frac{5}{16}]$. The Lebesgue measures (lengths) of these intervals are $\mathfrak{L}(\mathbb{I}_{\{\mathtt{0}\}})=\frac{1}{2}$, $\mathfrak{L}(\mathbb{I}_{\{\mathtt{1}\}})=\frac{1}{2}$, $\mathfrak{L}(\mathbb{I}_{\{\mathtt{00}\}})=\frac{1}{4}$, and finally $\mathfrak{L}(\mathbb{I}_{\{01001\}})=\frac{1}{32}$. ✧

Lemma 2.8 ($\mathbb{I}_\sigma\setminus\{1\}\subseteq(0,1)$, $\mathfrak{L}(\mathbb{I}_\sigma)=2^{-|\sigma|}$, $\mathbb{I}_\sigma\cap\mathbb{I}_{\sigma'}$, $\mathfrak{L}(\bigcup_{\sigma\in S}\mathbb{I}_\sigma)=\Omega_S$ *for prefix-free* S):
Let $\sigma,\sigma'\in\mathbf{\Sigma}$ be fixed.
(1) The interval $\mathbb{I}_\sigma$ is a half-open subinterval of $(0,1]$, i.e., $\mathbb{I}_\sigma\subseteq(0,1]$.
(2) The length of $\mathbb{I}_\sigma$ is $\frac{1}{2^{|\sigma|}}$, i.e., $\mathfrak{L}(\mathbb{I}_\sigma)=2^{-|\sigma|}$.
(3) If σ is not a prefix of σ' and σ' is not a prefix of σ, then $\mathbb{I}_\sigma\cap\mathbb{I}_{\sigma'}=\emptyset$.
(4) If $S\subseteq\mathbf{\Sigma}$ is prefix-free, then $\mathfrak{L}(\bigcup_{\varsigma\in S}\mathbb{I}_\varsigma)=\Omega_S$.

Proof:

(1) is trivial; for (2) notice that

$$\begin{aligned}\mathfrak{L}(\mathbb{I}_\sigma) &= (\!(\texttt{0.}\sigma\texttt{111}\ldots)\!)_\mathbf{2} - (\!(\texttt{0.}\sigma)\!)_\mathbf{2} \\ &= (\!(\texttt{0.}\,\underbrace{\texttt{0}\ldots\texttt{0}}_{|\sigma|\text{-times}}\,\texttt{111}\ldots)\!)_\mathbf{2} \\ &= \textstyle\sum_{j=1}^{\infty} 2^{-(|\sigma|+j)} \\ &= 2^{-|\sigma|}.\end{aligned}$$

(3) If $\alpha \in \mathbb{I}_\sigma \cap \mathbb{I}_{\sigma'}$, then $\alpha = (\!(\texttt{0.}\sigma x_1x_2x_3\ldots)\!)_\mathbf{2}$ and $\alpha = (\!(\texttt{0.}\sigma' y_1y_2y_3\ldots)\!)_\mathbf{2}$, where the sequences $\{x_i\}_{i>0}$ and $\{y_i\}_{i>0}$ are not all 0. Thus, by Remark 2.5, the identity $(\!(\texttt{0.}\sigma x_1x_2x_3\ldots)\!)_\mathbf{2} = (\!(\texttt{0.}\sigma' y_1y_2y_3\ldots)\!)_\mathbf{2}$ implies that either σ should be a prefix of σ' or σ' should be a prefix of σ.

(4) We have $\mathfrak{L}(\bigcup_{\varsigma\in S}\mathbb{I}_\varsigma) = \sum_{\varsigma\in S}\mathfrak{L}(\mathbb{I}_\varsigma)$ since $\mathbb{I}_\varsigma$'s are pairwise disjoint by item (3). The result follows now from item (2) and Definition 1.6. ❑

Proposition 2.9 (Kraft's Inequality, 1949):

For every prefix-free $S \subseteq \mathbf{\Sigma}$, we have $\Omega_S \leqslant 1$.

Proof:

By Lemma 2.8, item (4), we have $\Omega_S = \mathfrak{L}(\bigcup_{\sigma\in S}\mathbb{I}_\sigma)$, and $\bigcup_{\sigma\in S}\mathbb{I}_\sigma \subseteq (0,1]$ holds by item (1) of Lemma 2.8. Therefore, $\Omega_S \leqslant \mathfrak{L}(0,1] = 1$. ❑

For an alternative proof of Proposition 2.9, see, e.g., [14, Thm. 11.4, pp. 182-3]. Let us notice that the converse of Kraft's inequality is not true, since, as we saw in Examples 1.7 and 2.2, $\Omega_{\{\texttt{0},\texttt{00}\}} = \frac{3}{4} < 1$, but the set $\{\texttt{0},\texttt{00}\}$ is not prefix-free.

One way to ensure that the set of all the programs becomes prefix-free is to adopt the following convention:

Convention 2.10 (Prefix-Free Programs):

Every program ends with the "`END`" command (see [13, p. 3]). This command can appear nowhere else in the program, only at the very end. ✧

Every other sub-routine may start with "`begin`" and finish with "`end`," just like the programs of Example 1.5.

Example 2.11 (Prefix-Free Programs):

The Program *i* in the following table is a prefix of Program *ii* (and a suffix of Program *iii*).

Program *i*	Program *ii*	Program *iii*
`BEEP`	`BEEP` `PRINT` "error!"	`PRINT` "error!" `BEEP`

With Convention 2.10, the programs should look like the following:

Program I	Program II	Program III

`BEEP`
`END`
	`BEEP`
	`PRINT` "error!"
	`END`
		`PRINT` "error!"
		`BEEP`
		`END`

Program I is not a prefix of Program $\mathbb{I}$ (though, even with the above convention, Program I is a suffix of Program $\mathbb{III}$, which is not a problem). ✧

From now on, let us be given a ***fixed programming language*** by Convention 2.10. A question that comes to mind is:

Question 2.12 (Is Ω_S a Probability?):
Why can the number Ω_S be interpreted as the *probability* that a randomly given binary string $\sigma \in \mathbf{\Sigma}$ belongs to S? Even when $S \subseteq \mathbf{\Sigma}$ is a prefix-free set. ✧

Let us repeat that the number Ω_S could be greater than one for some sets S of finite binary strings (Example 1.7), but if the set S is prefix-free, then Ω_S is a number between 0 and 1 (Proposition 2.9). Let us also note that Ω satisfies Kolmogorov's axioms of a measure: $\Omega_{\bigcup_i S_i} = \sum_i \Omega_{S_i}$ for every family $\{S_i\}_i$ of pairwise disjoint sets; thus, $\Omega_\emptyset = 0$. But it is not a probability measure. Restricting the sets to the prefix-free ones will not solve the problem, as they are not closed under unions (Example 2.2). Now that, by Convention 2.10, all the programs are prefix-free, a special case of Question 2.12 is:

Question 2.13 (Is $\mathbf{\Omega}$ a Halting Probability?):
Why can the number $\mathbf{\Omega}$ be said to be the *halting probability* of the randomly chosen finite binary strings? ✧

Unfortunately, many scholars seem to have believed that the number $\mathbf{\Omega}$ is the halting probability of input-free programs; see, e.g., [9, 2, 14, 4, 15]. Even though the Ω's of prefix-free sets are non-greater than one, Ω is not a probability measure, even when restricted to the prefix-free sets, as those sets are not closed under disjoint unions. Restricting the sets to the subsets of a fixed prefix-free set whose Ω is 1 (such as $\mathscr{C}$ in Example 1.7) can solve the problem. But for the input-free programs, even with Convention 2.10, we do not have this possibility:

Lemma 2.14 ($\Omega_{\mathbb{P}} \neq 1$):
$\Omega_{\mathbb{P}} < 1$.

Proof:
Find a letter or a short string of letters (such as `X` or `XY`, etc.) that is not a prefix of any command, and no program can be a prefix of it. Let $\mathfrak{X}$ be its `ASCII` code, and put $P' = \mathbb{P} \cup \{\mathfrak{X}\}$. The set P' is still prefix-free, and so Kraft's inequality (Proposition 2.9) can be applied to it: $\Omega_{\mathbb{P}} + 2^{-|\mathfrak{X}|} = \Omega_{P'} \leqslant 1$. Since $2^{-|\mathfrak{X}|} > 0$, then we have $\Omega_{\mathbb{P}} < 1$. ❑

For making Ω a probability measure, we suggest a two-fold idea:
(1) We consider sets of input-free programs only, and
(2) We divide their Omega by $\Omega_{\mathbb{P}}$ to get a probability measure.

Definition 2.15 ($\mho_S$):
For a set $S \subseteq \mathbb{P}$ of input-free programs, let $\mho_S = \dfrac{\Omega_S}{\Omega_{\mathbb{P}}}$. ✧

It is easy to verify that this *is* a probability measure: we have $\mho_\emptyset = 0$, $\mho_{\mathbb{P}} = 1$, and for every indexed family $\{S_i \subseteq \mathbb{P}\}_i$ of pairwise disjoint sets of input-free programs, we have $\mho_{\bigcup_i S_i} = \sum_i \mho_{S_i}$.

2.1 Summing Up

Let us recapitulate. The number $\mathbf{\Omega}$ (Definition 1.9) was meant to be “the probability that a computer program whose bits are generated one by one by independent tosses of a fair coin will eventually halt” [4, p. 236]. But the fact of the matter is that if we generate a finite binary code by tossing a fair coin bit by bit, then it is very probable that the resulted string is not the binary code of a program at all. It is also highly probable that it is the code of a program that takes some inputs (see Example 1.5). Lastly, if the generated finite binary string is the binary code of an input-free program, then we are allowed to ask whether it will eventually halt after running. After all this contemplation, we may start defining or calculating the probability of halting.

The way $\mathbf{\Omega}$ was defined works for any prefix-free set of finite binary strings (Definition 1.6). Kraft’s inequality (Proposition 2.9) ensures that the number Ω_S, for every prefix-free set S, lies in the interval $[0,1]$. But why on earth can Ω_S be called the probability that a randomly given finite binary string belongs to S? (Question 2.12). The class of all prefix-free sets is not closed under disjoint unions (Example 2.2), and there is no *sample space* for the proposed measure: the Ω of all the binary codes of the input-free programs is not equal to 1 (Lemma 2.14), even though that set is prefix-free by Convention 2.10. Summing up, there is no measure to see that $\mathbf{\Omega}$ is the halting probability of a randomly given finite binary string, and the answer to Question 2.13 is a big “no”.

Even though Ω satisfies Kolmogorov’s axioms of a measure, it is not a probability measure, as some sets get measures bigger than one. Restricting the sets to the prefix-free ones will not solve the problem, as they are not closed under union. Restricting the sets to the subsets of a fixed prefix-free set whose Ω is 1 can solve the problem by making Ω a probability measure; so can restricting the sets to the subsets of a fixed prefix-free set (such as $\mathbb{P}$) and then dividing the Ω’s of its subsets by the Ω of that fixed set (just like Definition 2.15).

This was our proposed remedy. Take the sample space to be $\mathbb{P}$, the set of the binary codes of all the input-free programs. Then, for every set S of (input-free) programs ($S \subseteq \mathbb{P}$), let $\mho_S = \frac{\Omega_S}{\Omega_{\mathbb{P}}}$ (Definition 2.15). This is a *real* probability measure that satisfies Kolmogorov’s axioms. Now, the *new* halting probability is $\boldsymbol{\mho} = \mho_{\mathbb{H}} = \frac{\mathbf{\Omega}}{\Omega_{\mathbb{P}}}$. Dividing $\mathbf{\Omega}$ by a computable real number ($\Omega_{\mathbb{P}}$) does make it look more like a (conditional) probability, but will not cause it to lose any of the non-computability or randomness properties. Our upside-down Omega, $\boldsymbol{\mho}$, should have most (if not all) of the properties of $\mathbf{\Omega}$ established in the literature.

3 The Source of Error

Let us see what possibly went wrong by reading through one of Chaitin’s books:

> let’s put **all possible** programs in a bag, shake it up, close our eyes, and pick out a program. What’s the probability that this program that we’ve just chosen at random will eventually halt? Let’s express that probability as an infinite precision binary real between zero and one. [..] You sum for each program that halts the probability of getting precisely that program by chance: $\Omega = \sum_{\text{program } p \text{ halts}} 2^{-(\text{size in bits of } p)}$. Each k-bit self-delimiting program p that halts contributes $1/2^k$ to the value of Ω. The self-delimiting program proviso is crucial: Otherwise the halting probability has to be defined for programs of **each particular size**, but it cannot be defined over **all**

programs of **arbitrary size**. [8, p. 112, original emphasis]

We are in partial agreement with Chaitin on the following matter:

Lemma 3.1 (Halting Probability of Input-Free Programs with a Fixed Length):

The halting probability of all the input-free programs with a fixed length ℓ is equal to $\sum_{p \text{ halts}}^{|p|=\ell} 2^{-|p|}$.

Proof:

Fix a number ℓ. The probability of getting a fixed binary string of length ℓ by tossing a fair coin (whose one side is '0' and the other '1') is $\frac{1}{2^\ell}$, and the halting probability of the input-free programs with length ℓ is

$$\frac{\text{the number of halting programs with length } \ell}{\text{the number of all binary strings with length } \ell} = \frac{\#\{p \in \mathbb{P} \colon p \text{ halts } \& \ |p| = \ell\}}{2^\ell},$$

since there are 2^ℓ binary strings of length ℓ (see [12]). Thus, the halting probability of programs with length ℓ can be written as $\sum_{p \text{ halts}}^{|p|=\ell} 2^{-|p|}$. ❑

Definition 3.2 ($\mathscr{N}(\ell)$):

Let $\mathscr{N}(\ell)$ denote the number of halting input-free programs of length ℓ ([15]). ✧

So, $\mathbf{\Omega}$ can be written as $\sum_{\ell=1}^{\infty} \mathscr{N}(\ell) 2^{-\ell}$; see [15, p. 1]. By what we quoted above, from [8], according to Chaitin (and almost everybody else), the halting probability of programs is $\sum_{\ell=1}^{\infty} \mathscr{N}(\ell) 2^{-\ell} = \sum_{p \text{ halts}} 2^{-|p|} (= \mathbf{\Omega})$! Let us see why we believe this to be an error. First, we take a look at an elementary example.

Example 3.3 (One Tail or Two Heads):

Let us toss a fair coin once or twice and compute the probability of getting either one tail or two heads. For the number of tosses, we choose it randomly at the beginning, by, say, taking one ball from an urn that contains two balls with labels 1 and 2 (each ball has a unique label; one has 1 and the other has 2). If we get the ball with label 1 out of the urn, then we toss the coin once, and if we get the other ball, the one with label 2, then we toss the coin twice. We wish to see what the probability of getting one tail (T) or two heads (HH) could be. Our event is $E' = \{\mathtt{T}, \mathtt{HH}\}$, and our sample space is $\mathbb{S} = \{\mathtt{H}, \mathtt{T}, \mathtt{HH}, \mathtt{HT}, \mathtt{TH}, \mathtt{TT}\}$. From the method by which the number $\mathbf{\Omega}$ is defined in the literature, the probability is assumed to be $\Omega_{E'} = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$. But we can show that this is impossible under any reasonable probability measure (note that E' is both a prefix-free and a suffix-free set). If p is the probability of getting H, then it will be the probability of getting T too. Similarly, the probability of each of HH, HT, TH, and TT is a fixed number q. We should have $2p + 4q = 1$, the probability of the whole of the sample space $\mathbb{S}$. Then, the probability of E' will be $p + q$. Since $p + q = \frac{2p+2q}{2} \leqslant \frac{2p+4q}{2} = \frac{1}{2} < \frac{3}{4} = \Omega_{E'}$, then $\Omega_{E'}$ is not the probability of E' (getting one tail or two heads by tossing a fair coin randomly once or twice) under any reasonable probability measure.

Let us also observe that for the prefix-free set $P' = \{\mathtt{T}, \mathtt{HH}, \mathtt{HT}\}$ we have $\Omega_{P'} = 1$, but the probability of P' (under any probability measure p, q with $2p + 4q = 1$) is $p + 2q = \frac{2p+4q}{2} = \frac{1}{2}$, which is the half of $\Omega_{P'}$; see also Example 3.5 below. ✧

Theorem 3.4 (Halting Probability [by any measure] $<\mathbf{\Omega}$):

The halting probability of input-free programs is less than $\mathbf{\Omega}$ under any probability measure on $\mathbf{\Sigma}$.

Proof:

For every positive integer ℓ, let $\boldsymbol{\pi}_\ell$ be the probability of an(y) element of $\mathbf{\Sigma}$ with length ℓ. Therefore, $\sum_{\ell=1}^{\infty} 2^\ell \boldsymbol{\pi}_\ell = 1$, since there are 2^ℓ binary strings of length ℓ. The halting probability (with the probability measure $\boldsymbol{\pi}$) is then $\sum_{\ell=1}^{\infty} \mathscr{N}(\ell)\boldsymbol{\pi}_\ell$; see Definition 3.2. Let $2^m\boldsymbol{\pi}_m$ be the maximum of $\{2^\ell\boldsymbol{\pi}_\ell\}_{\ell=1}^{\infty}$. We distinguish two cases:

(1) If $2^m\boldsymbol{\pi}_m = 1$, then for every $\ell \neq m$, we should have $\boldsymbol{\pi}_\ell = 0$. Hence,

$$\sum_{\ell=1}^{\infty} \mathscr{N}(\ell)\boldsymbol{\pi}_\ell = \mathscr{N}(m)\boldsymbol{\pi}_m = \mathscr{N}(m)2^{-m} < \sum_{\ell=1}^{\infty} \mathscr{N}(\ell)2^{-\ell} = \mathbf{\Omega},$$

since there exists some $\ell \neq m$ with $\mathscr{N}(\ell) > 0$.

(2) So, we can assume that $2^m\boldsymbol{\pi}_m < 1$. In this case,

$$\sum_{\ell=1}^{\infty} \mathscr{N}(\ell)\boldsymbol{\pi}_\ell = \sum_{\ell=1}^{\infty} \mathscr{N}(\ell)2^{-\ell}\cdot 2^\ell\boldsymbol{\pi}_\ell \leqslant 2^m\boldsymbol{\pi}_m \sum_{\ell=1}^{\infty} \mathscr{N}(\ell)2^{-\ell} < \sum_{\ell=1}^{\infty} \mathscr{N}(\ell)2^{-\ell} = \mathbf{\Omega}.$$

Therefore, regardless of the probability measure ($\boldsymbol{\pi}$), the number $\mathbf{\Omega}$ exceeds the probability of obtaining an input-free halting program by tossing a fair coin a finite (but unbounded) number of times. ❑

Thus, there is no reason to believe that the halting probability (of "all programs of arbitrary size") is $\sum_{p \text{ halts}} 2^{-|p|} (=\mathbf{\Omega})$. As pointed out by Chaitin, the series $\sum_{p \text{ halts}} 2^{-|p|}$ could be greater than 1, or may even diverge, if the set of programs is not taken to be prefix-free (what "took ten years until [he] got it right"). So, the fact that, for prefix-free programs, the real number $\sum_{p \text{ halts}} 2^{-|p|}$ lies between 0 and 1 (by Kraft's inequality, Proposition 2.9) does not make it a probability of finite strings. As we showed above, the number $\mathbf{\Omega}$ is not the probability of halting the randomly given finite binary strings by *any* probability measure on $\mathbf{\Sigma}$.

Example 3.5 ($E, P, \mathscr{C}$):

Let us fix a probability measure $\boldsymbol{\pi}$ on $\mathbf{\Sigma}$; hence, for each positive integer ℓ, the probability of any element of $\mathbf{\Sigma}$ with length ℓ is $\boldsymbol{\pi}_\ell$, and so $\sum_{\ell=1}^{\infty} 2^\ell\boldsymbol{\pi}_\ell = 1$. Consider the prefix-free sets $E = \{\mathtt{1},\mathtt{00}\}$, $P = \{\mathtt{1},\mathtt{00},\mathtt{01}\}$ (cf. Example 3.3), and $\mathscr{C} = \{\mathtt{1}\} \cup \{\mathtt{01}^n\mathtt{0}\}_{n=0}^{\infty}$ (in Example 1.7). We have $\Omega_E = \frac{3}{4}$, and $\Omega_P = \Omega_{\mathscr{C}} = 1$. The probability of E is

$$\boldsymbol{\pi}(E) = \boldsymbol{\pi}_1 + \boldsymbol{\pi}_2 \leqslant \tfrac{1}{2}(\textstyle\sum_{\ell=1}^{\infty} 2^\ell\boldsymbol{\pi}_\ell) = \tfrac{1}{2} < \tfrac{3}{4} = \Omega_E.$$

Similarly, the probability of P is

$$\boldsymbol{\pi}(P) = \boldsymbol{\pi}_1 + 2\boldsymbol{\pi}_2 \leqslant \tfrac{1}{2}(\textstyle\sum_{\ell=1}^{\infty} 2^\ell\boldsymbol{\pi}_\ell) = \tfrac{1}{2} < 1 = \Omega_P,$$

and the probability of $\mathscr{C}$ is

$$\boldsymbol{\pi}(\mathscr{C}) = \textstyle\sum_{\ell=1}^{\infty} \boldsymbol{\pi}_\ell \leqslant \tfrac{1}{2}(\sum_{\ell=1}^{\infty} 2^\ell\boldsymbol{\pi}_\ell) = \tfrac{1}{2} < 1 = \Omega_{\mathscr{C}}.$$ ✧

Let us see one of the most recent explanations as to why $\mathbf{\Omega}$ is considered to be the halting probability of input-free programs.

> Given a prefix-free machine M, one can consider the 'halting probability' of M, defined by $\Omega_M = \sum_{M(\sigma)\downarrow} 2^{-|\sigma|}$. The term 'halting probability' is justified by the following observation: a prefix-free machine M can be naturally extended to a partial functional from 2^ω, the set of infinite binary sequences, to $2^{<\omega}$, where for $X \in 2^\omega$, $M(X)$ is defined to be $M(\sigma)$ if some $\sigma \in \mathrm{dom}(M)$ is a prefix of X, and $M(X)\uparrow$ otherwise. The prefix-freeness of M on finite strings ensures that this extension is well-defined. With this point of view,

> Ω_M is simply $\mu\{X \in 2^\omega : M(X)\downarrow\}$, where μ is the uniform probability measure (a.k.a. Lebesgue measure) on 2^ω, that is, the measure where each bit of X is equal to 0 with probability 1/2 independently of all other bits. [3, p. 1613]

See [14, p. 207] for a similar explanation. So, the expression "halting probability" refers to the probability of some real numbers, not of finite binary strings. Let us consider a randomly given real number $\alpha \in (0,1]$. The probability that α is less than $\frac{1}{4}$ is, of course, $\frac{1}{4}$, since the length of $(0,\frac{1}{4})$ is $\frac{1}{4}$. The probability that α is rational is 0. Let us calculate the probability that the finite string `01001` is a prefix of the unique infinite binary expansion after `0.` of α (see Definition 2.3). If α is like that, then $\alpha = (\!(\mathtt{0.01001}x_1x_2x_3\ldots)\!)_{\mathbf{2}}$ for some bits $x_1, x_2, x_3, \cdots$. This means that α belongs to the interval $\mathbb{I}_{\{01001\}}$ (see Definition 2.6), so the probability is $\frac{1}{32}$ (see Example 2.7).

Lemma 3.6 (Probability of Some Events on Real Numbers):

(1) The probability that a randomly given real $\alpha \in (0,1]$ has a fixed finite binary string σ as a prefix in its infinite binary expansion after 0. is $\mathfrak{L}(\mathbb{I}_\sigma)$.

(2) The probability that a randomly given real $\alpha \in (0,1]$ has a prefix from a fixed set of finite binary strings $S \subseteq \mathbf{\Sigma}$ in its infinite binary expansion after 0. is $\mathfrak{L}(\bigcup_{\sigma\in S}\mathbb{I}_\sigma)$.

Proof:

(1) Every such α belongs to the interval $\mathbb{I}_\sigma$ (see Definition 2.6). So, the probability is $\mathfrak{L}(\mathbb{I}_\sigma)$; cf. [12]. Item (2) follows similarly. ❑

Corollary 3.7 (Omega Numbers as Probabilities of Real Numbers):

(1) The probability that a randomly given real $\alpha \in (0,1]$ has a prefix from a fixed prefix-free set of finite binary strings $S \subseteq \mathbf{\Sigma}$ in its infinite binary expansion after 0. is Ω_S.

(2) Chaitin's $\mathbf{\Omega}$ is the probability that the unique infinite binary expansion after 0. of a randomly given real $\alpha \in (0,1]$ contains a finite binary strings as a prefix that is the binary code of a halting input-free program.

Proof:

(1) follows from Lemma 3.6(2) and Lemma 2.8(4). Item (2) is a special case of (1) when $S = \mathbb{H}$ (see Definition 1.8). ❑

After all, $\mathbf{\Omega}$ is the probability of something, an event on real numbers.

3.1 Some Suggestions

Definition 3.8 (integer code, $\mathscr{H}$):

Every finite binary string $\sigma \in \mathbf{\Sigma}$ has an integer code defined as $(\!(1\sigma)\!)_{\mathbf{2}} - 1$, illustrated by the following table.

binary string	0	1	00	01	10	11	000	001	010	011	...
integer code	1	2	3	4	5	6	7	8	9	10	···

Let $\mathscr{H}$ be the set of the integer codes of all the strings in $\mathbb{H}$ (see Definition 1.8). ✧

Example 3.9 (integer code):

The integer code of the binary string `01001` is 40, and the finite binary string with the integer code 25 is `1010` (see Example 2.4). ✧

Chaitin's $\mathbf{\Omega}$ has many interesting properties that have attracted the attention of the brightest minds and made them publish papers in the most prestigious journals and collection books. Most properties of $\mathbf{\Omega}$, which we proved not to be a probability of random strings, are also possessed by $K = \sum_{n\in\mathscr{H}} 2^{-n}$ (see [9, p. 33]). This number is in the interval $(0,1)$, so it can be a halting probability with a good measure: for a set of positive integers $S \subseteq \mathbb{N}^+$, let $\mathfrak{p}(S) = \sum_{n\in S} 2^{-n}$. Then all the probability axioms are satisfied: $\mathfrak{p}(\mathbb{N}^+) = 1$ and $\mathfrak{p}(\bigcup_i S_i) = \sum_i \mathfrak{p}(S_i)$ for every pairwise disjoint $\{S_i \subseteq \mathbb{N}^+\}_i$. One question now is: why not take this number as *a halting probability*? Notice that this has some non-intuitive properties: if E is the set of all the even positive integers and O is the set of all the odd positive integers, then *the probability* that a binary string has an *even* integer code becomes $\mathfrak{p}(E) = \sum_{n\in E} 2^{-n} = \frac{1}{3}$, and *the probability* that a binary string has an *odd* integer code turns out to be $\mathfrak{p}(O) = \sum_{n\in O} 2^{-n} = \frac{2}{3}$, twice the evenness probability!

For $\mathbf{\Omega}$, the geometric distribution (see, e.g., [12]) is in play, with the parameter $p = \frac{1}{2}$. Why not take other parameters, such as $p = \frac{1}{3}$ and then define a halting probability as $\sum_{\sigma\in\mathbb{H}} 3^{-|\sigma|}$ (or $\sum_{n\in\mathscr{H}} 2\cdot 3^{-n}$)? Note that $\sum_{n>0} 2\cdot 3^{-n} = 1$, and Kraft's inequality applies here too: $\sum_{\sigma\in S} 3^{-|\sigma|} \leqslant 1$ for every prefix-free set $S \subseteq \mathbf{\Sigma}$. Or, why not Poisson's distribution (see, e.g., [12]) with a parameter λ? Then, a halting probability could be $\sum_{n\in\mathscr{H}} \frac{e^{-\lambda}\lambda^{-n}}{n!}$. One key relation in defining K is the elementary formula $\sum_{n>0} 2^{-n} = 1$. Let $\{\alpha_n\}_{n>0}$ be any sequence of positive real numbers such that $\sum_{n>0} \alpha_n = 1$. Then one can *define* a halting probability as $\sum_{n\in\mathscr{H}} \alpha_n$ or $\sum_{\sigma\in\mathbb{H}} 2^{-|\sigma|}\alpha_{|\sigma|}$. Most, if not all, of the properties of $\mathbf{\Omega}$ should be possessed by these new *probabilities*. This seems like a wild, open area to explore.

4 The Conclusion

Chaitin's $\mathbf{\Omega}$ number *is not* the probability that a randomly given finite binary string is the binary code of a halting input-free program under *any* probability measure. It *is* the probability that the unique infinite binary expansion after $0.$ of a randomly given real number in the unit interval has a prefix that is the binary code of a halting input-free program. There is no unique *halting probability* of finite binary strings, and one can get different values for it by different probability measures (over a fixed prefix-free programming language).

References

[1] BARMPALIAS, GEORGE (2020); "Aspects of Chaitin's Omega", in: J.N.Y. Franklin & C.P. Porter (eds.), *Algorithmic Randomness: Progress and Prospects*, Lecture Notes in Logic 50, Cambridge University Press, pp. 175–205. DOI: `10.1017/9781108781718.007`

[2] BECHER, VERÓNICA & FIGUEIRA, SANTIAGO & GRIGORIEFF, SERGE & MILLER, JOSEPH S. (2006); Randomness and Halting Probabilities, *The Journal of Symbolic Logic* 71:4, pp. 1411–1430. DOI: `10.2178/jsl/1164060463`

[3] BIENVENU, LAURENT & CSIMA, BARBARA F. & HARRISON-TRAINOR, MATTHEW (2021); Some Questions of Uniformity in Algorithmic Randomness, *The Journal of Symbolic Logic* 86:4, pp. 1612–1631. DOI: 10.1017/jsl.2021.58

[4] CALUDE, CRISTIAN S. & CHAITIN, GREGORY J. (2010); WHAT IS ... a Halting Probability?, *Notices of the American Mathematical Society* 57:2, pp. 236–237. PAPER ID: rtx100200236p

[5] CHAITIN, GREGORY J. (1975); A Theory of Program Size Formally Identical to Information Theory, *Journal of the Association for Computing Machinery* 22:3, pp. 329–340. DOI: 10.1145/321892.321894

[6] CHAITIN, GREGORY J. (1988); Randomness in Arithmetic, *Scientific American* 259:1 (July), pp. 80–85. JSTOR: 24989161

[7] CHAITIN, GREGORY J. (1995); Randomness in Arithmetic and the Decline and Fall of Reductionism in Pure Mathematics, *Chaos, Solitons & Fractals* 5:2, pp. 143–159. DOI: 10.1016/0960-0779(93)E0017-6

[8] CHAITIN, GREGORY J. (2005); *Meta Math!: The Quest for Omega*, Pantheon Books. ISBN: 9780375423130 (*the page number in the text refers to the pre-print* arXiv:math/0404335v7).

[9] GARDNER, MARTIN [& BENNETT, CHARLES H.] (1979); Mathematical Games: The Random Number Omega Bids Fair to Hold the Mysteries of the Universe, *Scientific American* 241:5 (November), pp. 20–35. JSTOR: 24965329

[10] GLAZIER, HARRIET E. (1916); The Mathematics of Common Things, *School Science and Mathematics* 16:8, pp. 667–674. DOI: 10.1111/j.1949-8594.1916.tb01753.x

[11] KRAFT, LEON G. (1949); *A Device for Quantizing, Grouping, and Coding Amplitude-Modulated Pulses*, M.Sc. Thesis, MIT. URL: https://b2n.ir/m52630

[12] LINDE, WERNER (2024); *Probability Theory: A First Course in Probability Theory and Statistics*, De Gruyter (2nd ed.). ISBN: 9783111324845

[13] MONRO, DONALD M. (1978); *basic* BASIC*: An Introduction to Programming*, Edward Arnold (Publishers) Ltd. (1st ed.). ISBN: 0713127325 INTERNET ARCHIVE: basicbasicintrod0000monr

[14] ROZENBERG, GRZEGORZ & SALOMAA, ARTO (2007); "The Secret Number: An Exposition of Chaitin's Theory", in: C. S. Calude (ed.), *Randomness and Complexity: From Leibniz to Chaitin* (dedicated to Gregory J. Chaitin on the occasion of his 60th birthday), World Scientific, pp. 175–215. DOI: 10.1142/9789812770837_0011

[15] SCHMIDHUBER, CHRISTOF (2022); Chaitin's Omega and an Algorithmic Phase Transition, *Physica A: Statistical Mechanics and its Applications*, 568 (Article 126458) pp. 1–15. DOI: 10.1016/j.physa.2021.126458