Turing's note on normal numbers

Verónica Becher

Universidad de Buenos Aires & CONICET

In an unpublished manuscript[1] with title "A note on normal numbers" Alan Turing gives the first explicit algorithm to compute a *normal* real number. Normality demands that the infinite expansion of a real number be seriously balanced: a number is normal *in a given scale* (numbering base), if every block of digits of the same length occurs with the same limit frequency in the expansion of the number expressed in that scale. For example, if a number is normal in the scale of two, the digits "0" and "1" occur, in the limit, half of the times; each of the blocks "00", "01", "10" and "11" occur one fourth of the times, and so on. A real number that is normal to *every scale* is called absolutely normal, or just *normal*.[2] Émile Borel stated this definition in 1909 and proved the existence of normal numbers, showing that, indeed, almost all numbers are normal. Borel's proof is based on measure theory, and being purely existential, it gives no method of constructing an example of a normal number (Borel, E., Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo* 27, 247–271, 1909).

With his note Turing gives an answer to the problem of finding an example of a normal number, raised by Borel. Turing gives, first, a constructive proof that almost all numbers are normal, and then, an algorithm to construct normal numbers, which leads to his computable examples.

As defined by Turing in his breakthrough article "On computable numbers with an application to the Entscheidungsproblem" (*Proceedings of the London Mathematical Society* 2:42, 230–265, 1936), the computable real numbers are those whose infinite expansion can be generated by a mechanical (finitary) method, outputting each of its digits, one after the other. There is no evident reason for the normal numbers to have a non-empty intersection with the computable numbers. A measure-theoretic argument is not enough to see that these sets intersect: the set of normal numbers in the unit interval has Lebesgue measure one, but the computable numbers are just countable, hence they form a null set. Along the note Turing uses the term *constructive*, but never uses the term *computable*, which would have better expressed the finitarily-based constructiveness he actually achieves.

Email address: vbecher@dc.uba.ar (Verónica Becher)

[1]A typewritten document together with a handwritten draft is in Turing's archive in King's College, Cambridge; the scanned versions are available on the Web in http://www.turingarchive.org.

[2]A thorough presentation of normal numbers can be read in Kuipers, L., Niederreiter, H., *Uniform distribution of sequences*, Dover Books on Mathematics, New York, 2006.

Turing's note is undated. Presumably it was written not much after 1936, because the conserved draft consists of six pages handwritten in the back of the galley proofs of his celebrated *"On computable numbers..."*. Turing's calligraphy is hard to follow, there are numerous crossing outs, and each page starts in small lettering that slightly grows towards the end of the page. The typewritten document, in which only mathematical formulae are handwritten, is much more complete. Turing's note remained unpublished until its inclusion in the *Collected Works of A.M. Turing: Pure Mathematics*, J.L. Britton editor, North Holland, 1992.

In eleven lines of the draft that Turing did not include in the typewritten document, he appraises the results of his note. Turing cites David Champernowne's[3] example of normality in the scale of ten —but not proved normal in any other scale— and says that it may also be natural that an example of a *normal* number (i.e., normal in *every* scale) be demonstrated as such and written down. Then he writes "*this note cannot, therefore, be considered as providing* <u>*convenient*</u> *examples of normal numbers*"[4].

Champernowne's number is formed by writing down all the positive integers in order, in decimal notation, 0.12345678910111213... [5] Turing's reference to it suggests what he would have considered to be a *convenient* example: a number with a simple mathematical definition and easily computable. According to the modern theory of computational complexity, which was only developed in the 1960's and required the Turing machine as its computational model, Turing's algorithm has exponential complexity: the number of operations needed to compute the n-th digit of the output sequence is exponential in n. We now know that this is intractable for every past or present computer. One can interpret Turing's negative assessment of the number produced by his algorithm as a trail of his intuitive considerations on its computational complexity. Years later, in his article *"Solvable and unsolvable problems"* (*Science News* 31, 1954, and included in the *Collected Works*) he will write, tangentially, about algorithmic solutions that cause combinatorial explosion .

Still in the handwritten draft Turing says that the purpose of his note is, rather, to counter the then dominant idea that the existence proof of normal numbers provides no example of them. And he adds that the arguments in the note, in fact, follow the existence proof fairly closely. Here Turing is obviously referring to the proof of the measure of normal numbers.[6]

---

[3]David Champernowne was the first friend that Turing made when he entered King's College Cambridge. This is reported of Andrew Hodges's superb biography *Alan Turing: the Enigma*, Walker and Company, New York, 2000.

[4]Turing's underlining.

[5]To prove it normal in the scale of ten, Champernowne ingeniously bounds the number of occurrences of each block of digits in the initial segments of the sequence (The construction of decimals in the scale of ten, *Journal of the London Mathematical Society* 8, 254–260, 1933). In this proof it is crucial to know, explicitly, the digit in each position of the sequence. The technique is not relevant to Turing's note.

[6]A version of this proof appears in the book by Hardy, G.H. and Wright, E.M., *An Introduction to the Theory of Numbers*, Oxford University Press, 1979, with first edition in 1938.

There is a letter exchange[7] between G.H. Hardy and Turing, where Hardy recalls he searched the literature when Champernowne was doing his work "but could not find anything satisfactory anywhere". Hardy's letter ends saying that his "feeling is that Lebesgue made a proof himself that never got published". Actually, Henri Lebesgue constructed a normal number in 1909, but didn't publish it until 1917 (Sur certaines démonstrations d'existence, *Bulletin de la Société Mathématique de France* 45, 132–144). In the same journal issue, Wacław Sierpiński presented his example of a normal number, based on a seemingly simpler but equivalent characterization of normality (Démonstration élémentaire du théorème de M. Borel sur les nombres absolument normaux et détermination effective d'un tel nombre, 127–132). Both, Lebesgue and Sierpiński, give a partially constructive proof of the measure of the set of normal numbers, and define their respective examples as the limit of a set that includes all non-normal numbers (this limit point is outside the set). Their examples are not finitarily defined. At that time computability theory was not even born, so it is not surprising that neither Lebesgue nor Sierpiński used a stronger notion of constructiveness. However, these antecedents could have been the reason that Turing did not publish his construction.

Although Turing's note is incomplete, it is correct except for some minor technical errors. We completed it by giving full proofs and corrected the errors (Becher,V., Figueira,S., Picchi,R., Turing's unpublished algorithm for normal numbers. *Theoretical Computer Science* 377, 126–138, 2007). In doing so we tried to recreate Turing's ideas as accurately as possible. Turing proves two theorems. The first provides a finitarily based method to construct a set of normal real numbers in the unit interval, of Lebesgue measure exactly $1-1/k$, for a given parameter k.

Turing's theorem 1. *We can find a constructive function $c(k,n)$ of two integer variables with values in finite sets of pairs of rational numbers such that, for each k and n, if $E_{c(k,n)} = (a_1,b_1) \cup (a_2,b_2) \cup ...(a_m,b_m)$ denotes the finite union of the intervals whose rational endpoints are the pairs given by $c(k,n)$, then $E_{c(k,n)}$ is included in $E_{c(k,n-1)}$ and the measure of $E_{c(k,n)}$ is greater than $1-1/k$. And for each k, $E(k) = \bigcap_n E_{c(k,n)}$ has measure $1-1/k$ and consists entirely of normal numbers.*

The construction is uniform of the parameter k. Turing prunes the unit interval by stages. It starts with the set $E_{c(k,0)}$ equal to the whole unit interval. At stage n, the set $E_{c(k,n)}$ is the finite approximation to $E(k)$ that results from removing from $E_{c(k,n-1)}$ the points that are *not* candidates to be normal, according to the inspection of an initial segment of their expansions. At the end of this infinite construction all rational numbers have been discarded, because of their periodic structure. All irrational numbers with an unbalanced expansion have been discarded. But also many normal numbers are discarded, because their initial segments start unbalanced. Turing covers all initial segment sizes, all

[7]Letter sent by G.H. Hardy to A.M. Turing, dated June 1, Trinity College, presumably in the late 1930s. Hardy answers a letter from Turing of March 28, apologizing for not responding earlier and for not giving him a definitive satisfactory response. It is in Turing's archive in King's College and available in the digital archive with code AMTD/D/5 image 6.

scales, and all blocks, by increasing functions of the stage n. And puts a decreasing bound on the acceptable discrepancy between the actual number of blocks in the inspected initial segments and the perfect number of blocks expected by the property of normality. These functions (initial segment size, scale, block length and discrepancy) must be such that, at each stage n, the set of discarded real numbers has a small measure. To bound this measure Turing uses a constructive version of the Strong Law of Large Numbers. Thus, at each stage, finitely many intervals with rational endpoints and very small measure are removed. Turing tailors the sets $E_{c(k,n)}$ so as to have measure greater than $1 - 1/k$. The set $E(k)$ is the limit of this construction, hence it is the countable intersection of the constructed sets $E_{c(k,n)}$, and it consists entirely of normal numbers.

This construction leads to a direct proof that the property of *randomness* of real numbers implies normality. The mathematical definition of randomness is much younger than that of normality and also much younger than Turing's proof, since it is from the mid 1960s due to Per Martin Löf and Gregory Chaitin, whose different definitions were proved to be equivalent. By Martin Löf's characterization, random real numbers are those that belong to no computably definable null set. In present day terminology, Turing's construction in Theorem 1 shows that the real numbers that are not normal are properly included in a computably definable null set (the countable intersection of the complement of $E(k)$, for all k). Thus, if a real number is not normal, it can not be random.

From a more general perspective, the proof of Theorem 1 conveys the impression that Turing intuitively knew, ahead of his time, that traditional mathematical concepts equipped with finite approximations, such as measure or continuity, could be made *computational*. This line of research has become mainstream and has developed under the general name of *effective* mathematics.

Turing's second theorem gives an affirmative answer to the then outstanding question of whether there are computable normal numbers, and provides concrete instances. In fact, it gives much more:

Turing's theorem 2. *There is an algorithm that, given an integer k and an infinite sequence ν of zeros and ones, produces a normal number $\alpha(k,\nu)$ in the unit interval, expressed in the scale of two, such that in order to write down the first n digits of $\alpha(k,\nu)$. the algorithm requires at most the first n digits of ν. For a fixed k these numbers $\alpha(k,\nu)$ form a set of measure at least $1 - 2/k$.*

Our reconstruction of the proof of Turing's theorem 2 (in the aforementioned article in *Theoretical Computer Science* 377, 126–138, 2007) supersedes J.L. Britton's editorial notes on this theorem in the volume *Pure Mathematics* of the *Collected Works*[8], where it is asserted that the proof given by Turing is inadequate, and speculated that the theorem could indeed be false.

[8]Notes 7 to 12 on page 119, elaborated in pages 264 and 265.

The algorithm is uniform on the parameter k and it works by stages. Turing splits the unit interval by halves, successively. Initially it starts with the whole unit interval and at each stage it chooses either the left half or the right half of the current interval. The invariant of the algorithm is that the intersection of the current interval with the set $E(k)$ of normal numbers of Theorem 1 has positive measure. To ensure this condition at stage n it uses the finite approximation of the set $E(k)$ given by $E_{c(k,n)}$. The algorithm chooses the half of the current interval whose intersection with $E_{c(k,n)}$ reaches a minimum measure that avoids running out of measure in later stages. In case both halves reach this minimum, the algorithm uses the n-th symbol of the input sequence ν to decide. Since the chosen intervals at successive stages are nested and their measures converge to zero, their intersection contains exactly one number which must be normal. This is the number $\alpha(k,\nu)$ output by the algorithm, and it is the trace of the left/right selection at each stage.

When the input ν is a computable sequence —Turing puts the infinite sequence of all zeros— the algorithm produces a computable normal number. To prove that for a fixed k, the set of output numbers $\alpha(k,\nu)$ for all possible inputs ν has measure at least $1-2/k$, Turing bounds the measure of the unqualified intervals up to stage n, as the n first bits of the sequence ν run through all possibilities. The algorithm can be adapted to intercalate the bits of the input sequence at fixed positions of the output sequence. Thus, one obtains non-computable normal numbers in each Turing degree.

The time complexity of the algorithm is the number of needed operations to produce the n-th digit of the output sequence $\alpha(k,\nu)$. This just requires to compute, at each stage n, the measure of the intersection of the current interval with the set $E_{c(k,n)}$. Turing gives no hints on properties of the sets $E_{c(k,n)}$ that could allow for a fast calculation of their measure. The naive way does the combinatorial construction of $E_{c(k,n)}$, in time exponential in n. Turing's algorithm *verbatim* would have exponential complexity, but its correctness proof is missing in Turing's note. Our reconstruction of the algorithm —that we give together with its correctness proof— has, unfortunately, doubly exponential time complexity —because the number of intervals we consider in $E_{c(k,n)}$ is exponentially larger than in Turing's literal construction—. Also our recursive reformulation of Sierpiński's normal number is computable in doubly exponential time (Becher,V., Figueira,S., An example of a computable absolutely normal number, *Theoretical Computer Science* 270, 947–958, 2002). On the computational complexity of computable normal numbers, this is to our knowledge the best that is known today.