UNIQ Summer School July 30th, 2021

Computable and Automatic Numbers

James Worrell

1 Introduction

The topic of this lecture is computable and automatic numbers. This subject goes back to Turing's 1936 paper *On Computable Numbers*, which introduced the notion of a Turing machine (one of the central concepts in theoretical computer science). The topic of computable numbers connects algorithms, as embodied by finite-state automata and Turing machines, to number theory and numerical analysis.

Warning! These notes talk about things, such as finite-state automata and Turing machines, that are introduced much more slowly and carefully in the first year of the Computer Science degree at Oxford. I invite you to read through them and follow the pointers to further background reading, but don't expect to fully understand what is written here. The lecture will explain some of these ideas in a more intuitive way.

2 Algebraic and Transcendental Numbers

Let α be a real number. We say that α is **rational** if there are integers p, q, with $q > 0$, such that $\alpha = p/q$. We say that α is **algebraic** if there is a polynomial $P(x)$ with integer coefficients such that $P(\alpha) = 0$. Finally we say that α is **transcendental** if it is not algebraic. For example, $\frac{22}{7}$ and 3.14159 are rational numbers, $\sqrt{2}$ is algebraic but not rational, and π is transcendental. Note also that every rational number is algebraic; can you see why?

To see that $\sqrt{2}$ is algebraic we can observe that $P(\sqrt{2}) = 0$ for the polynomial $P(x) = x^2 - 2$. The proof that π is transcendental is very tricky and is something that one would typically come across in a second-year mathematics course in Number Theory or Galois Theory.

The following property of rational numbers is useful.

Proposition 1. Let α be a rational number. Then there exists a constant $C > 0$ such that every rational number a/b different from α satisfies

$$\left|\alpha - \tfrac{a}{b}\right| \geq \tfrac{C}{b}\,.$$

Proof. Since α is assumed to be rational, we have $\alpha = p/q$ for some integers p, q with $q > 0$. Assume that $\alpha \neq a/b$. Then

$$\begin{aligned} \left|\alpha - \tfrac{a}{b}\right| &= \left|\tfrac{p}{q} - \tfrac{a}{b}\right| \\ &= \left|\tfrac{bp-aq}{bq}\right| \\ &\geq \tfrac{1}{bq}\,. \end{aligned}$$

In the last line we used the fact that $|bp - aq|$, being a strictly positive integer, is at least 1. To conclude, observe that by the above derivation, if $C = 1/q$ then $|\alpha - a/b| \geq C/b$ for all b. □

Proposition 1 can be used to show that $\sqrt{2}$ is irrational. Towards a contradiction, imagine that $\sqrt{2}$ were rational. Then by Proposition 1 there would exist C such that for all rational numbers $a/b \neq \sqrt{2}$ we have $|\sqrt{2} - a/b| > C/b$. But no matter how small C is, since $0 < \sqrt{2} - 1 < 1$, we can find $n > 0$ such that $0 < (\sqrt{2} - 1)^n < C$. Then, by the Binomial Theorem, we can expand the product $(\sqrt{2} - 1)^n$ in the form $b\sqrt{2} - a$ for some integers a, b. Then we have $|b\sqrt{2} - a| < C$ and hence $|\sqrt{2} - a/b| < C/b$. We have arrived at a contradiction, and hence the original assumption that $\sqrt{2}$ is rational must be wrong. We conclude that $\sqrt{2}$ is irrational.

The following property of algebraic numbers, due to Joseph Liouville in 1844, can be used to show that certain numbers are not algebraic. (**Exercise.** Explain how Theorem 2 generalises Proposition 1.) I would suggest to skip the proof at first and rather understand how the result is used.

Theorem 2. Let α be an irrational number such that $P(\alpha) = 0$ for some polynomial $P(x) = a_0 + a_1 x + \cdots + a_d x^d$ with integer coefficients. Then there is a constant C such that if a, b are integers with $b > 0$, then

$$\left|\alpha - \frac{a}{b}\right| \geq \frac{C}{b^d}\,. \tag{1}$$

Proof. Let M be the maximum value of the derivative $P'(x)$ as x ranges over the interval $[\alpha-1, \alpha+1]$ and let $\alpha_1, \ldots, \alpha_m$ be a list of the distinct roots of P other than α. Choose a positive constant C such that

$$C < \min\{1, \frac{1}{M}, |\alpha - \alpha_1|, \ldots, |\alpha - \alpha_m|\}\,.$$

Let a and $b > 0$ be integers. If $|\alpha - a/b| \geq C$ then we certainly have $|\alpha - a/b| \geq \frac{C}{b^d}$. On other hand, if $|\alpha - a/b| < C$ then

$$\left|\alpha - \frac{a}{b}\right| < \min\{1, \frac{1}{M}, |\alpha - \alpha_1|, \ldots, |\alpha - \alpha_m|\}$$

and hence

$$\frac{a}{b} \in [\alpha - 1, \alpha + 1] \text{ and } \frac{a}{b} \notin \{\alpha_1, \ldots, \alpha_m\}\,.$$

Since $\frac{a}{b} \in [\alpha - 1, \alpha + 1]$ we have

$$\begin{aligned} |P(a/b)| &= |P(a/b) - P(\alpha)| \\ &\leq M|a/b - \alpha| \quad \text{(by the Mean-Value theorem.)} \end{aligned}$$

Since $\frac{a}{b} \notin \{\alpha, \alpha_1, \ldots, \alpha_m\}$ we have than $b^d P(a/b)$ is a non-zero integer and so $|P(a/b)| \geq 1/b^d$. Combining the previous two inequalities we have $M|a/b - \alpha| \geq |P(a/b)| \geq 1/b^d$ and hence

$$|a/b - \alpha| \geq \frac{1}{Mb^d} > \frac{C}{b^d}\,.$$

□

Let's see how Theorem 2 can be used to show the transcendence of the number

$$\alpha := \sum_{k=1}^{\infty} 10^{-k!}.$$

(**Exercise:** describe the decimal expansion of α.) Now for each n we have

$$\sum_{k=0}^{n} 10^{-k!} = \frac{10^{n!-1} + 10^{n!-2} + \cdots + 1}{10^{n!}} = \frac{a}{b},$$

where $a = 10^{n!-1} + 10^{n!-2} + \cdots + 1$ and $b = 10^{n!}$ are integers. We thus have

$$\begin{aligned} |\alpha - a/b| &= \sum_{k=n+1}^{\infty} \frac{1}{10^{k!}} \\ &\leq \frac{1}{10^{(n+1)!}} \sum_{k=0}^{\infty} \frac{1}{10^k} \\ &= \frac{10}{9} \frac{1}{10^{(n+1)!}} \\ &= \frac{10}{9} \frac{1}{b^{n+1}}. \end{aligned}$$

Since n can be chosen as large as we like, there cannot exist C and d such that (1) holds for all integers a, b. We conclude that α is not algebraic.

3 Computable Numbers

We say that a real number α is **computable** if its decimal expansion can be computed by a Turing machine in the sense that, given input $n \in \mathbb{N}$, the Turing machine outputs the first n digits in the decimal expansion of α. A nice introduction to Turing machines and how they work can be found at `https://isaaccomputerscience.org/concepts/dsa_toc_turing_machines`.

The following are two examples of computable numbers:

$$\begin{aligned} \sqrt{2} &= 1.41421356237309504880168872420969807856 9\cdots \\ \pi &= 3.14159265358979323846264338327950288419 7\cdots \end{aligned}$$

Roughly speaking, these numbers are computable because one can approximate them to any desired accuracy. For instance, we can approximate $\sqrt{2}$ using Newton's method as follows. Consider the sequence $(x_n)_{n=0}^{\infty}$ with $x_0 = 2$ and

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{2}{x_n}\right).$$

This sequence converges to $\sqrt{2}$ as n goes to infinity. Indeed, with a bit of algebra (**Exercise:** try it!) we can derive the formula

$$x_{n+1} - \sqrt{2} = \frac{1}{2x_n}(x_n - \sqrt{2})^2,$$

which shows that the error in the next step is less than the square of the error in the current step. Since the initial error $|x_0 - \sqrt{2}|$ is less than $1/2$, the error after n steps is at most 2^{-2^n}, which is very small indeed.

Over the past several hundred years many formulas have been given that allow us to compute approximations of π. A simple such example (with rather slow convergence) is the Gregory-Leibniz series

$$\pi = 4\sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} = 4\left(\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots\right).$$

By truncating this infinite sum one can compute arbitrarily many digits of π.

We are glossing over a slightly subtle issue here. Approximating a given number α to within error 10^{-n} is *not* the same as computing the first n digits of its decimal expansion. For example, suppose $\alpha = 0.999$. Then $r := 1.000$ is an approximation of α to within error 10^{-3} but α and r disagree on the first digit of their decimal expansion. However, if we know that the decimal expansion of α does not contain "very long" consecutive sequences of 0's and 9's then we can indeed compute as many digits of α as we like by finding an approximation with suitably small error. For example, one can use Theorem 2 to show that for any algebraic number α there exists a constant $\varepsilon < 1$ such that the first n digits in the decimal expansion α does not contain a string of consecutive 0's or consecutive 9's of length more than εn.

We are now going to introduce a special class of computable numbers. A real number α is said to be **linear-time computable** if there is a Turing machine that on input n outputs the first n digits of the decimal expansion of α and does so in time at most Cn for some constant C.

The following was conjectured to be true by Juris Hartmanis and Richard Stearns in 1965.

Conjecture 3. If a real number α is linear-time computable then it is either rational or transcendental.

The conjecture remains open to the present day: it has neither been proved or refuted.

The Hartmanis-Stearns conjecture can be equivalently reformulated as saying that an irrational algebraic number (such as $\sqrt{2}$) cannot be linear-time computable. The conjecture has numerous consequences. For example, we have:

Theorem 4. If the Hartmanis-Stearns conjecture is true then there is no algorithm to multiply two n-bit numbers that runs in time linear in n.

One can prove Theorem 4 by arguing that a linear-time multiplication algorithm could be used a sub-routine by a Turing machine that can compute the first n digits in the decimal expansion of $\sqrt{2}$ in time linear in n by computing a suitably close numerical approximation to $\sqrt{2}$.

4 Automatic Sequences

In this section we describe a weak version of the Hartmanis-Stearns conjecture that is known to be true, i.e., the weak the version will be a theorem rather than a conjecture. This version involves a very restricted kind of Turing machine, called a **finite-state automaton**, which we introduce immediately below.

Let $k \geq 2$ be a natural number. A **k-automaton** is a simple type of computing machine that consists of the following components:

- a finite set Q is of **states**,
- an finite $\Sigma_k = \{0, 1 \ldots, k-1\}$ of **input symbols**,
- a **transition function** $\delta : Q \times \Sigma_k \to Q$,
- an **initial state** $q_0 \in Q$,
- a finite set Δ of **output symbols**,
- an **output function** $\tau : Q \to \Delta$.

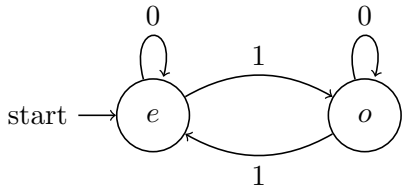
Here is how such an automaton works. It starts its computation in its initial state q_0. Given input a length-n sequence $w = w_1 w_2 \cdots w_n$ of input symbols, the automaton processes w left-to-right in n steps, moving through a sequence of states $q_0, q_1, \ldots, q_n$. In the i-th step it reads input letter w_i and moves from its current state q_{i-1} to a new state $q_i = \delta(q_{i-1}, w_i)$. When it has finished processing its input, the automaton outputs the symbol $\tau(q_n)$.

An infinite sequence $(a_n)_{n=0}^{\infty}$ taking values in Δ is said to be **k-automatic** if there is an k-automaton such that for all $n \in \mathbb{N}$, if the automaton is given as input the base-k representation of n, starting from the most significant digit, then it outputs a_n.

The **Thue-Morse** sequence $(a_n)_{n=0}^{\infty}$ is an example of a 2-automatic sequence. This is the sequence

$$\boldsymbol{a} = eooeoeeooeeoeooeoeeoeooeeooeoeeo\cdots$$

where $a_n = e$ if the number of 1's in the binary expansion of n is even and $a_n = o$ if the number of 1's in the binary expansion of n is odd. The automaton that computes this sequence is shown below: the labelled arrows describe the transition function and the labels on the states describe the output function.



For example, the 9th element of the Thue-Morse sequence is e since, inputting 1001 (the binary representation of nine) to the automaton, one ends in a state with label e.

Another example of an automatic sequence is the **paper folding sequence**. Take a rectangular sheet of paper and repeatedly fold it in half lengthways, right half over left. On unfolding the sheet of paper we see the creases as a sequence of valleys (v) and ridges (r). After five folds we successively see

$$v \qquad vvr \qquad vvrvvrr \qquad vvrvvrrvvvrrvrr$$
$$vvrvvrrvvvrrvrrvvvrvvrrrvvrrvrr$$

In each step we obtain the next sequence by taking the current sequence, adding a middle v, and then following with another copy of the current sequence but with v and r interchanged. The infinite sequence obtained in the limit is called the paper folding sequence. An alternative recursive

definition of the paper folding sequence is that it is the sequence $(a_n)_{n=0}^{\infty}$ such that for all $m \geq 0$ we have

$$a_{4m} = v \qquad a_{4m+2} = r \qquad a_{2m+1} = a_m \, .$$

The paper folding sequence is 2-automatic: it can be computed by an automaton with four states, input alphabet $\{0, 1\}$, and state labels v and r. I will describe the automaton in the lecture, but you might like to think about it in the meantime. (**Hint**: try and implement the recursive definition of the sequence. The automaton will need to "remember" a finite amount of information in its states, such as the last few bits of the input word.)

It turns out that if the decimal expansion of a number α is k-automatic then α is also linear-time computable by a Turing machine. The rough idea is that a linear-time Turing machine can simulate an automaton, which is a much simpler type of computing device. We finish with the promised weak version of the Hartmanis-Stearns conjecture.

Theorem 5. The decimal expansion of an irrational algebraic number α is not an automatic sequence.

Theorem 5 was proven by Adamczewski, Bugeaud, and Luca in 2004. The proof exploits the highly repetitive. structure of automatic sequences. This combinatorial feature is used together with a very sophisticated modern number-theoretical development of Theorem 2, called the Subspace Theorem.