Turing's Mathematical Work

P.D. Welch*

2010 Mathematics Subject Classification. Primary 01-A70; Secondary 03-02.
Keywords. Turing, computable numbers, ordinal logics.

Abstract. We sketch a brief outline of the mathematical, and in particular the logical, achievements of Turing in this, his centenary year.

1. Introduction

This is the centenary year of Alan Mathison Turing's birth: there have been many celebrations of the life and work of the man, with a veritable accumulation point around his birthday, June 23rd. It would have been impossible 20 years ago to imagine this year's stream of events, a considerable portion of which is not restricted to academic circles, but is in the very public eye: perhaps the arc of his life strikes a particular chord as someone emblematic of his country's history, his mileu, of a time past. As is appropriate for a Proceedings of this type, in this review we intend to take stock of his purely mathematical contributions, and put aside his war-time coding work, and the post-1945 work on the development of computers, and of morphogenesis on which he was working when he died. In this we have made a somewhat personal choice of his papers. This includes some of his unpublished work. His major contributions are in mathematical logic and I concentrate largely on explications of his two main papers there. However, within logic, there are a number of papers (and unpublished work) on type theory that are perhaps a bit too specialised or too dated for this account, so we have simply decided to omit any discussion of them. For a full list of his papers the reader should of course consult the mathematical volumes of the *Collected Works* [17] and [18] as well as the further volumes on Computation and Morphogenesis for his work there. This account is, for the main, chronological.

His mathematical upbringing was in a conventional English public school. Sherborne College where he was sent as a boarder (as his parents lived abroad - a fate of many children of that class in the Britain of that era) appears not to have exerted a great influence on him scientifically. He seems to have educated himself in many respects: he evidenced a lively curiosity in all things scientific, in matters biological, mechanical and physical from a young age. At home he would make up chemistry experiments, and there is a charming drawing by his mother of Alan playing hockey for his team - except that he isn't: the teams are engaged on the

*The author is grateful to the Isaac Newton Institute, Cambridge for a Visiting Research Fellowship during part of the preparation of this paper.

horizon, and Turing is bent over a flower examining it in the foreground. (He expressed interest in the puzzle of how flowers know how to grow - a question that stayed with him and resurfaced in his last work on morphogenesis.) He showed mathematical strength certainly, but without evidencing any Gauss-like precocity. However it was enough for him to win a scholarship to King's College, Cambridge in 1931 (although he failed to get one to Trinity, his first choice, and at that time the acme for the aspiring mathematician or scientist).

At Sherborne he had been interested in relativity, and the relatively new field of quantum mechanics. He had read Arthur Eddington's "*The Nature of the Physical World*". Eddington was an astrophysicist, who in Britain at least, was famous for having led the expedition to verify during the 1919 solar eclipse Einstein's predictions on the gravitational effects on the curvature of light. For the undergraduate Turing, the scientific luminaries at that time in Cambridge would have been Eddington himself, who would be an early influence, and G.H. Hardy. His undergraduate tutor at King's College was the group theorist Philip Hall. Besides Hardy he also read and absorbed von Neumann's "*Mathematische Grundlagen der Quantum Mechanik*", also a topic of enduring interest. During his undergraduate studies he is supposed to have given an improved proof of a theorem of Sierpiński, but what that was, or which theorem it was, has been lost in the mists of time.

2. The Central Limit Theorem

In 1933 he attended Eddington's lectures entitled "*The Distribution of Measurements in Scientific Experiments*." This must have sparked his curiosity more than usual since he distilled for himself a mathematical problem from Eddington's heuristic description, which he then proceeded to solve. This resulted in a theorem in fact it was the Central Limit Theorem, but to Turing this was unknown. It seems to be a pattern throughout his life, that he would endeavour to work things out for himself, preferably from first principles. For a young mathematician it is perhaps excusable not to be conversant with the relevant literature, but this tendency of working from scratch seems to have stayed with him.

The version of the Central Limit Theorem he proved had been discovered 12 years earlier by the Finnish mathematician Jarl Lindberg [10]. S. Zabell [21] gives an account of the history of the Central Limit Theorem and a full discussion of Turing's proof and its context. De Moivre's original formulation had been in terms of expressing the probability of success, S_n, after n trials from an infinite sequence $X_1, X_2, \ldots, X_n, \ldots$ of random variables, and its approximation to the Gaussian Error function.

Turing developed a condition for convergence that follows from Lindebergh's convergence condition. Lindbergh's condition was later (1935) shown by Feller and Levy to be necessary. Feller also discovered a subsequence phenomenon if his condition failed. Turing anticipated this by also demonstrating that a subsequence of the X_i would contribute a set of values converging to the Gaussian limit. He also proved a special case of the later (1936) theorem of Cramér: *If X and Y*

are independent, and $X + Y$ is Gaussian then X and Y are Gaussian. Turing showed just the special (and simpler) case that if it is *assumed* that additionally X is Gaussian then Y can be deduced to be Gaussian. The other insight stressed by Zabell, is that whereas earlier statements and proofs (and textbook versions) of the Central Limit Theorem were in terms of *densities* that needed stronger assumptions, he realised that the best results would be obtained by working with *distribution functions* rather than densities throughout - an insight used also by Lindebergh to get the optimal results.

Turing remained interested in statistical theories throughout his life. The article by I.J. Good in [17] gives an account of Turing's ideas concerning statistical evidence from the Bletchley Park years:

> Turing did not publish these war-time statistical ideas because, after the war, he was too busy working on the ground floor of computer science and artificial intelligence. I was impressed by the importance of his statistical ideas, for other applications, and developed and published some of them in various places. (Good, [8] p. 211)

Notwithstanding the lack of priority, Philip Hall encouraged him to write up this work as a Fellowship Dissertation for the King's College competition in 1934, which was done, being entitled *On the Gaussian Error Function*. This was accepted on 16 March 1935, Hall arguing that the rediscovery of a known theorem was a significant enough sign of Turing's strength (which he argued had not yet achieved its full potential). Turing thus won a 3 year Fellowship, renewable for another 3, with £300 *per annum* with room and board. He was 22 years old.

Probably under Hall's influence his first published paper was in group theory. This was a small and easily stated advance on a recent theorem of von Neumann's. The latter had defined two notions of 'left' and 'right' periodicity in [20] but had missed the fact that they are equivalent. Turing proved this, and it appeared as a two page paper in April 1935 [11]. By coincidence von Neumann arrived on a sabbatical visit from Princeton that month and proceeded to lecture on the subject; it is from this time that the two must have been acquainted.

3. *"On Computable Numbers"*

Probably more decisive to meeting von Neumann, was his contact with Max Newman. In Spring 1935 he went on a Part III course of Newman's on the Foundations of Mathematics. (Part III courses at Cambridge were, and are, of a level beyond the usual undergraduate curriculum but preparatory to undertaking a research career.) Newman was a topologist, and interested in the theory of sets. Newman attended Hilbert's lecture at the 1928 International Congress of Mathematicians where three strands of the latter's 'Program' were stated.

Hilbert had worked on Foundational matters for the previous decades and would continue to do so. His aim to obtain a secure foundations for mathematics by finding proofs of consistency of large parts (if not all) of mathematics by a process of systematic axiomatisation, and then showing that these axiomatisations were safe by providing finite consistency proofs, looked both reasonable and possible. By systematic effort Hilbert and his school had reduced the questions of the consistency of geometry to analysis. There seemed reasonable hope that genuinely finitary methods of proof could render arithmetic provably consistent within finite arithmetical means. Hilbert's program might be summarised as tripartite.

• (I Completeness) The question, or rather Hilbert's belief, that mathematics was *complete*: that is, given any properly formulated mathematical proposition P, either a proof of P could be found, or a disproof.

• (II Consistency) The question of *consistency*: given a set of axioms for, say, arithmetic, such as the Dedekind-Peano axioms, PA, could it be shown that no proof of a contradiction can possibly arise? Hilbert stringently wanted a proof of consistency that was finitary, that made no appeal to infinite objects or methods.

• (III Decidability - the *Entscheidungsproblem*) Could there be a finitary process or algorithm that would *decide* for any properly formulated proposition P whether it was derivable from axioms or not?

Of course the main interest was consistency, but there was both hope (discernible from some of the writings of the Göttingen group) that there was a positive solution to the *Entscheidungsproblem.* However, as is well known, Gödel's Incompleteness Theorems block Hilbert's program.

Theorem 3.1. (Gödel-Rosser First Incompleteness Theorem - 1931) *For any theory T containing a moderate amount of arithmetical strength, with T having an effectively given list of axioms, then: if T is consistent, then it is incomplete, that is for some proposition neither $T \vdash P$ nor $T \vdash \neg P$.*

The theorem is, deliberately, written out in a semi-modern form. Here, it suffices that T contain the Dedekind-Peano axioms, PA, to qualify as having a 'moderate amount of arithmetical strength'. The axioms of PA can be written out as an 'effectively given' list, since although the axioms of PA include an infinite list of instances of the Induction Axiom, we may write out an effective prescription for listing them. Hence PA satisfies the theorem's hypothesis. Gödel had used a version of the system of *Principia Mathematica* of Russell and Whitehead but was explicit in saying that the theorem had a wide applicability to sufficiently strong "formal systems" (although without being able to specify completely what that meant). This immediately established that PA is incomplete, as is *any* theory containing the arithmetic of PA. This destroys any hope for the full resolution of Hilbert's program that he had hoped for. In a few months there was more to come:

Theorem 3.2. (Gödel's Second Incompleteness Theorem - 1931) *For any consistent T as above, containing the axioms of* PA*, the statement that '*T *is consistent' (when formalised as '*Con_T*') is an example of such an unprovable sentence.*

Symbolically:

$$T \nvdash \mathrm{Con}_T$$

The First Theorem thus demonstrated the incompleteness of any such formal system, and the Second the impossibility of demonstrating the consistency of the system by the means of formal proof within that system. The first two of Hilbert's questions were thus negatively answered. What was left open by this was the *Entscheidungsproblem*. That there might be some effective or finitary process is not ruled out by the Incompleteness Theorems. But what could such a process be like? How could one *prove* something about a putative system that was not precisely described, and certainly not *mathematically* formulated?

Church and the λ-calculus

One attempt at resolving this final issue was the system of functional equations called the "λ-calculus" of Alonzo Church. This gave a strict, but rather forbidding, formalism for writing out terms defining a class of functions from base functions and a generalised recursion or induction scheme. Church had only established that the simple number successor function was "λ-definable", when his future PhD student Stephen Cole Kleene arrived in 1931; by 1934 Kleene had shown that all the usual number theoretic functions were also λ-definable. They used the term "effectively calculable" for the class of functions that could be computed in the informal sense of effective procedure or algorithm alluded to above.

Church ventured that the notion of λ-definability should be taken to coincide with "effectively calculable".

Church's Thesis (1934 - First version, unpublished) ***The effectively calculable functions coincide with the λ-definable functions.***

At first Kleene tried to refute this by a diagonalisation argument along the lines of Cantor's proof of the uncountablility of the real numbers. He failed in this but instead produced a theorem: the *Recursion Theorem.* Gödel's view of the suggestion contained in the thesis when Church presented it to him, was that it was "thoroughly unsatisfactory."

Gödel meanwhile had formulated an expanded notion of primitive recursive function that he had used in his Incompleteness papers; these became known as the *Herbrand-Gödel general recursive functions*. He lectured on these in 1934 whilst visiting the IAS, Princeton. Church and Kleene were in the audience, and seem to have decided to switch to the perhaps more mathematically appealing general recursive functions.

By 1935 Church could show that there was no λ-formula "$A \,\mathrm{conv}\, B$" iff the λ-terms A and B were convertible to each other within the λ-calculus. Moreover, mostly by the work of Kleene, they could show the λ-definable functions were co-extensive with the general recursive functions. Putting this "non-λ-definable-conversion" property together with this last fact, there was therefore a problem which, when coded in number theory, could not be solved using general recursive functions. This was published by Church [6]. Another thesis was formulated:

Church's Thesis (1936 - second version) ***The effectively calculable functions coincide with the [H-G] general recursive functions.***

Gödel still indicated at the time that the issue was unresolved, and that he was unsure that the general recursive functions captured all informally calculable functions.

"On Computable Numbers"

Newman and Turing were unaware of these developments in Princeton. Turing's classic paper's first subject is ostensibly 'Computable Numbers' and is said to be only "with an application to the *Entscheidungsproblem*". He starts by restricting his domain of interest to the natural numbers, although he says it is almost as easy to deal with computable functions of computable real numbers, but he will deal with integers as being the 'least cumbrous.' He briefly initiates the discussion with calling computable numbers those 'calculable by finite means.'

In the first Section he compares a man computing a real number to a machine with a finite number of states or 'm-configurations' $q_1, \ldots, q_R$. The machine is supplied with a 'tape' divided into cells capable of containing a single symbol from a finite alphabet. The machine is regarded as scanning, and being aware of, only the single symbol in the cell being viewed at any moment in time. The possible behaviour of the machine is determined only by the current state q_n and the current scanned symbol S_r which make up the current configuration of the machine. The machine may operate on the scanned square by erasing the scanned symbol or writing a symbol. It may move one square along the tape to the left, or to the right. It may also change its m-configuration.

He says that some of the symbols written will represent the decimal expansion of the real number being computed, and others (subject to erasure) will be for scratch work. He thus envisages the machine continuously producing output, rather than halting at some stage. It is his contention that "these operations include all those which are used in the computation of a number." His intentions are often confused with statements such as 'Turing viewed any machine calculation as reducible to one on a Turing machine' or some thesis of this form. Or that he had 'distilled the essence of machine computability down to that of a Turing machine.' He explicitly warns us that no "real justification will be given for these definitions until Section 9."

In Section 2 he goes on to develop a theory of his "automatic" or a-machines giving and discussing some definitions. He also states: *"For some purposes we may use machines whose motion is only partly determined. When such a machine reaches one of these ambiguous configuations, it cannot go on until some arbitrary choice has been made ..."*

Having thus in two sentences prefigured the notion of what we now call a *non-deterministic Turing machine* he says that he will stick in the current paper only to a-machines, and will drop the 'a'. He remarks that such a non-deterministic machine 'could be used to deal with axiomatic systems.' (He is probably thinking here of the choices that need to be made when developing a proof line-by-line in a formal system.) The succeeding sections develop the theory of the machines, the theory of a *"universal machine"* is explicitly described, as is in particular the conception of program as input or stored data; further, the mathematical argument using Cantor's diagonalisation technique, to show the impossibility of determining by a machine, whether a machine program was 'circular' (that is, writing only finitely many output symbols) or not. (Thus, as he does not consider a complete computation as a halted one, he instead considers first the problem of whether one can determine a looping behaviour.)

Section 9, "The extent of the computable numbers", is in some ways the heart of the paper, in particular for later discussions of the so-called 'Turing' or 'Church-Turing' theses. It is possibly of a unique character for a paper in a purely mathematical journal of that date (although perhaps reminiscent of Cantor's discussions on the nature of infinite sets in *Mathematische Annalen*). He admits that any argument that any calculable number (by a human) is "computable" (*i.e.* in his machine sense) is bound to hang on intuition and so be mathematically somewhat unsatisfactory. He argues that the basis of the machine's construction earlier in the paper is grounded on an analysis, which he then proceeds to give, of what a human computor does when calculating. This is done by appealing to the obvious finiteness conditions of human capabilities: the possibilities of surveying the writing paper, observing symbols together with their writing and erasing.

It is important to see that this analysis should be taken *prior to* the machine's description. (Indeed one can imagine the paper re-ordered with this section placed at the start.) He had asked:

> *"What are the possible* processes *which can be carried out in computing a real number"* [My emphasis].

It is as if the difference between the Princeton approach and Turing's is that the former appeared to be concentrating on discovering a definition whose extension covered in one blow the notion of effectively calculable, where as Turing concentrated on process, the very act of calculating.

According to Gandy, [7], Turing has in this section in fact proved a *theorem* albeit one with unusual subject matter. What has been achieved is a complete analysis of human computation in terms of finiteness of the human acts of calculation broken-down into discrete, simple, and locally determined steps. Hence:

Turing's Thesis: Anything that is humanly calculable is computable by a Turing machine.

(i) Turing provides a philosophical paradigm of analysis when defining "effectively calculable": a vague intuitive notion is given a unique meaning which moreover can be stated with complete precision.

(ii) He also makes possible a completely precise understanding of what is a 'formal system' thereby making an exact statement of Gödel's results possible (see the quotation below). He claims to have a machine that will enumerate the theorems of predicate calculus. This also makes possible a correct formulation of Hilbert's 10'th problem. It is important to note in this regard that Turing thus makes expressions along the lines of "such and such a proposition is undecidable" have mathematical content.

(iii) In the final 4 pages he gives his solution to the *Entscheidungsproblem*. He proves that there is no machine that will decide of any formula φ of the predicate calculus whether it is derivable or not.

He was 23. His mentor and teacher Max Newman was astonished, and at first reacted with disbelief. He had achieved what the combined mental resources of Hilbert's Göttingen school and Princeton had not, and in the most straightforward, direct, even simple manner. Within 14 months of starting to attend the Foundations of Mathematics course he had solved the last general open problem associated with Hilbert's program.

However this triumph was then tempered by the arrival of Church's preprint of [5] which came just after Turing's proof was read by Newman. The latter however convinced the London Mathematical Society that the two approaches were sufficiently different to warrant publication; this was done in November 1936, with an appendix demonstrating that the the machine approach was co-extensional with the λ-definable functions, and with Church as referee. Gödel again:[1]

> "When I first published my paper about undecidable propositions the result could not be pronounced in this generality, because for the notions of mechanical procedure and of formal system no mathematically satisfactory definition had been given at that time. . . . The essential point is to define what a procedure is."
>
> "That this really is the correct definition of mechanical computability was established beyond any doubt by Turing."

4. Normal Numbers

Turing's unpublished *"A Note on Normal Numbers"* (in [17]) dates presumably from about 1936 (the manuscript is on the reverse of some pages of a proof copy of the *"On Computable Numbers"* manuscript). The notion of *normal number* is due to Borel who showed, measure theoretically and hence highly non-constructively, that almost all real numbers are *normal*. A number, say in $(0, 1)$ is called normal if *in every base*, every block of digits of the same length occurs with the same limit frequency. Thus, in a binary expansion 0 and 1 must each occur half the number of times, each of the blocks 00, 01, 10 and 11 one quarter of the times and so on. As Turing's typescript starts out:

> Although it is known that almost all numbers are normal no example of a normal number has been given. I propose to show how normal

[1]There are several approving quotes from Gödel; this is taken from an unpublished (and ungiven) Lecture in the *Nachlass* Gödel, Collected Works, Vol III, p166-168.

numbers may be constructed and to prove that almost all numbers are normal constructively.

Becher [1] gives an account of the typescript and accompanying manuscript notes. In the latter Turing gives a partial example due to his friend David Champernowne in the explicit base 10 only: 0.1234567891011121314... by simply stringing together all base 10 numerals one after the other (so a 'semi-normal' number). So this example had a simple description. Turing asserts that his solution, although constructive - it makes use of his own new theory of computable reals - does not give what he calls a 'convenient' solution, such as exemplified by Champernowne's number. Nevertheless it is perfectly constructive, and indeed Turing uses this word in his paper rather than 'computable' which would have been perfectly appropriate. Both Sierpinski and Lebesgue gave constructions of normal numbers, but these proofs are not finitary and so not computable or constructive in the modern sense, but Becher speculates that perhaps these previous proofs put Turing off from publishing his own note.

Theorem 4.1 (Turing). *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 $E_{c(k,\,n)} = (a_1, b_1) \cup \cdots \cup (a_{1m}, b_m)$ denotes the finite union of the intervals whose rational endpoints are the pairs given by $c(k,n)$, then $E_{c(k,\,n)} \subset E_{c(k,\,n\,-\,1)}$ and the measure of $E_{c(k,\,n)}$ is greater than $1 - \frac{1}{k}$. Further, for each k, $E(k) = \bigcap_n E_{c(k,\,n)}$ has measure $1 - \frac{1}{k}$ and consists entirely of normal numbers.*

Becher *et al.* ([2]) have reconstructed the proof of the following second theorem (see her discussion in [1] of this in relation to the introductory note of J.L. Britton in [17] which had questioned the veracity of this theorem). It produces explicitly computable normal numbers:

Theorem 4.2 (Turing). *There is a rule whereby given an integer k, and an infinite sequence θ of zeros and ones, we can find a normal number $\alpha(k,\theta) \in (0,1)$ and in such a way that for a fixed k these numbers form a set of measure at least $1 - \frac{2}{k}$, and so that the first n digits of θ determine $\alpha(k,\theta)$ to within 2^{-n}.*

In modern day terms, the 'rule' is a computable algorithm, and when the sequence θ is a computable one, then the output is a computable normal number. Becher points out that the time complexity of the algorithm needed to produce the n'th digit of $\alpha(k,\theta)$ is doubly exponential in n, and such appears to be the best to date. (They also note that an 'effectivized' version of Sierpinski's argument also gives a doubly exponential time algorithm.) There it is also remarked that the proof shows that *random numbers* (a later concept related to work of Martin-Löf, and others) are all normal.

5. Princeton Years

Afte the triumph of the "On Computable Numbers" it was natural for Turing to visit Princeton which he did in 1937 but was somewhat dismayed to find only

Church and Kleene there. (He had naturally hoped to meet Gödel, but their paths were not to cross.) He published quite quickly two papers on group theory (described in a letter to Philip Hall - as 'small papers, just bits and pieces' - nevertheless they appeared in Compositio [12] and Annals of Mathematics [13]). He had first asked von Neumann for a problem, and von Neumann passed on one from Ulam concerning the possibility of approximating continuous groups with finite ones which Turing soon answered negatively in [13].

Let G be a multiplicative group with a product $\cdot$ and a metric d. Let $\varepsilon > 0$ be fixed. A finite group H_ε with a product $\circ$ is said to be an *ε-approximation to G* if $H_\varepsilon \subseteq G$ and (i) every $x \in G$ is within distance ε of some $h \in H_\varepsilon$ (ii) $a, b \in H_\varepsilon \Rightarrow d(a \circ b, a \cdot b) < \varepsilon$. G itself is said to be *approximable* if it has an ε-approximation for every $\varepsilon > 0$. Turing then proved two theorems:

Theorem 5.1 (Turing). *Let G be an approximable group with a faithful representation over complex matrices of degree n. Then may be approximated by finite groups with faithful representations of the same dgeree n.*

Theorem 5.2 (Turing). *An approximable Lie group is compact and abelian.*

The Compositio paper (which Turing had stated in the letter as something 'Baer thinks is worth publishing') concerns the problem of determining the extensions of a given group G by a given group H inducing given classes of automorphisms.

He stayed on in Princeton on a Procter Fellowship (of these there were three, one each for candidates from Cambridge, Oxford and the Coll'ege de France). He decided to work towards a Ph.D. under Church. He still had a King's Fellowship, and thus a Ph.D. would not have been of great use to him in the Cambridge of that day. He completed his thesis in two years (even whilst grumbling about Church's "suggestions which resulted in the thesis being expanded to an appalling length" - it is 106 pages.) To illustrate the thesis problem, by an example (where we may think of T_0 as PA again). Set:

$$T_1 : T_0 + \mathrm{Con}(T_0)$$

where "$\mathrm{Con}(T_0)$" is some expression arising from the Incompleteness Theorems expressing that "T_0 is a consistent system"; as $\mathrm{Con}(T_0)$ is not provable from T_0, this is a deductively stronger theory; continuing, define:

$$T_{k+1} : T_k + \mathrm{Con}(T_k) \text{ for } k < \omega, \text{ and then: } \quad T_\omega = \bigcup_{k<\omega} T_k.$$

$$T_{\omega+1} = T_\omega + \mathrm{Con}(T_\omega) \ \textit{etc.}$$

We thus obtain a transfinite hierarchy of theories. What can one in general prove from a theory in this sequence? Turing called these theories "Logics" and was thus investigating the question as to what extent such a sequence could be 'complete':

Question: Can it be that for any problem A there might be an ordinal α so that T_α proves A or $\neg A$?

Actually he was aiming at a more restricted question, namely what he called *number theoretic problems* which are those that can be expressed in an '$\forall\exists$' form (the twin primes conjecture comes to mind). He does not clarify why he alights on this particular form of the question. To formally write down in the language of PA a sentence that says "Con(PA)" one really needs a formula $\varphi_0(v_0)$ that defines for us the set of gödel code numbers n of instances of the axiom set $T_0 = \mathrm{PA}$. There are infinitely many such formulae but we choose one which is both simple (it is Σ_1, meaning definable using a single existential quantifier), and *canonical* in that it simply defines the the axioms numbers in a straightforward manner. Assuming we have a φ_0, we then may set $\varphi_{k+1}(\bar{n}) \longleftrightarrow \varphi_k(\bar{n}) \vee \mathrm{Con}(\varphi_k)$ where $\mathrm{Con}(\varphi_k)$ expresses in a Gödelian fashion the consistency of the axiom set defined by φ_k. But what to do at stage ω?

Turing solved this, and devised a method for assigning sets of sentences, so theories, to all constructive (also now called *recursive* or *computable*) ordinals by the means of *notations*. In essence a notation for an ordinal is merely some name for it, but a system of notations (which Turing used) was invented by Kleene using the λ-calculus. Nowadays we also use the idea of being able to name the ordinal α by the natural number index e of a computable function $\{e\}$ which computes the characteristic function of a wellorder of $\mathbb{N}$ of order type α.

This essentially yields a tree order on the set of notations $\mathcal{O} \subset \mathbb{N}$ with infinite branching at all and only constructive ordinal limit points, with $n <_{\mathcal{O}} m \leftrightarrow |n| < |m|$, where $|\cdot|$ is the ordinal rank function (defined by transfinite recursion along $<_{\mathcal{O}}$) satisfying:

$$|0| = 0; \quad |2^a| = |a| + 1; \quad |3^e| = lim_{n\to\infty}|\{e\}(n)|.$$

However $\mathcal{O}$ is a *co-analytic* set of integers, and is thus highly complex. Let $suc(a) =_{df} 2^a$, let $lim(e) =_{df} 3^e$.

Definition 5.3. A *progression based on a theory* T is a primitive recursive mapping $n \longrightarrow \varphi_n$ where φ_n is an $\exists$ formula such that PA proves:

(i) $T_0 = T$; (ii) $\forall n \left(T_{\mathrm{suc}(n)} = T_n + \mathrm{Con}(\varphi_n)\right)$; (iii) $T_{\mathrm{lim}(n)} = \bigcup_m T_{\{n\}(m)}$.

Thus one attaches *in a uniform manner* formulae φ_a to define theories T_a to *every* $a \in \mathbb{N}$ of the form $suc(a), lim(a)$. However this does not tell us how to build progressions, the existence of which has to be justified through the use of the Recursion Theorem. An *explicit consistency sequence* is then defined to be the restriction of a progression to a path through $\mathcal{O}$. With these tools Turing proved a form of an enhanced Completeness Theorem.

Theorem 5.4 (Turing's Completeness Theorem). *For any progression there is a primitive recursive function $\psi \rightarrowtail b(\psi)$, so that for any true $\forall$ sentence of arithmetic, ψ, there is a $b = b(\psi) \in \mathcal{O}$ with $|b| = \omega + 1$, so that $T_b \vdash \psi$.*

Thus we may for any true ψ find a path through $\mathcal{O}$ of length $\omega + 1$,

$$T = T_0, T_1, \ldots, T_{\omega+1} = T_b$$

with the last proving ψ. At first glance this looks like magic: how does this work, and can we use it to discover more $\forall$-facts about the natural number system?

However there is a trick here: as Turing readily admits, what one does is construct for *any* $\forall$ sentence ψ an extension $T_{b(\psi)}$ proving ψ with $|b(\psi)| = \omega + 1$; Then *if* ψ *is true* we are able to show that $T_{b(\psi)}$ is a consistent extension in a proper consistency sequence (notice that conditional in the antecedent of the theorem's statement); however *if* ψ *is false* $T_{b(\psi)}$ turns out to be merely inconsistent, and so proves anything. In general it is harder to answer ?$b \in \mathcal{O}$? than the original $\forall$ question is, and so we have gained no new arithmetical knowledge. The outcome of the investigation is thus somewhat equivocal: we *can* say that some progressions of theories will produce truths of arithmetic, but we cannot determine which ones they will turn out to be. He seems to have regarded the results as somewhat disappointing.

There is a remarkable aside however. Almost as a throw-away comment he introduces what has come to be called a *relativised Turing Machine* or (as he called it) an *oracle machine*. This machine is allowed an instruction state that permits it to query of an 'oracle', considered perhaps as an infinite bit-stream of information, about the members of $B \subseteq \mathbb{N}$ written out on a separate tape, whether ?$n \in B$? An answer is received and computation continues. With this one can develop the idea of 'relative computability' - whether membership of m in set A can be determined from answers to finitely many membership questions about set B. This notion is central to modern computability theory. However Turing introduces the concept, (dubbing it an 'oracle' or *o*-machine) and uses it somewhat unnecessarily to prove the point that there are arithmetic problems that are not in his sense number theoretic problems. And then ignores it for the rest of the paper.

The paper, duly published in 1939, lay somewhat dormant until taken up by Spector and Feferman some 20 years later. Feferman did a far reaching analysis of the notion of general progressions, using not just formalized consistency statements as Turing had done, but also other forms, that roughly speaking ensured the preservation of truth.

6. Riemann Zeta Cogs

In 1937 Turing became increasingly interested in the Riemann Hypothesis. In March 1939 he submitted the paper [14] (although this did not appear until 1943). He intends that the calculations therein should be good for calculating 'mid-range' values of the ζ function (between $30 < t < 1000$). E.C. Titchmarsh in Oxford had calculated that the first 104 zeroes lay on the principal line, and in fact had used punched card machinery (used for astronomical predictions) to assist in the computations. Turing intended to do the same for the next few thousand, and he had noticed an analogue machine for computing waves and making tidal predictions at the University of Liverpool. Turing must have written to Titchmarsh for the latter replied on Dec 1st 1937, that he had also seen the Liverpool machine without thinking of this possible application. In March 1939 he applied to the Royal Society for a grant (Hardy and Titchmarsh were mentioned as referees). The machine would locate approximate zeros and the final calculations would be done by hand. He was awarded a 40 pound grant. Whereas the Liverpool machine used strings and pulleys to add a series of waves, the proposed machine would use meshing gears and cogs - which would have to be cut rather precisely to simulate rational approximations to real numbers. A blueprint was drawn up by his friend Donald MacPhail in July, and he made a start cutting the brass gears himself in the Engineering Department, whilst bemused visitors would find his room in King's littered with parts. However the machine was not to be, since on September 3rd Britain declared war on Germany and on the next day Turing set off for the Government Coding and Cypher School at Bletchley Park.

7. Manchester

After his war-time work Turing could have returned to Cambridge. His Fellowship had been kept open for him, and indeed extended for a further three years, and he was to remain a Non-Resident Fellow for the rest of his life. However he was interested in having his Universal Machine built and went to the National Physical Laboratory (a government run institution) with hopes of bringing this about. However frustrated by bureaucratic delays and with the invitation by Max Newman (now Professor at Manchester) to join the group there (already building a machine) he was appointed to a position there in May 1948. His work there was focussed on the nascent computers being built, both design, program writing and planning for their use. One suitable problem chosen by Newman was looking for Mersenne primes $2^p - 1$. They managed on the Mnchester Mark 1 machine to search through all $p \leq 353$ without discovering a further prime (the next was for $p = 521$ discovered by a computer search in 1951, but this was beyond their reach at the time). Subsequently Turing worked on zeroes of the Riemann Zeta function once more.

The 1953 paper *A Method for calculation of the Zeta function*, [16], according to Heath-Brown's assessment in [17], is more interesting than the earlier 1943 paper: whilst the former was surpassed by Titchmarsh's theorems this paper provides a means for assessing whether one has *all* the zeroes within a given interval under consideration. The paper gives an insight to what it must have been like to do the first example of machine computing applied to such a problem: the frustrations of wrestling with unreliable hardware, limited computer time, space and other bounding factors.

The paper (nicely analysed by Booker in [3]) is in two parts, the first dealing with the mathematics, and a shorter second part detailing the computations with a brief sketch of the Manchester University machine itself, finishing with an outline of the calculation method. Titchmarsh's method, derived from Riemann's, was to define a function $Z(t)$, that had the same zeroes t as the zeta function had for $\zeta(\frac{1}{2} + it)$. By then calculating in some other fashion $N(T)$ the number of non-real zeroes of ζ with imaginary part up to T, one has a comparison with the count of zeroes obtained from Z. $N(T)$ can be approximated by $M(T)$:

$$M(T) = \frac{T}{2\pi} \ln \left(\frac{T}{2\pi e} \right) + \frac{7}{8}.$$

Although the error $E(T) = N(T) - M(T)$ is aymptotically 0, it can be large for large T, which makes it useless for judging whether we have all the zeroes up to any given T. Littlewood had shown that $E(T)$ has average value close to 0 when T is large, so Turing had the idea of looking at $E(T)$ for a range of T rather than at single values. Hence any missing zeroes from the count would have $E(T)$ oscillating around a different integer value. To use this rigorously one needs to have explicit constants in Littlewood's theorem, and Turing derives these in the paper. Heath-Brown describes this as a "much easier and more elegant method than the technique used by Titchmarsh (which was rather hit or miss - the new method is fail-safe). It is the method adopted in all recent computations."

In Manchester Turing was interested in the unsolvability of word problems. In an unpublished note (not a finished manuscript - *"The Word Problem in Compact groups"* in the *Nachlass*) he tries to show that this is soluble. This must date to post-1948 as he references Tarski's RAND-project report on the latter's decision method for elementary algebra and geometry dating from that year. In 1953 Turing wrote his last academic research paper in pure mathematics: *"The word problem in semi-groups with cancellation"* ([15]). Given a set with an associative multiplication operation (a semi-group), obeying additionally the cancellation laws $ab = ac \to b = c$ and $ba = ca \to b = c$, can there be such, which has a presentation in terms of two finite sets of symbols and of equations, but so that the problem of deciding whether two 'words' W_0 and W_1 each made up of a string of symbol multiplicands are in fact equal, is unsolvable? In 1947 such a problem had been shown unsolvable by Post and Markov independently for pure semi-groups. In this paper Turing shows that here too the general problem is unsolvable. As [4] states, this argument was to be the basis of both Boone's and Novikov's (independent) proofs of the unsolvability of the word problem for groups.

Acknowledgements: Some segments of this article have appeared in the EMS *Newsletter* and thus our debts are accumulating: mention should be made again in particular that the events of Turing's life, are for the most part taken from Andrew Hodges's wonderful biography; for this article the accounts of various areas beyond our expertise are mentioned in the bibliography below. In particular from the forthcoming volume on *Turing: His work and Impact,* I made substantial use of the clear and illuminating articles of Veronica Becher, S. Zabell and Andy Booker to whom I'd like to express my gratitude. The commentary of D. Heath-Brown within, and of the editors of, the various volumes of the *Collected Works* have also been very useful.

References

[1] V. Becher. Turing's note on normal numbers. In S.B. Cooper and J. van Leeuwen, editors, *Alan Turing - His work and Impact.* Elsevier, 2012.

[2] V. Becher, S. Figueira, and R. Picchi. Turing's unpublished algorithm for normal numbers. *TCS*, 377:126–138, 2007.

[3] A. Booker. Turing and the primes. In S.B. Cooper and J. van Leeuwen, editors, *Alan Turing - His work and Impact*. Elsevier, 2012.

[4] W.W. Boone. An analysis of Turing's "The word problem in semi-groups with cancellation". *Annals of Mathematics*, 67(1):195–202, January 1956.

[5] A. Church. A note on the *Entscheidungsproblem*. *Journal of Symbolic Logic*, 1:40–41, Correction 101–102, 1936.

[6] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58, 1936.

[7] R.O. Gandy. The confluence of ideas in 1936. In Herken, editor, *The Universal Turing machine: a half century survey*, pages 55–111. Oxford University Press, Oxford, 1988.

[8] I.J. Good. A.M. Turing's Statistical work in World War II. In J.L. Britton, editor, *Alan Turing's Collected Works*. North Holland, Amsterdam, 1992.

[9] G.H. Hardy. Mathematical Proof. *Mind*, 38:1–25, 1929.

[10] J. Lindebergh. Eine neue Herleitung des Exponentialgesetzes in der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 15:211–225, 1922.

[11] A.M. Turing. Equivalence of left and right almost periodicity. *Journal of the London Mathematical Society*, 10, 1935.

[12] A.M. Turing. The extension of a group. *Composition Mathematicae*, 5:357–367, 1938.

[13] A.M. Turing. Finite approximations to Lie groups. *Annals of Mathematics*, 39(1):105–111, 1938.

[14] A.M. Turing. A Method for calculation of the Zeta function. *Proceedings of the London Mathematical Society*, 48(2):180–197, 1943.

[15] A.M. Turing. The word problem in semi-groups with cancellation. *Annals of Mathematics*, 52(2):491–505, September 1950.

[16] A.M. Turing. Some calculations of the Riemann Zeta function. *Proceedings of the London Mathematical Society*, 3((3)):99–117, 1953.

[17] A.M. Turing. Pure Mathematics. In J.L. Britton, editor, *Collected Works*. North-Holland, Amsterdam, 1992.

[18] A.M. Turing. Mathematical Logic. In R.O. Gandy and C.E.M. Yates, editors, *Collected Works*. North-Holland, Amsterdam, 2001.

[19] J. von Neumann. Zur Hilbertsche Beweistheorie. *Mathematische Zeitschrift*, 26:1–46, 1927.

[20] J. von Neumann. Almost periodic functions in a group. *Transactions of the American Mathematical Society*, 36:445–492, 1936.

[21] S. Zabell. Alan Turing and the central limit theorem. In S.B. Cooper and J. van Leeuwen, editors, *Alan Turing - His work and Impact*. Elsevier, 2012.

P.D. Welch, School of Mathematics, University of Bristol, Bristol, BS8 1TW, England;
Visiting Research Fellow, Isaac Newton Institute, Cambridge.

E-mail: p.welch@bristol.ac.uk