What Would Alan Turing Have Done After 1954?

Andrew Hodges

Wadham College, University of Oxford

Summary. Incomplete aspects of Turing's work are surveyed, with particular reference to his late interest in the foundations of quantum mechanics, and refuting the assertion that his work raised the prospect of constructing physical "oracle-machines."

Alan Turing died on 7 June 1954 at the age of 41. It is of course an unanswerable question as to what he would have done if he had lived. His life was full of surprises at every turn. But I shall use this counterfactual theme to survey some incomplete threads in his life and work, some of them under-appreciated. I shall also address recent mistaken claims that Turing anticipated the agenda of so-called "hypercomputing."

1 A Survey of Turing's Legacy in 1954

In his last year, Turing was exploring many avenues in his morphogenesis theory. The problem of explaining the Fibonacci patterns in plants was probably less tractable than he had at first hoped. But there were other directions in which his biological theory might have advanced if he had lived longer. He might well have pursued a connection with von Neumann's ideas for discrete self-organizing systems, usually considered as the foundation of "artificial life." He might have seized upon the decoding of DNA in 1953, which introduced discrete logic into biology. It is also notable that it was through numerical simulations of non-linear equations, made possible by the computer, that chaotic phenomena became accessible to investigation in the 1950s. Such numerical simulations were Turing's *forte* by 1954. It seems quite possible that he would have seen the nature of chaos rather quicker than other people did. So there was great scope for broadening his applied-mathematical interests.

But Turing had by no means abandoned pure mathematics. He had probably lost interest in mathematical logic for its own sake. But he might well have gone on to contribute to other decision problems within mathematics. In 1950 he had done work on decidability problems in *semigroups* [35] and then Turing described P. S. Novikov's new result on the undecidability of the "word problem" for *groups* in a semi-popular article appearing in 1954 [38].

He explained a word problem in terms of a problem in knot theory. This illustration itself pointed to another fascinating and growing area in post-war mathematics, and also reflected the more geometrical turn of his interests. The 1954 article went on to explain Gödel's theorem. This was perhaps the first popular article on the subject, which was not at all well known in those days. So this last paper also suggests another role that a longer-lived Alan Turing might have taken — a great communicator of mathematics and science to a wide audience. But it also suggests that Turing might have taken up, for instance, the outstanding question (Hilbert's Tenth Problem) of the solvability of Diophantine equations, not settled in the real world until 1970, Martin Davis having a prominent role in the story and being a distinguished expositor of it [12].

What about the future of his work in computer science? Immediately after Turing's death in 1954, his student and friend Robin Gandy wrote to Max Newman, Turing's colleague and patron, with an account [17] of what struck him as unfinished in Turing's work. Gandy wrote comments under eight different headings. Of these only one was on morphogenesis; only one of them, the sixth, was in computer science, and was as follows:

> I always hoped he would return one day to the practical problems of making a machine learn. There should be somewhere a copy of the report he wrote on this after his sabbatical year at Cambridge from the NPL.

We may well rejoice in the fact that the basis for Christof Teuscher's work, which has in turn brought about this *Festschrift* for Alan Turing's ninetieth birthday, is the practical exploration of the theory of networks in this report [34], entitled futuristically *Intelligent Machinery*.

However, it is worth noting Gandy's message that Turing had shown little interest in pursuing this work in practice. When he had the 1951 Manchester computer at his disposal, he had not used it to follow up his "learning" proposals. This was true also of his ideas for programming. It is very striking that he continued to write raw machine code for the Manchester machine, although he of all people knew that the machine itself could have been made to do the routine work. In 1946, years ahead of others, he had seen the potential of the stored program for interpreters, compilers and scripts [32]:

> The process of constructing instruction table should be very fascinating. There need be no real danger of it ever becoming a drudge, for any processes that are quite mechanical may be turned over to the machine itself.

In 1947 he explicitly recognized the general nature of programming languages [33]:

> ... one could communicate with these machines in any language provided it was an exact language, i.e. in principle one should be able to communicate in any symbolic logic, provided that the machine

were given instruction tables which would allow it to interpret that logical system.

In 1950 his M.Sc. student Audrey Bates worked on putting a small part of Church's lambda-calculus in a form where it could be mechanized by the Manchester computer [1]. This work could have led to LISP programming, which was also inspired by the lambda-calculus, but he never followed it up. The same is true of the work he did on program proofs in 1949; this was never taken up and had to wait for others in the 1960s.

The computer scientist John McCarthy would have invited Turing to Dartmouth College in 1956, for what is usually thought of as the conference that began Artificial Intelligence. What would Turing have said, if he had accepted such an invitation? He would have been living witness to the fact that Artificial Intelligence research had started well before 1956. The wartime origin was described in [20, e.g. pp. 210–214, 265, 291–294] with a deeper analysis in [21, 23]. Perhaps he would have advocated avoiding the separation of "top-down" from "bottom-up" research that was in fact to characterize AI research so strongly for the next thirty years. For Turing in 1948 and again in 1950 [34, 36] had described both approaches together, saying that both should be tried out. But he had made little effort to make such trials himself. Turing preferred making the first attack at a new idea and then leaving the details for others to work out. This was true of his programming theory, his bottom-up ideas on neural networks, and his top-down ideas on machine chess-playing. So it is by no means obvious that a longer life would have led him to continue with AI research.

There is, however, another arena where his knowledge of mathematical logic might have been brought into practical computer science to make a first attack on a new area: this is what we have known as complexity theory since the 1970s.

Practical time constraints on algorithmic solutions formed a vital aspect of Turing's wartime work. It seems quite possible that he was consulted by GCHQ after 1948 about the use of computers for large-scale problems, such as the famous Venona problem of Soviet messages which was the top Anglo-American priority in that period. If so, it is also possible that research in large-scale efficient computer-based searching and sorting would have brought him to complexity theory ideas.

Turing's wartime work mainly lay in probability theory and Bayesian statistics. Afterwards he left it to Jack Good to write up a civilian version of his theory, and he made no effort to pursue the parallel of his work with Shannon's information theory. But possibly he would one day have gone on to combine his knowledge of computation and probability: in particular he had left the concept of randomness oddly informal. He described machines with "random elements" but these were left to Shannon and others in 1956 work to define properly [13].

Looking further ahead, the ideas of Gregory Chaitin on randomness and computability give a picture of a field Turing might have opened — even if not necessarily agreeing with all Chaitin's views.

A minor feature of Turing's postwar work, but one that might have blossomed with longer life, is the application of computing methods in pure mathematics. His colleague Max Newman was very quick to exploit the Mersenne Prime problem to illustrate the power of computation, and discussed very advanced ideas at the inauguration of the Manchester computer [24] in the use of probabilistic methods in algebra and number theory. Probabilistic primality testing, as used in public-key cryptology today, might have been working much earlier in Turing's hands.

He might also have made powerful advances in cryptology itself. It is striking how he made general statements about this field, and we do not know where his thoughts were leading. In a 1936 letter [29] he reported to his mother from Princeton:

> I have just discovered a possible application of the kind of thing I am working on at present. It answers the question "What is the most general kind of code or cipher possible," and at the same time (rather naturally) enables me to construct a lot of particular and interesting codes.

This tantalizing statement, with its fascinating link between computability and cryptology, leaves us only wanting to know the answer Turing found to his question, and the identity of the particular and interesting codes. Possibly the latter were related to Turing's 1937–8 cryptological work, which was reported to me by Dr Malcolm McPhail in 1978 in the following terms (see [20, p. 138]):

> ... he would multiply the number corresponding to a specific message by a horrendously long but secret number and transmit the product. The length of the secret number was determined by the requirement that it should take 100 Germans working eight hours a day on desk calculators 100 years to discover the secret factor by routine search. Turing actually designed an electric multiplier ...

Again, we are left wondering what the scheme actually was (for multiplication is too simple), and what was his theory of its security. It is by no means clear what Turing was doing, and he may well have had many advanced ideas that were never published. In 1950 he divulged [36]:

> I have set up on the Manchester computer a small programme using only 1000 units of storage, whereby the machine supplied with one sixteen figure number replies with another ... I would defy anyone to learn from these values sufficient about the programme to be able to predict any replies to untried values.

In the paper this plays the role of showing how a computable process — in fact a *small program* — can be totally surprising, thus making a point about the mechanizability of mental processes. But read another way it is a claim to a cipher system unbreakable even with chosen plaintext — the modern criterion of security.

Once again we can only speculate on what he was doing for GCHQ, and why GCHQ had tried to get him back to work full-time, until his 1952 exclusion. What might have he done if the political establishment had treated him differently? Would his effect on the cold war history of 1954 have been as significant as it was on the Atlantic war of 1944? Both were great wars of information and intelligence.

There is a science-fiction story by the writer Greg Egan [14], which starts on a political footing, discussing what might have happened if Alan Turing had been treated differently by his rulers, and has all sorts of imaginative elements, including a dialogue with the theologian C. S. Lewis. But it goes on to focus on scientific advances by and around a counter-factual Turing of the late 1950s. An important point is that it correctly introduces a focus on fundamental *physics*, a point to which I shall return in concluding this survey. The story is called *Oracle*, a reference to the uncomputable oracle of Turing's 1938–9 paper on ordinal logics [31]. Roughly speaking, an oracle has to contain an infinite amount of information in a finite space, so as to be able to solve a problem unsolvable by any Turing machine, e.g. to supply on demand the answer to the halting problem for every Turing machine. In this excerpt a fictional character links the oracle with time travel:

> ... "Time travel," Helen said, "gives me the chance to become an Oracle. There's a way to exploit the inability to change your own past, a way to squeeze an infinite number of timelike paths — none of them closed, but some of them arbitrarily near it — into a finite physical system. Once you do that, you can solve the halting problem ..."

2 Church's Thesis and Copeland's Thesis

This brings me naturally to B. J. Copeland's influential views on what Turing would have done, because he has also raised the prospect of actually building such oracles — not as science fiction, but as a serious possibility for future technology. This is the prospectus of so-called hypercomputation. Moreover, he and his colleague D. Proudfoot have associated these ambitions with Turing's views and given the impression that these are lost ideas of Turing's which can now be recovered and perhaps implemented.

There is a very general sense in which I agree with Copeland: the physical world should not be assumed computable without further investigation. This point was made long ago by Chaitin [2] and no doubt by many others.

Certainly we should now be more penetrating in the analysis of the concept of "mechanical," with the benefit of modern physical knowledge. I also agree that Turing himself, if he had lived, would have been very interested in such investigation — just as that science-fiction story suggests. His interest in mathematical logic was not the rather narrow and technical one sometimes found in the modern discipline: his work might be characterized as using post-Gödel logic as a branch of applied mathematics. But, for reasons to be outlined in what follows, I find no reason whatever to associate Turing with the "hypercomputation" prospectus which Copeland has advanced. In making this association, Copeland has emphasized to his wide audience that he contradicts the picture of Turing's ideas as advanced by other, more conventional commentators. I have had the privilege of being treated as a representative of this traditional school of thought. Thus, Copeland and Proudfoot informed the readership of the *Times Literary Supplement* [7] that:

> Taking their cue from Turing's 1939 paper, a small but growing international group of researchers is interested in the possibility of constructing machines capable of computing more than the universal Turing machine ... research in this direction could lead to the biggest change computing has seen since 1948. Hodges's Turing would regard their work as a search for the impossible. We suspect that the real Turing would think differently.

By machines capable of computing more than the universal Turing machine, Copeland refers to the 'oracles' which he and D. Proudfoot described in terms of infinite-precision measurements in their *Scientific American* article [8], and which are criticized by Martin Davis in this volume. The allusion to 1948 (the first working stored-program computer, giving rise to the IT industry of today) shows the economic seriousness of what he has in mind. If Turing's name were truly associated with this possibility, that would give it much greater significance and credibility.

What is the difference between Copeland's "real" Turing and "my" Turing? I had written in [20, p. 109], summarizing what Turing had achieved in 1936:

> Alan ... had discovered ... a universal machine that could take over the work of any machine ...

Copeland claims [7] that I made an important error here in writing "machine" rather than writing explicitly "Turing machine." This is because:

> Turing himself described abstract machines whose mathematical abilities exceed those of the universal Turing machine (in a ground-breaking paper published in 1939).

Copeland in a more academic paper [11] criticizes the same sentence, for the same reason, and there says that I expressed a "common view." Indeed I did. My statement about machines lay in entirely respectable company:

not only within the mainstream of mathematical logic, but reflecting the description of Turing's work that Church himself gave. Although Turing's description of the Turing machine was couched in terms of imitating a human being following some procedure, Church characterized computable functions, when introducing them to the world in the *Journal of Symbolic Logic*, in these words [3]:

> The author [i.e. Turing] proposes as a criterion that an infinite sequence of digits 0 and 1 be "computable" that it shall be possible to devise a computing machine, occupying a finite space and with working parts of finite size, which will write down the sequence to any desired number of terms if allowed to run for a sufficiently long time. As a matter of convenience, certain further restrictions are imposed on the character of the machine, but these are of such a nature as obviously to cause no loss of generality — in particular, a human calculator, provided with pencil and paper and explicit instructions, can be regarded as a kind of Turing machine.

Thus Church described computable functions as those that could be performed *by some machine*. Church drew no distinct line between the human being following a rule, and the action of a finite machine. (If anything, the words "in particular" suggest that Church conceived of a human calculator as the *most powerful* example of a machine.) Church offered no hint of speculation about machines that could exceed the power of Turing machines. In fact, Church's characterization of computability actually excluded this possibility.

Church was famous for meticulous clarity, and he was supervising Turing's Ph.D. at Princeton when he wrote this review, so I cannot believe he made this statement lightly, in ignorance or defiance of Turing's views. Furthermore, he repeated it in 1940 [5] when he knew all about the Turing "oracles" that Copeland thinks are "machines" standing in contradiction to the "common view."

It appears that the background to Copeland's assertion is the desire to maintain simultaneously that so-called "hypercomputing" machines can be built, and that the Church-Turing thesis is correct. This position can only be maintained if Church's thesis was never intended to apply to machines. The readership of *Scientific American* was informed [8] that it was "a myth" that Church's thesis referred to machines, and that

> In truth, Church and Turing claimed only that a universal Turing machine can match the behavior of any human mathematician working with paper and pencil in accordance with an algorithmic method — a considerably weaker claim that certainly does not rule out the possibility of hypermachines.

But the primary characterization of Turing machines in [3], as given above, shows that Church made no such restriction. Indeed, had Church set out to cultivate amongst his readers the "myth" denounced by Copeland and Proudfoot, he could hardly have done so more effectively. Copeland in [6] quotes a secondary statement from Church [4] which employs the expression "an arbitrary machine," and asserts that what Church meant was only that the Turing machine concept or its equivalents would have arbitrary elements in their technical formulation. In mathematical parlance, however, the expression "an arbitrary machine" simply means "any machine whatever," and if there were any doubt about this interpretation one need only look at the primary statement by Church as quoted above.

It is worth standing back to see the context in a little more generality, since the point at issue here does not in fact depend on the exact words used by Turing or Church; it stems from the very nature of what was being addressed by Turing's theory of mind and machine. The problem that faced Turing in 1936, as it again faced him in his theory of "machine intelligence" (see [22] for a recent survey) is that of whether machines can do as much as the mind. This problem is not, of course, Turing's alone: it is a fundamental problem of science, and whether we study Gödel or Penrose, Lucas or Hofstadter, Searle or Dennett, everyone agrees that the basic *question* is whether human minds are super-mechanical, though there is widespread disagreement about the *answer*. Copeland and Proudfoot alone suggest that the problem is the *other way round*, giving the impression that Turing defined computability as he did, because there might be superhuman machines. Copeland offers in [6] as explanation for Turing's definition of computability:

> For among a machine's repertoire of atomic operations there may be those that no human being unaided by machinery can perform.

But this consideration is entirely foreign to Turing's thought. This sentence represents a quite unjustified projection of Copeland's "hypercomputation" thesis into the classical formulations of 1936.

A possibly confusing element is that Turing defined an entity called an "oracle-machine," and indeed described an oracle-machine as "a new type of machine." Is this a contradiction? No: Turing's "oracle-machine," defined for the purpose of exploring the uncomputable within mathematical logic, involves a generalized use of the word "machine" for something that is only *partly* mechanical. (In contrast, of course, Church's thesis concerns the scope of the *purely* mechanical.) The oracle formalizes non-mechanical steps, which can (if given any extra-mathematical interpretation at all) be compared with the "intuition" of seeing the truth of a formally unprovable Gödel statement. The oracle is a non-mechanical entity inside a partially mechanical entity, the oracle-machine. Any doubt about what Turing meant should be dispelled by the clear statement in [31] that:

> We shall not go any further into the nature of this oracle apart from saying that it cannot be a machine.

The nature, and indeed the essential purpose of the oracle, is that it is not a machine. There is a precedent for Turing's use of the word "machine" in this generalized sense: the "choice-machines" defined in Turing's original great paper [30], which ask for a human operator's decisions — by definition, *not* mechanical. These choice-machines also are only *partly* mechanical. If the winner of a Turing Test for machine intelligence were revealed to have a human choice-maker hidden inside the computer, we should not consider the victory much of an achievement. Likewise, if "oracle-machines" were allowed in deciding the Entscheidungsproblem, the question would become trivial. In both cases the *whole point* lies in whether the task can or cannot be done by *purely* mechanical means, and it stands as Turing's great achievement that over sixty years later his encapsulation of the "purely mechanical" by the Turing machine definition still holds sway.

Summarizing, there is nothing in Turing's "ground-breaking paper of 1939" [31] to justify Copeland's sensational technological and economic prospectus about "constructing" oracle-machines.

Nor is there anything in Turing's later work to support Copeland's prospectus for an oracle-based hyper-computer revolution. In Turing's 1948 report [34], which contained an extended account of "machines" in general, oracle-machines never appeared in the analysis. We can also look again at Gandy's 1954 letter [17] for evidence regarding Turing's legacy. Gandy supplied Newman with a long section on Turing's views on the reception of his ordinal logics [31]. This has been cited by Copeland and Proudfoot [10] to suggest that Turing thought his 1939 paper had not been given the attention it deserved. Indeed he did, but Gandy's extensive remarks on Turing's views all referred to his much more advanced ideas in mathematical logic. They did not mention oracles, let alone suggest something to do with seeing oracles as objects that might exist. Martin Davis emphasizes, in this volume, as does Feferman [16], that the "oracle" plays only a very small part in [31].

Copeland has also commended in [6] the later contribution of Gandy to this question, stressing that in [18] Gandy distinguished "Thesis M" (that anything done by a machine is computable) from Church's Thesis. Gandy undertook a rigorous definition of the concept of machine, with this distinction in mind. Copeland does not observe, however that (1) Gandy never even considered counting an "oracle-machine" in this category and that (2) Gandy's results *lend support* to what Church assumed in 1937, viz. that "purely mechanical" does indeed imply "computable." It is not surprising that Gandy never considered oracle-machines in his analysis of the mechanical: he fully reflected Turing's thought as his student and legatee, as well as representing the tradition of mathematical logic.

We now pass to Turing's famous 1950 paper [36], which summarizes Turing's post-1945 claim that the action of the brain must be computable, and therefore can be simulated on a computer. I have already referred to how

Turing used a pseudo-random program to exemplify how a machine can create a "surprise." This was entirely typical of his argument that something *apparently* non-mechanical can in fact be readily computable. But in fact this example also illustrates how his 1950 argument was not merely about the sufficiency of computable functions. Turing's argument was that a *totally finite* machine (with a fixed finite store) would suffice to simulate the finite brain. Thus, in that cipher-based example, Turing emphasized *how small* a store was needed to embody the effect of a "surprise." This point leads me to make a further defense against the charges made in [7] and [11]. For there Copeland asserted that I had overlooked an important reference to uncomputable operations in Turing's 1950 paper [36], asserting that therein one might find Turing saying that:

> An example of a discrete-state machine whose behavior cannot be calculated by a universal Turing machine is a digital computer with an infinite-capacity store and what Turing calls a "random element". (pp. 438–439)

But in fact, an inspection of Turing's argument shows that the "infinite store" just corresponds to the unbounded tape of the Turing machine. It is the arena within which computable operations are defined, not something going beyond computability. As for the "random element," Turing specifically gave a pseudo-random (i.e. entirely computable) illustration of it, namely the digits of π. Thus, these references in Turing's paper only corroborate the fact that Turing saw mental processes as falling within the scope of the computable. In [11] Copeland further argues:

> Hodges ... fails to include the crucial words "discrete state machines ... can be described by such tables *provided they have only a finite number of possible states.*"

But this qualification of "finitely many states" is not crucial at all. In his 1950 paper Turing gave the philosophical world a rather abbreviated description of computability which avoided bringing in the concept of the infinitely long "tape." Instead, his discussion was focused on *totally finite machines*, which do not need to use any tape; or in other words, the states of the tape are absorbed into the states of the machine. (This is why Turing had to refer, rather awkwardly, to an "infinite store" when referring to the full definition of computability.) The condition Copeland asserts to be so important is the condition on a process to be representable by a *tapeless machine*. This is a much more restrictive condition than computability. (A Turing machine has only finitely many configurations, but in general will have an unbounded number of possible states of its tape.)

Again we might well stand back a little to see this in context. The concept of computability takes its power from the fact that it successfully generalizes the concept of a totally finite machine, to one which still has "finite means" but is allowed unlimited time and space for marking a tape. Copeland's blurring of the distinction between the state table of a totally finite machine, and the finite table of behavior of a Turing machine misses the essential point of the definition of computability.

To summarize: this condition does not allude to uncomputable functions in any way. On the contrary, Turing's context shows that in 1950 his focus was on the successful evocation or at least imitation of intelligence within a *finite subset* of computable functions.

3 Computability and Quantum Physics

But now let us move on past 1950, and come finally to the *physics* that I think the most telling and novel aspect of what Turing had started to do and where he might have gone on to far more if he had lived. This deserves to be better known, and here I must acknowledge Copeland more positively. Recently he has published the full script of Turing's 1951 BBC radio talk [37], prefacing it with an analysis [9]. This talk mostly paraphrased Turing's famous 1950 paper [36] in a form suitable for a short talk, but, as Copeland usefully points out, it had a significant new feature. It had a mention of quantum mechanics, introduced specifically as a loophole in Turing's otherwise general argument that the action of the brain must be computable. Turing explained that for the success of this argument it is

> ... necessary that this machine [the brain] should be of the sort whose behavior is in principle predictable by calculation. We certainly do not know how any such calculation should be done, and it was even argued by Sir Arthur Eddington that on account of the indeterminacy principle in quantum mechanics no such prediction is even theoretically possible.

This is *the only* sentence in all Turing's work that points to something physical that may not be reducible to computable action. But it is a significant one. It runs against what Turing had said about simulating the nervous system by a computer in [36]. And here, exceptionally, Turing does *not* appeal to pseudo-random simulation as a satisfactory discussion of "randomness."

This discussion has nothing whatever to do with oracles. There is no mention whatever of infinite information sources in here. (Note also that Turing's thought is still in the context of wondering whether any machine can do as much as the mind, and not in the spurious reverse problem posited by Copeland!) The question raised by Turing is to do with fundamental physics: is the physical space-time of quantum mechanical processes, with its so-called Heisenberg uncertainty principle, compatible with a Turing machine model?

This sentence, taken seriously, makes a link between the computability of mental processes and Turing's late work in physics. Although I described this late physics work in [20], page 495, and noted Turing's harking back to

Eddington, I had not seen the importance of this possible connection between fundamental physics and the question of the computability of the mind. To describe more satisfactorily this work of 1953–4, I return yet again to Gandy's 1954 letter [17]. In fact, it was to this subject, rather than to computer science, mathematics, logic or morphogenesis, that Gandy devoted the most attention:

> During this spring [1954] he spent some time inventing a new quantum mechanics ... it did show him at his most lively and inventive; he said "Quantum mechanists always seem to require infinitely many dimensions; I don't think I can cope with so many, I'm going to have about 100 or so — that ought to be enough don't you think?" Then he produced a slogan "Description must be non-linear, prediction must be linear."
>
> A slightly more serious contribution ... uses "the Turing Paradox"; it is easy to show using standard theory that if a system starts in an eigenstate of some observable, and measurements are made of that observable N times a second, then, even if the state is not a stationary one, the probability that the system will be in the same state after, say, 1 second, tends to one as N tends to infinity; i.e. that continual observation will prevent motion ...

His "non-linear" description in quantum mechanics would have implied some essentially new theory, and the word "measurement" tells us the focus of his attempted innovation. Turing was referring here to the puzzle of the *reduction, collapse, or measurement* process in quantum mechanics. No-one even now can say when or how it occurs — as Turing was pointing out with his Paradox.

The problematic foundations of quantum mechanics were not new to Alan Turing. His interest went back to 1928. Then he had read Eddington's *The Nature of the Physical World,* with Christopher Morcom his beloved schoolfriend. In fact Alan Turing was one of the first serious readers of von Neumann's 1932 monograph on the Mathematical Foundations of Quantum Mechanics. It was his school prize book, given after Christopher Morcom had suddenly died in 1930. In 1933 Alan Turing reported of it, "My prize book from Sherborne is turning out very interesting, and not at all difficult reading, although the applied mathematicians seem to find it rather strong." [28]

Von Neumann's axioms distinguished the **U** (unitary evolution) and **R** (reduction) rules of quantum mechanics. Now, quantum computing so far (in the work of Feynman, Deutsch, Shor, etc.) is based on the **U** process and so computable. It has not made serious use of the **R** process: the unpredictable element that comes in with reduction, measurement, or collapse of the wave function. Maybe Turing, if he had lived, would have developed quantum computing — but from the scraps that have survived it appears that it was the mystery of the **R** process that really intrigued him.

Recently the **R** process has been studied with fresh experimental interest, and in my view these more recent investigations give the flavor of where Turing's thought might have gone. Elitzur and Vaidman [15] have shown that the logic of "reduction" can produce an extraordinary result. Suppose a "live bomb" is a device which effects "measurement" or "reduction," whilst a "dud bomb" is a device which does not. Then the type of device can be tested by observing the final state of a photon which hits the device. Using classical measurement, the determination would amount to seeing whether or not the device "exploded." With a quantum measurement it is possible to deduce that the device was "live" without any explosion taking place! Quantum mechanics should not be thought of as necessarily introducing uncertainty into a classical picture: in this example it implies the testing *with certainty* of a counterfactual story — what *would have happened* if the photon had hit the detonator of the live bomb. The logical structure here is no different from that known to von Neumann in 1932, but modern technology with perfect mirrors and the detection of single photons makes it possible to investigate that logic far more stringently. In particular, Anton Zeilinger and co-workers in Vienna are conducting ingenious experiments designed to test the limits of the **U** and **R** rules. These investigations do not analyze the internal dynamics of the **R** process and explain when, how, and indeed *whether* it actually happens, which Turing was probably trying to do. But they are probing the logic of quantum mechanics in a way that would have fascinated him.

Turing was probably trying to make quantum mechanics fully predictable, which no-one has been able to do, and perhaps also, as Gandy hinted in his note, more finite. That, if achieved, would have filled in the loophole in his argument about mechanizing thought. If so, Turing's agenda was in a sense the opposite of that of Penrose [25,26]. Penrose has argued that the **R** process must be uncomputable because thought cannot be computable — as follows from taking a very strong view of the implications of Gödel's theorem. But Turing is on common ground with Penrose in taking quantum mechanics and Gödel's theorem very seriously in discussing the question of Artificial Intelligence.

There are still open questions about quantum mechanics, almost as open as when Alan Turing was twenty and wrote his first ideas about the mind [27]:

> It used to be supposed in science that if everything was known about the Universe at any particular moment then we can predict what it will be all through the future ... More modern science however has come to the conclusion that when we are dealing with the atoms and electrons ... We have a will which is able to determine the action of atoms probably in a small portion of the brain ...

By "modern science" he meant quantum mechanics, as he had learnt at school from Eddington. At that stage he thought of there being some unknown quantum mechanical law which accounted for the action of human will. Presumably he changed his mind, since the emphasis of all his post-war work was so strongly towards eliminating such concepts as will and consciousness. But we cannot tell what he might have gone on to think after 1954. In his last years, he insisted on his individuality and his freedom. As a human being, he actually took his own will and consciousness very seriously, and this is one of the great paradoxes of his life and his work.

Church's thesis and the Turing machine are rooted in the concept of "doing one thing at a time." But we do not really know what "doing" is — or time — without a complete picture of quantum mechanics, and the relationship between the still mysterious wave-function and macroscopic observation. Alan Turing found his greatest strength when studying the interfaces between conventional compartments of scientific thought, and might have come up with something between logic and physics that no-one could possibly have predicted.

References

1. Bates, M. A. (1950). On the mechanical solution of a problem in Church's lambda-calculus, M.Sc. thesis, Manchester University, October 1950.
2. Chaitin, G. J. (1982). Gödel's Theorem and Information, International Journal of Theoretical Physics, **22**, 941–954
3. Church, A. (1937). Review of [30]. J. Symbolic Logic. **2**, 42–43.
4. Church, A. (1937). Review of Post (1936). J. Symbolic Logic. **2**, 43.
5. Church, A. (1940). On the concept of a random sequence, Bull. Amer. Math. Soc. **46**, 130–5.
6. Copeland. B. J. (1997). The Church-Turing thesis, in E. N. Zalta (ed.), Stanford Encyclopaedia of Philosophy, `http://plato.stanford.edu`.
7. Copeland B. J. and D. Proudfoot (1998). Enigma variations (London: Times Literary Supplement, 3 July 1998.)
8. Copeland B. J. and D. Proudfoot (1999). Alan Turing's forgotten ideas in computer science. Scientific American, **253:4**, 98–103.
9. Copeland B. J. (1999). A lecture and two radio broadcasts on machine intelligence by Alan Turing, in Machine Intelligence **15**, K. Furukawa, D. Michie. and S. Muggleton (eds.), Oxford University Press, Oxford.
10. Copeland B. J. and D. Proudfoot (1999). Review of *The Legacy of Alan Turing*, Mind, **108**, 187–195.
11. Copeland, B. J. (2000). Narrow versus Wide Mechanism: Including a Re-examination of Turing's Views on the Mind-Machine Issue. J. of Phil. **96**, 5–32.
12. Davis, M. (1958). Computability and Unsolvability (New York: McGraw-Hill); with appendix on Hilbert's Tenth Problem, Dover edition (1982)
13. De Leeuw, K., E. F. Moore, C. E. Shannon, and N. Shapiro (1956). Computability by Probabilistic Machines. Automata Studies, Shannon, C. and J. McCarthy, eds., Princeton University Press, 183–212.
14. Egan, G. (2000). Oracle, Asimov's Science Fiction, July 2000.
15. Elitzur, A. C. and L. Vaidman (1993). Quantum-mechanical interaction-free measurements, Found. of Physics **23**, 987–97
16. Feferman, S. (1988). Turing in the Land of O(Z), in [19]; an updated version appears in [39].
17. Gandy, R. O. (1954). letter to M. H. A. Newman, in the Turing Archive, King's College, Cambridge; included in [39].

18. Gandy, R. O. (1980). Principles of Mechanisms, in The Kleene Symposium, eds. J. Barwise, H. J. Keisler and K. Kunen, North-Holland, Amsterdam.
19. Herken R. (ed.) (1988). The universal Turing machine: a half-century survey, Kammerer und Unverzagt, Berlin; Oxford University Press, Oxford.
20. Hodges, A. (1983). Alan Turing: the enigma (Burnett, London; Simon & Schuster, New York; new editions Vintage, London, 1992, Walker, New York, 2000). Further material is on http://www.turing.org.uk.
21. Hodges, A. (1997). Turing, a natural philosopher (Phoenix, London; Routledge, New York, 1999). Included in: The Great Philosophers (eds. R. Monk and F. Raphael, Weidenfeld and Nicolson 2000)
22. Hodges, A. (2002). Alan M. Turing, in E. N. Zalta (ed.), Stanford Encyclopaedia of Philosophy, http://plato.stanford.edu.
23. Hodges, A. (2003). Alan Turing and the Turing Test, in The Turing Test Sourcebook: Philosophical and Methodological Issues in the Quest for the Thinking Computer, ed. Robert Epstein, forthcoming.
24. Newman, M. H. A. (1951). The influence of automatic computers on mathematical methods, Manchester University Computer Inaugural Conference, July 1951.
25. Penrose, R. (1989). The emperor's new mind, Oxford University Press, Oxford.
26. Penrose, R. (1994). Shadows of the mind, Oxford University Press, Oxford.
27. Turing, A. M. (1932). Handwritten essay, Nature of Spirit, photocopy in the Turing Archive, King's College, Cambridge; see [20, p. 63].
28. Turing A. M. (1933). Letter to Sara Turing, 16 October 1933, in the Turing Archive, King's College Cambridge; see [20, p. 79].
29. Turing A. M. (1936). Letter to Sara Turing, 14 October 1936, in the Turing Archive, King's College Cambridge; see [20, p. 120].
30. Turing A. M. (1936–7). On computable numbers, with an application to the Entscheidungsproblem, Proc. London Maths. Soc., ser. 2, **42**, 230–265; also in M. Davis, (ed.) The Undecidable (Raven, New York, 1965), and in [40]
31. Turing A. M. (1939). Systems of Logic defined by Ordinals, Proc. Lond. Math. Soc., ser. 2, **45**, 161–228; also in M. Davis (ed.). The Undecidable (Raven, New York, 1965). and in [40]. This was Turing's 1938 Ph.D. thesis, Princeton University.
32. Turing, A. M. (1946). Proposed electronic calculator, unpublished report for National Physical Laboratory, London; published in A. M. Turing's ACE Report of 1946 and other papers (eds. B. E. Carpenter and R. W. Doran, MIT Press, Cambridge, MA, 1986), and in [39].
33. Turing, A. M., 1947, Lecture to the London Mathematical Society on 20 February 1947, published in A. M. Turing's ACE report of 1946 and other papers, (eds. B. E. Carpenter and R. W. Doran, MIT Press, Cambridge, MA, 1986), and in [39].
34. Turing A. M. (1948). Intelligent machinery, unpublished report for National Physical Laboratory, London; published (ed. D. Michie) in Machine Intelligence **7**, 1969, and in [39].
35. Turing A. M., (1950). The word problem in semi-groups with cancellation, Ann. of Math. **52** (2), 491–505.
36. Turing A. M. (1950). Computing machinery and intelligence, Mind **49**, 433–460, reprinted in [39].

37. Turing, A. M. (1951). BBC radio talk, transcript in the Turing Archive, Kin
College, Cambridge; published in Machine Intelligence **15**, eds. K. Furukaw
D. Michie. and S. Muggleton, Oxford University Press, Oxford, 1999.
38. Turing, A. M. (1954). Solvable and unsolvable problems, Science News **31**, 7–
39. Turing, A. M. (1992). Collected Works: Mechanical Intelligence. D.C. Ince, e
North-Holland, Amsterdam.
40. Turing, A. M. (2001). Collected Works: Mathematical Logic. R. O. Gandy a
C. E. M. Yates, eds., North-Holland, Amsterdam.