

Some Assumptions about Problem Solving Representation in Turing's Model of Intelligence

Raymundo Morado and Francisco Hernández-Quiroz

National University of Mexico (UNAM)
Francisco Hernández-Quiroz
fhq@ciencias.unam.mx

Abstract: Turing machines as a model of intelligence can be motivated under some assumptions, both mathematical and philosophical. Some of these are about the possibility, the necessity, and the limits of representing problem solving by mechanical means. The assumptions about representation that we consider in this paper are related to information representability and availability, processing as solving, non-essentiality of complexity issues, and finiteness, discreteness and sequentiality of the representation. We discuss these assumptions and particularly something that might happen if

they were to be rejected or weakened. Tinkering with these assumptions sheds light on the import of alternative computational models.

Keywords: Turing machines, intelligence, problem solving, representation, computational models Text

Acknowledgement: We would like to thank Teresa Numerico, Luciano Floridi and Gordana Dodig-Crnkovic for their useful suggestions.

1 Introduction.

What does a zeroes-and-ones-writing car box have to do with intelligence? This paper explores some idealizations that may lead to model methodical intelligence with Turing Machines (TM's) and equivalent models.¹ Often the specification of TM's leaves unexplained the intuitions that lead to posit those characteristics and not others.

As it is well known, Turing presented this model in Turing (1936-7). In that paper Turing made no explicit contention about TM's being a model of intelligence in general. That changed in Turing (1950) whose main point is to analyze the notion of thinking machines. By machines he meant digital computers equivalent to TM's. Turing's ideas are remarkably coherent in these two papers. We believe that most assumptions mentioned below coincide with Alan Turing's ideas (even if they are not worded in the same fashion) and that they offer a starting point to discuss different aspects of the TM's as a model of an acceptable notion of intelligence. Nevertheless, it must be clear that, unless we explicitly quote Turing, the identification of such assumptions is a matter of our interpretation based on the features of the model.²

¹ Gödel (1934), Church (1936), Post (1936, 1943, 1946), among others.

² We thank Teresa Numerico for reminding us that the historical Turing does not always coincide with the traditional view of Turing, at least as far as we have been able to reconstruct it. Turing himself was conscious of the limitations of his model and in both published and unpublished work he explored other possibilities. Our analysis does not take into account these possibilities, as they are not normally considered part of Turing's model of intelligence.

Our goal is to revisit the possible assumptions of the model and to ponder some considerations in favor or against adopting it. We consider two main kinds of assumptions in Turing's model of effective computation: problem representation related, and solving method related. In this paper we deal with the assumptions about representation in general and about the specific kind of representation offered. The assumptions about method are the focus of Hernández-Quiroz and Morado (2005).

Our exegesis tries to shed light on both essential and non-essential principles behind the TM's model. A principle might be essential in the sense that its removal from the model changes the computational power of the model. It also might be essential in the sense that its removal changes the nature of the model whether it changes its computational power or not. In the first sense of "essential", some of the principles can be removed without affecting the computational power of the model, but there are others whose removal opens the door to models with greater, lesser, or just plain different expressive and computational capability. In the second sense of "essential", while some principles are of a cosmetic nature or redundant, some can change the nature of the model to a degree that their presence might force us to change our views about the adequacy of the model to our unschooled intuitions about intelligence and mechanical problem solving.

Turing assumptions can be regarded in two different ways: On the one hand he is trying to characterize positively what can be solved mechanically. On the other hand, he is trying to put limits to what is not mechanically solvable. Under this light, some of his unrealistic assumptions are useful in order to obtain a negative proof of existence. For instance, to show that no TM can do something, we need to avoid any unnecessary limits to the possible length of its tape; any restriction in length could be challenged as unfair to the possibility of bigger machines. So, the tape must be infinite in order to curb any arbitrary limitations. If a function is not computable by any TM with an infinite tape, then it is definitely not computable by a TM with a finite tape. This way we can show that any tape length has been implicitly considered. Of course, this is an unrealistic assumption with theoretical advantages. It is not intended to be a portrayal of actual TMs but a device to show the limitations in power of any TM.

2 General Assumptions about Representation.

2.1 Problem Representability.

In the TM model, we suppose that any problem that can be solved mechanically can be represented.³ We do not assume that each problem is linguistic in nature, but only that it can be represented. Accordingly, Turing uses a system of formulae both to present the problem or initial state (which can be empty), and to present the answer.

Is this assumption acceptable? We must distinguish between the need and the possibility of having some representation to solve a problem. It has been claimed that some problem-solving systems proceed without a representation of the problem. This claim is buttressed by the intuition that some primitive organisms can solve problems without representing them.⁴

Turing is normally understood as thinking about representation in languages with a finite base vocabulary.⁵ In this context the possibility of always having a representation is also debatable. The number of languages generated by all grammars from a finite vocabulary is countable and therefore the number of problems that can be stated as an expression in one of these languages is also countable. On the other hand, every relation between natural numbers (if we restrict ourselves to arithmetical problems) can be seen as posing the problem of finding the characteristic function for the set of pairs of numbers in the relation and we know the number of these relations to be non-countable. Of course, if we go beyond arithmetic the number of problems is even bigger.

³ Turing was probably thinking of something like this when he wrote: "Computing is normally done by writing certain symbols on paper" in Turing (1936-7) p. 249.

⁴ Churchland (1999).

⁵ Cfr. Soare (1999).

A possible reply might be to say that perhaps all those non-representable problems fall into the category of problems not mechanically solvable. But there is another argument against this. There is an intuitive sense in which all physical problems are mechanical problems, and reality computes the relevant values as it goes. Some people claim that all such physical problems correspond to real-valued functions and it could well be the case (if not actually, at least in principle) that the number of these physical problems be non-denumerable.⁶ But syntactical representations are at most denumerable, so the number of “mechanically solvable problems” is greater than the number of possible syntactical representations and, therefore, there are non-representable problems that are mechanically solvable.

2.2 Information Processing

“There is, however, a limit to the number of mechanical linkages in an automobile engine that can be replaced by information-transmitting devices. The engine ought, after all, to deliver mechanical power to the wheels of the car.”
Joseph Weizenbaum (1976), p. 42.

Turing’s model presupposes that to solve a problem is akin to processing information. Many solutions require of us to do something, to create, destroy or modify reality. In Turing’s model such modifications correspond to modifications in the representation of the problem. This leads to model the capability of solving problems as the capability of modifying a representation until it becomes a representation which is the solution. For instance, in the realm of mathematics, writing down a solution is often seen as solving the problem. (There are even mathematicians who would claim they found a solution to a problem P if they find a method to solve a class of problems which contains P, even if the method cannot be physically carried out.)

But acting on reality cannot be identified with acting on one of its representations. To think that the original problem is solved as soon as we have processed its representation into a representation of the solution would be tantamount to sympathetic magic. (We do sometimes talk about winning a war on paper, or finding an exit in a map.)

Intuitionism had already reminded us that finding a solution is different from finding that there is a solution. Turing’s model overlooks that solving and transforming the representation of the problem into a representation of the solution are two different things. This model ignores the need for an action from the agent in order for a solution to become reality.

2.3 Solving as syntactic transformation

Turing assumes that the representation can be syntactic in the sense that it can be accomplished with a finite⁷ set of symbols. The model assumes that to solve a problem is to change its representation, *i.e.*, to “read” the problem and to “write,” perhaps on the same spot, the solution. The actions of the computer will be therefore specified by telling how to modify the current symbol or how to move on to read another one. This notion of symbolic transformation was dear to Hilbert (1928).⁸ Since we want the computer to be able to react differently to the same symbol, we also must specify the inner changes of its internal state. So, we have two paths of action: (1) to update the internal estate and replace the scanned symbol with another one, or (2) to update the internal state and take a step to examine another symbol. Therefore, the

⁶ Van Gelder and Port (1995), p. 22.

⁷ “I shall also suppose that the number of symbols which may be printed is finite”, Turing (1936-7), p. 249.

⁸ Some features of Turing’s Model can be better understood if we remember that his 1936-7 paper was an attempt to give an answer to the Decision Problem as posed by Hilbert, *i.e.*, finding an effective method to assess whether a given statement in predicate calculus is also a mathematical theorem. We thank Professor Luciano Floridi for suggesting to us to develop further this line of research in a future paper.

transition function must specify the action of reaching another internal state and of moving on to search for symbols. If nothing is specified, the machine stops.

This important assumption is tightly tied to the question of acceptable solving methods and deserves special consideration which we try to give in Hernández-Quiroz and Morado (2005).

2.4 Stimulus and representation

In the TM model it is assumed that the machine perceives stimuli and can be affected by different situations. This assumption is necessary if we also accept that to solve a problem is like processing information. Both assumptions stand or fall together. A limit case would be the empty stimulus, e.g., when all relevant elements of the problem are hardwired into the machine and then no external stimuli are needed. We can build machines that automatically generate Fibonacci sequences, Pi expansions, logarithmic tables, etc., and do not need to react to external stimuli. An extreme example is Geulincx's fantasy of a clockwork mechanism that solves all problems in the external world by virtue of its internal workings, and therefore gives the illusion of being reacting to external phenomena (Geulincx (1691)).

Nevertheless, if our machine is to be general enough to handle different but related problems, it needs to register aspects of individual instances of the problems in order to give differentiated answers. (An alternative would be to have infinite machines, each one solving a particular instance of a problem, just the opposite of having a Universal Turing Machine.)

Furthermore, since each situation is represented symbolically, the machine in our model must be able to read symbols, that is, to react in the presence of a symbol. It is not essential that there be only one reading head; there might be several ones, acting simultaneously (just as we simultaneously receive information from several senses).

Peter Wegner (1997) has criticized the way TM's handle their input. He claims that a TM's input is fixed before any computation begins and therefore interaction (i.e., a dynamically changing input) cannot be accounted for properly by TM's. If his argument is correct, then it only emphasizes the importance of a mechanism for handling stimuli.

2.5 Information acquisition

The machine must be capable of some actions to gather information. We assume that in order to grasp more symbols the machine could do something such as moving its reading head or moving itself. The information (which might be null) can be read part by part or all at once. This idea of focusing the attention of the machine is captured in the model with the movements of the reading head.

Notice that this model is equivalent to one in which the input tape moves under the scanning head of the TM. We must distinguish between the changes in stimuli and the ability to choose where to look for the next stimuli. The issue is not the capacity to receive external stimuli, but the ability to control in some degree which stimulus will be considered next.

The basic idea that the model should incorporate some form of information acquisition seems inescapable. Nevertheless, the particulars about how to acquire information are debatable. See Morado and Hernández-Quiroz (2005) on the methodological assumptions.

3 Specific assumptions about representation

3.1 Finiteness of vocabulary

In Turing's model, the number of symbols is finite. This is no big limitation if we accept that the calculation must be a finite number of transformations on finite sequences of symbols. Therefore each calculation will need less than infinite symbols and so our language does not need to provide with more

than a finite number of them. As a matter of fact, we can limit ourselves to two symbols without loss of generality.

Given the methodological assumption that effectively calculable solutions demand only a finite number of operations, this assumption also holds.⁹

3.2 Finiteness of the stimulus

Although the finite number of symbols does not force a finite input, it is common to assume that in order to solve effectively a problem the initial input must be finite (possibly empty). Computation is supposed to start with a finite number of input symbols so the computer, in all its limitations, can take note of the input before finishing the processing, although a complete scan of the input is not necessary before starting the processing nor before finding a solution.

But finiteness of the stimulus is not necessarily the case. Even with infinite inputs it is possible to solve some problems. *E.g.*, there are data streams of which we can ask (and answer) whether they contain more than two digits. To get the answer it is enough to analyze a finite initial segment of the infinite input.

It looks like any infinite representation of a problem that can be solved in a finite amount of time, has to be solved by inspections of only a finite initial segment of the infinite representation. But then this finite initial segment is an equivalent representation of the original problem and the assumption stands. Even if not all solving of a problem is a solving of a finite representation, it is always equivalent to one.

3.3 Sequentiality of the representation

It is taken for granted that after each action the agent is allowed to scan its environment. TM's represent the sequence of symbols with a "tape." To the left and to the right of the currently scanned symbol there might be other symbols representing the information we would encounter were we to take a course of action or another. Of course, at any given moment there might be many more than two alternative actions for the agent, and more than one stimulus at the same time.¹⁰ All cases of multiple and multi-dimensional tapes reduce to the one-tape, one-dimensional case.

This assumption is not surprising given the assumption that the input must be represented finitely. There are relatively trivial encodings that transform non-sequential information into a sequential one.

3.4 Two kinds of memory

TM's have a kind of memory, residing in the set of possible machines states (called m-configurations by Turing). Although there is no maximum number of states for all machines, each one has a pre-established bound which is strictly finite and defined within the decision matrix for the TM's. This is analogous to the common belief that the number of different states for our brain or mind is finite, that there are quanta of brain or mind configurations.

If this were the only kind of memory for TM's, the computational power would be reduced to only that of regular automata. But Turing wants a model with more power, where the computing agent can store and retrieve additional information. There is therefore the need to assume some other form of memory.

⁹ This assumption is discussed more fully in our aforementioned paper. We would like to remark here that such an assumption is in accord with the conception (already common at that time) that to solve *effectively* a problem demands the processing of a finite amount of information. Cfr. Hilbert (1918).

¹⁰ This is the case under the assumption that problems are separable, and no interference between concurrent information streams occur, an assumption that has been challenged by Peter Wegner (1997). We would like to thank Gordana Dodig-Crnkovic for calling our attention about this assumption.

3.5 Access to memory

The second kind of memory corresponds to the information stored on the tape at any given moment of the computation. TM's can access this information for reading and writing in two different directions. If we restrict this assumption in any way, we end up with a less powerful computational model. *E.g.*, if the agent can only read in one direction and write in the opposite one, the tape becomes a stack and the resulting mechanism has the computational power of just a push-down automaton.

Note however that machines with a read and write two-direction tape can be simulated with a two-stack machine (with a read-only tape).

3.6 Spatial complexity

Turing's model takes for granted that we do not know whether there is a pre-established limit to the problem's spatial complexity. We are not talking here about the complexity of the input, but of the size order of the memory needed for the processing. Because of this, the tape must be infinite. If we were to reject this assumption, we would have a strictly weaker mechanism than a Turing Machine, namely, linear bounded automaton.

Why not bite the bullet, abandon this assumption and say that an agent can effectively compute exactly those things that can be computed with a finite memory? Well, just how finite? If we say: "however finite, as long as it is finite," then something would be effectively computable exactly when it were effectively computable with a finite memory. But this is equivalent to what can be done with an infinite memory. So Turing's assumption seems inescapable if we are not to limit beforehand the amount of memory available.

Turing's assumption might be necessary, but how realistic is it? Can we expand to any arbitrary length the amount of memory available for computation? Human memory might be infinitely expandable, for instance, through culture given other assumptions about the physical universe.

3.7 Discrete representation

It is assumed that the transit between m-configurations is discrete. Although Turing believes this is never so in reality, he considers it an innocent idealization. So, the tape is divided into discrete squares.

Without a doubt, actual computers work under these principles (pretending to be discrete). Nevertheless, there is an ongoing debate about whether it is possible to envision a different situation, for instance, with explicitly analogical methods and data. In this case, *e.g.*, it would be possible to compute real numbers in excess of the ones computable with TM's. This is a highly speculative approach although not without supporters (see Siegelmann (1995) and (1999)). See Davis (2004) for a criticism of this view.

4 Final Remarks

We have tried to make explicit principles that might support the TM's model of mechanical computation as a form of intelligence. There are both essential and non-essential ones. This distinction is important, especially if we do not endorse the TM's as the only model of mechanical problem solving.

Proponents of alternative models must face one of two tasks: If they want to say that the TM's model has principles that should be omitted, they should try to single out what essential principle or principles should be rejected. If they want to say that the model is lacking some necessary principles, they will have to state the missing principles. The rejection or addition of any essential principles will lead to the development of alternative models in the sense that they will have a different computational power.

References

- Church, A. (1936) "An unsolvable problem of elementary number theory". *The American Journal of Mathematics*, 58, pp 345–363.
- Churchland, P.M. (1996) "Learning and Conceptual Change: The View from the Neurons", in A. Clark and P.J.R. Millican, (eds.) (1996) *Machines and Thought*, Clarendon Press.
- Davis, M. (2004) "The Myth of Hypercomputation", in Ch. Teuscher (ed.), *Alan Turing: Life and Legacy of a Great Thinker*. Berlin-Heidelberg-New York, Springer-Verlag.
- Geulincx, A. (1691) *Metaphysica vera et ad mentem peripateticam*, Amsterdam.
- Gödel, K. (1934) "On Undecidable Propositions of Formal Mathematical Systems". Mimeographed lecture notes taken by Stephen C. Kleene and J. Barkley Rosser. Reprinted in S. Feferman, John W. Dawson, Jr., Stephen C. Kleene, G. Moore, R. Solovay, and Jean van Heijenoort (eds.) (1986) *Kurt Gödel: Collected Works*, Volume I: Publications 1929-1936.
- Hilbert, D. (1918) "Axiomatisches Denken", *Mathematische Annalen*, 78, pp 405-415.
- Hilbert, D. and Ackermann W. (1928) *Grundzüge der Theoretischen Logik*. Translated as *Principles of Mathematical Logic*, Chelsea Publishing Company, 1950.
- Morado, R. and Hernández-Quiroz F. (2005) "Some Assumptions about Problem Solving Method in Turing's Model of Intelligence", in M. H. Hamza (ed.) (2005) *Computational Intelligence*, Acta Press.
- Post, E. L. (1936) "Finite Combinatory Processes - Formulation 1". *Journal of Symbolic Logic*, 1, pp 103-105.
- Post, E. L. (1943) "Formal Reductions of the General Combinatorial Decision Problem". *American Journal of Mathematics*, 65, pp 197-215.
- Post, E. L. (1946) "A Variant of a Recursively Unsolvable Problem". *Bulletin of the American Mathematical Society*, 52, pp 264-268.
- Siegelmann, H.T. (1995) "Computation beyond the Turing Limit", *Science*, vol. 268, pp 545-548.
- Siegelmann, H.T. (1999) *Neural Networks and Analog Computation: Beyond the Turing Limit*, Birkhäuser, Progress in Theoretical Computer Science.
- Soare, R.I. (1999) "The History and Concept of Computability", in E.R. Griffor, (ed.) (1999) *Handbook of Computability Theory*, Elsevier.
- Turing, A. M. (1936-7) "On Computable Numbers, with an Application to the Entscheidungsproblem", *Proceedings of the London Mathematical Society*, series 2, vol. 42, pp 230-265.
- Turing, A. M. (1950) "Computing Machinery and Intelligence", *Mind*, 59, pp 433-460.
- van Gelder, T. and Port R. F. (eds.) (1995) *Mind as motion: explorations in the dynamics of cognition*, Cambridge, Mass., MIT Press.
- Wegner, P. (1997) "Why Interaction is More Powerful Than Algorithms", *Comm. ACM*, May 1997.
- Weizenbaum, J. (1976) *Computer Power and Human Reason: From Judgment to Calculation*, San Francisco, W. H. Freeman.