Tito Marques Palmeiro

Informatics: Science or *Téchne*?

Informática: Ciência ou *Téchne*?

Abstract

Informatics is generally understood as a "new technology" and is therewith discussed according to technological aspects such as speed, data retrieval, information control and so on. Its widespread use from home appliances to enterprises and universities is not the result of a clear-cut analysis of its inner possibilities but is rather dependent on all sorts of ideological promises of unlimited progress. We will discuss the theoretical definition of informatics proposed in 1936 by Alan Turing in order to show that it should be taken as final and complete. This definition has no relation to the technology because Turing defines computers as doing the work of solving problems with numbers. *This formal definition implies nonetheless a relation to the non-formalized elements around informatics, which we shall discuss through the Greek notion of téchne.*

Keywords: Alan Turing; Definition of informatics; Turing Machine; Téchne.

Resumo

A informática é geralmente entendida como uma "nova tecnologia" e é discutida de acordo com aspectos tecnológicos tais como velocidade, recuperação de dados, controle de informações e assim por diante. Seu uso generalizado, de eletrodomésticos a empresas e universidades não é o resultado de uma análise clara e precisa de suas possibilidades internas, mas é bastante dependente de todos os tipos de promessas ideológicas de progresso ilimitado. Discutiremos a definição teórica de informática proposta em 1936 por Alan Turing para mostrar que ela deve ser considerada como final e completa. Essa definição não tem nenhuma relação com a tecnologia, porque Turing define computadores como fazendo o trabalho de resolver problemas com números. Esta definição formal implica, no entanto, uma relação como elementos não formalizados em torno da informática, que discutiremos através da noção grega de téchne.

Palavras-chave: Alan Turing, Definição de Informática, Máquina de Turing, *Téchne.*

* Professor da Universidade Estadual do Rio de Janeiro.

The ancient Greek science of numbers opened the way, in the last century, to the new and unexpected endeavor called *informatics*. The mathematical origin of this new discipline explains why computer programs can be studied according to the standard of aprioristic truth and formally analyzed in their correctness, but this origin does not suffice to explain the proper nature of informatics. The absence of a generally accepted definition makes room for the commonsensical discourse praising outstanding technological evolutions, and tends to render it dependent on all sorts of ideological promises of unlimited progress.

In this text we shall not propose a definition of informatics, but defend the idea that this definition already exists, and that it is as old as informatics itself. In the following pages, we are going to discuss the origin of this definition in Alan Turing's 1936 paper, "On Computable Numbers, with an Application to the *Entscheidungsproblem*". One may object that a single text written a long time ago, on 36 sheets of paper (appendix included), could not explain everything that informatics came to be. However, we shall concentrate on it because the definition presented is not only "interesting", but must be considered as full and final.

1. Turing's definition

We do not intend to read Turing's paper according to its main mathematical issue, that of Hilbert's *Entscheidungsproblem*. This is an important problem for the historian of mathematics, but in what concerns the definition of informatics, we must rather search for the element responsible for its emergence. This element is named in the paper title, and is discussed since its first line:

> *The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means.* (TURING 1936, p. 230)

The "element" whose discussion is responsible for the emergence of informatics is the very object of mathematics, that is, *numbers*. Turing introduces a new set of numbers, the *computable numbers*, which are defined by the specific way they are calculated or "computed". For instance, the number *.25* is "computable" because it can be calculated by the division of *1* by *4*. Since it can be produced following one single rule, it belongs to Turing's proposed set. Another example: we need two rules to calculate the mean term between two numbers, "sum" and "division by *2*". Therefore, mean terms are also

computable numbers, as any number calculated using *1, 2*, or any other finite amount of rules. However, we must try to render this definition clearer. When Turing refers to the "expressions" of these numbers "as a decimal," he means, for instance, that the decimals of *1/3* may be expressed by a never-ending sequence of *3s: .333333...* Obviously we can never really finish calculating them, but Turing does not think these numbers in terms of whether their decimals are finite or not, but according to the amount of rules needed to calculate them.

The opening sentence of this paper offers what Turing called a *brief description* of computable numbers. This means that the idea of a finite amount of rules is not sufficient to fully define them. Something is missing, but what? At the end of the first paragraph of his paper Turing will refer to this missing element:

> *According to my definition, a number is computable if its decimal can be written down by a machine.* (TURING 1936, p. 230)

We now see that the missing element in the first and brief definition of computable numbers is: "a machine". However, with that we still do not have a full definition because we do not know what "machine" stands for. Turing only says "*a* machine", that is, a still not defined machine. This would have seemed impossible for the first readers of this paper because in 1936 no machine had ever been built to do what Turing proposed. Max Newman, a former professor of Turing, wrote in his 1955 short biography of Turing:

> *[…] it is difficult today to realize how bold an innovation it was to introduce talk about paper tapes and patterns punched in them, into discussions of the foundations of mathematics.* (NEWMAN 1955, p. 256)

It was in Newman's 1935 course on the Foundations of Mathematics in Cambridge that Turing was introduced to Hilbert's decision problem.[1] Newman read a draft of "On Computable Numbers…" in the spring of 1936 and when, 20 years later, he depicted Turing's paper as a "bold […] innovation", he was certainly not referring only to his own impressions, but to what any trained mathematician would have expected mathematics to be. Machines could be

1 PETZOLD 2008, p. 60: "In the spring term of 1935, Turing took a Foundations of Mathematics course given by Maxwell Herman Alexander Newman [...]. Also covered in Newman's course was the unresolved Entscheidungsproblem".

accepted as practical applications of mathematical procedures (just like the abacus or Pascal's adding machine), but they were not supposed to define mathematics itself.

The definition of computer brought about by Turing is a "bold […] innovation" because this machine does not belong to the history of traditional machines. A machine is a tool that allows us to do things that otherwise would be too hard or too repetitive. Machines expand our physical capabilities because they can help us level heavy weights, count time precisely, move fast and so on. At a first sight, this traditional understanding seems to apply also to computers because the accepted discourse about them stresses their incredible power to compute more data than any human being could ever do, and that they do it incredibly faster than us. Should not computers be defined according to the possibility of overcoming the human limits of calculus?

However, if we read Turing's paper carefully, the fact that real computing machines may be faster than us has no relationship to the definition he proposed. Turing does not refer to a machine that would expand our limited powers, but as he states in the first paragraph, to a machine based on the same *limitations* that affect us:

> *We have said that the computable numbers are those whose decimals are calculable by finite means. This requires rather more explicit definition. […] For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited.* (TURING 1936, p. 231)

Computable numbers are those numbers whose decimals can be calculated by finite means. Now we see that those means shall be finite because "the human memory is necessarily limited".

2. The intuition behind Turing Machines

In the sequel Turing will describe this machine, which will be named in the following year by Alonzo Church a *Turing Machine*.[2] He defines it in a single paragraph, with only 14 sentences and less than 300 words. It may seem surprising that the definition of computers could have been created with such economic means. Let's see, then, how Turing does it:

2 CHURCH 1937, pp. 42-43: "a human calculator, provided with pencil and paper and explicit instructions, can be regarded as a kind of a Turing Machine."

> *We may compare a man in the process of computing a real number to a machine which…* (TURING 1936, p. 231)

The concept of a Turing Machine is not the result of an inductive process of comparison of different cases of computing — for instance: of schoolboys solving quizzes, of people changing money on the street, of players gambling cards, of mathematicians presenting formal proofs and so on —, neither is it the result of a pure deduction of what it would ideally mean *to calculate* —, but it stems rather from something in-between, as it is the result of a unique sort of observation. The reference to "a man in the process of computing" is not purely physical, but is a conceptualization of experience. We must observe not gamblers, schoolboys or trained mathematicians, but *a man*. We must observe what any person would have done in the specific situation that Turing proposes us to think. He asks us to consider the general features associated with writing numbers on a sheet of paper in order to solve a problem. We may observe then some very simple things, such as that this man uses a pencil to write, a rubber to erase, and that he may end calculation. It is by "observing" this simple and almost child-like situation that Turing proposed two out of the three key features of his machine: an infinite tape with a finite number of symbols, and the six processes of writing, reading, erasing, shifting left, shifting right and halting.

Size, materials, power source or any other physical features have nothing to do with this machine. As it is not physical, it is also not time-bounded, and it is therefore a theoretical concept. However, it may seem that we cannot observe what is decisive in the situation chosen by Turing because we cannot know what this man thinks to himself... isn't mathematics about thinking? How can Turing propose his machine if we cannot observe the subjective "states of mind" of this man? Well, if he intended to magically gain access to someone's subjective mental states, he would not have bothered to propose this situation, for he could have simply analyzed someone doing mental calculation. Turing proposed a different situation — one that encompasses nonetheless a renewed sense of "states of mind".

In §9, when Turing justifies the previous definition of his machine, he refers explicitly to "states of mind" as not belonging to a subjective and mysterious realm, but as meaning simply: what has to be done now, after all that was read, written and erased up to this point. "States of mind" stand for the *rules* to be applied at each step of computation. They are called "configurations" in his machine, and they mean things like: at this step, if the symbol "X" is read, do "A" and go to "Step 11", otherwise do "B" and go to "Step 20".

Those rules are "expressed" through the same set of symbols used to express data on paper. That is why Turing may say in this section:

> *It is always possible for the computer [the man] to break off from his work, to go away and forget all about it, and later to come back and go on with it. If he does this he must leave a note of instructions (written in some standard form) explaining how the work is to be continued. This note is the counterpart of the "state of mind". (TURING 1936, p. 253)*

In modern philosophy and science, the metaphor of machines, and especially of clocks, was employed to explain the "inner processes" of our Spirit. Nevertheless, those metaphors are not restricted to this specific moment of culture as the words "*mechané*" and "*machina*" were related, in Greek and Latin, to "machination" and "inventiveness". Now we can see that Turing Machines have no relationship whatsoever to the metaphor of a machine-like Spirit, to what is ingenious in the mechanical operations of our mind. These machines are not witty, but follow simple rules; they do not explain "what is truly thinking" because *states of mind* or *machine configurations* concern what can be written down on a sheet of paper or on a tape.

This trait of Turing Machines explains that they are not related to more general questions such as "*What is Spirit?*" or "*What is a machine?*" Their concept is opened to *praxis*, that is, to *what has to be done*. So, if we go back to §9, to the section where Turing justifies his earlier definition of his machine in §1, we can find this bold statement:

> *The real question at issue is "What are the possible processes which can be carried out in computing a number?"* (TURING 1936, p. 249)

Informatics has nothing to do with the purely theoretical question "*What is…?*", but with the theoretic and practical question "*What has to be done…?*" Strangely enough, then, informatics has nothing to do with the question "*What is informatics?*". This may seem surprising, but Turing Machines are defined solely by "the possible processes which can be carried out in computing a number". The fact that their concept did not spring out of the traditional philosophical *what-question* has some consequences. We must see them further in the conclusion of this paper, but for the moment it is possible to say at least that, as the concept of informatics is not purely theoretical, it is not split from its physical implementations. For instance, the purely theoretical definition of the triangle as a closed geometric figure in which the sum of the

internal angles totals 180° is never matched by any real triangle. However, as the definition of informatics is based on the rules that real people do follow, then its implementations may perfectly match this concept. That is why Turing will explain, in a draft to a letter he wrote in 1948 to Church, that the problem of creating a specific machine rests on choosing:

> *conventions which [do] not restrict the essential generality of the machine.*[3]

The definition Turing proposed explains that the scientific status of informatics is not foreign to technological implementations.

3. Machine and téchne

According to what we have seen, a Turing machine is not a purely theoretical concept, but a theoretic-and-practical one. However, this does not mean that informatics would stand somewhere between science and technology, but it opens up a new possibility. The growing importance of technologies has no consequence in the definition proposed because a Turing Machine is neither time-related nor physical. That is why the title of this paper refers to a term close to *technology*, but which is a little bit surprising, the Greek word *téchne*. In Greek philosophy and science, *téchne* is a production that takes place in experience. However, this production is not dependent on experience.

In the beginning of *Metaphysics,* Aristotle discusses the type of knowledge proper to *téchne*. He considers it as close to the generality of *epistéme* (that is, to science) and, therefore, apart from *empeiría* (to experience).[4] While experience is knowledge of particulars, *téchne* is a production made possible by the knowledge of universals. Time, movement and other physical aspects define experience, whereas *téchne* and science are defined rather by principles.[5]

---

3 PETZOLD 2008, p. 102: "It was intended that the 'Turing machine' should always be the machine without attached conventions, and that all general theorems about machines should apply to this definition. [...] On the other hand when it was a question of describing particular machines a host of conventions became desirable. Clearly it was best to choose conventions which did not restrict the essential generality of the machine, but one was not called upon to establish any results to this effect. If one could find machines obeying the conventions and able to carry out the desired operations, that was enough."

4 ARISTOTLE, *Metaphysics* A 1, 981b24: "we consider that knowledge and proficiency belong to *téchne* rather than to experience".

5 Ibid, 981b26: "For the experienced know the fact, but not the wherefore; but the artists (*technítas*) know the wherefore and the cause."

However, we do not have to go back to Aristotle to verify a practical and theoretical sense of *téchne*, for we use today a Greek word created according to this twofold sense that is intimately related to our subject: "*arithmetic*", a word composed by *arithmós* (number) and *téchne*. Historically and conceptually considered, Turing Machines are in continuity with arithmetic as they calculate numbers, *computable numbers*.

In our context, what does it mean to say that arithmetic and informatics are related to *téchne*? It is easy to see that the *téchne* ofarithmetic has nothing to do with physical constraints, but concerns things like the notation used in calculus and the possible rules of transformation. Those notions are important to the *téchne* of informatics, but it depends also on something else. In his 1950 philosophical paper for *Mind Journal*, "Computing Machinery and Intelligence", Turing explained the mechanism of his machines by a seemingly unexpected approach:

> *Presumably the child-brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets. (Mechanism and writing are from our point of view almost synonymous.) Our hope is that there is so little mechanism in the child-brain that something like it can be easily programmed.* (TURING 1950, 59, 433-460, p. 456)

Turing compares the child-brain to an almost empty notebook. To say that this "little mechanism" can be "easily programmed" amounts to consider that there is not much writing to be done on it. How are we to understand then the comparison between mechanism and writing? Mechanism stands for the ideal deterministic world-view pursued by natural sciences, while writing is neither a type of knowledge nor a scientific paradigm, but is rather an inscription-based *téchne*. Turing is referring then to the traditional philosophical idea of a white sheet, or *tabula rasa*, as a free medium allowing for anything to be written on it. In spite of the new direction followed in this paper — that of Artificial Intelligence —, the idea that "mechanism and writing are [...] almost synonymous" does not stand for a new position on his machine. As a matter of fact, Turing is only making clearer something that was already at play in his 1936 paper. He said in §9, in the section where he justified the definition of the machine he proposed in §1:

> *Computing is normally done by writing certain symbols on paper.* (TURING 1936, p. 249)

Informatics is not a simple affair of arriving at the right answer, that is, it is not entirely enclosed by the formal rules guiding number calculation. This is only one of its possibilities, which nevertheless depends on something previous: on a *medium* where instructions and data can be written. As a consequence, writing, that is, writing-on-a-medium, may be considered as its defining *téchne*.

*

Which are the possible consequences of the proposed association of informatics to writing? Generally considered, "writing" is not equivalent to things like "language", or to whatever sort of "rules on symbols". Writing exceeds any possible formalization because it depends on an external medium.[6] In Turing's 1950 paper, this is rendered possible by the notebook, and in his 1936 "On Computable Numbers...", by the use of paper by the man, and by the tape by the machine. This would mean, then, that the writing *téchne* of a Turing Machine would not be entirely determined by the formalization of its table of instructions, that is, by the rules allowing to pass from a given state to another until a final halting state. This would be the case if Turing's paper dealt only with what that can be formalized, but, as we know, this table is only one of the elements of this machine. The first and foremost in Turing's *intuitive* approach is not the table of instructions, but the medium of writing: the sheet of paper. That is why the human operator may write his "state of mind" on it, or that a Universal Turing Machine shall read its program on the tape.[7] What is very interesting in Turing's definition, but not frequently stressed, is that someone, *external to the machine*, must previously write data and instructions on it; he must inscribe them in order for it to be a Turing Machine.

In his 1936 paper, Turing built four different versions of his machine, such as the Universal machine or those writing the sequences *.010101…*[8] or

6 LASSÈGUE & LONGO 2012 discusses this same passage of Turing's 1950 paper but from a different perspective. They take writing as a *system of writing*, while we ask here for gesture that requires a medium to write a symbol on it.

7 TURING 1936, p. 241-242: "It is possible to invent a single machine which can be used to compute any computable sequence. If this machine *U* is supplied with a tape on the beginning of which is written the S.D of some computing machine *M*, then *U* will compute the same sequence as *M*."

8 Ibid, ibidem, p. 233: "A machine can be constructed to compute the sequence 010101 ...."

.01011011101111…[9]. The content of the table of instructions is evidently different in each version, but we have, nonetheless, the same general machine. We can formalize what is done in each one, but we cannot formalize the fact that an empty Turing Machine simply waits for data and instructions to be written on it, nor that a machine programmed to solve one problem may be modified to solve another one at will.

4. Conclusion

When we talk about machines, we refer to the work produced by a finished machine. To this production, the specific technology employed is important. This is what explains the difference in nature, for instance, between a mechanical and an atomic clock. However, the definition of a Turing Machine is not that of a finished machine because what is inscribed on it can be changed at will. That is, this general machine does this now, but can solve another problem later. It is this state of *being-conceptually-unfinished* that allows for the diversity of its many implementations — if only they do not "restrict the essential generality of the machine". It is this also that explains its openness to any sort of technological implementation. This openness is more decisive than all the amazing future possibilities continually associated to informatics. These are responsible for impressive sociological transformations, but they do not affect the extremely simple concept Turing proposed. Any new technology, or any scientific proposition, could only overcome Turing's definition if they consciously searched to attack its founding element.

Turing began his paper proposing the set of computable numbers. These numbers are said "computable" because they are associated to a machine partaking our limitations. He proposed a machine bound to our incapacity of following infinite rules, but he could have proposed a machine that would compute by *infinite means* in order to further human memory. Had he done differently, the set of "infinitely computable numbers" could be inquired to be computable or not; however, only a new sort of machine could maybe prove something that would not be accessible for us. These machines would, then, effectively do what we cannot do. This means that the concept of informatics found in Turing's 1936 paper should be taken as final. This would not be the case only by the creation of new machines that would set us apart from our

9 Ibid, ibidem, p. 234: "As a slightly more difficult example we can construct a machine to compute the sequence 00101101110111110111111... "

historical relation to numbers — to the fact that we have always considered as our task to answer the question: *how can we calculate it?*

References

ARISTOTLE, *Metaphysics. Books I-IX* (Loeb Classical Library), translated by Hugh Tredennick. CAMBRIDGE, MA, Harvard University Press; London, William Heinemann Ltd. 1933, 1989.

CHURCH, Alonzo. "Review: On Computable Numbers, with an Application to the Entscheidungsproblem by A. M. Turing", *The Journal of Symbolic Logic*, Vol. 2, No. 1 (Mar., 1937), pp. 42-43.

LASSÈGUE, Jean & LONGO, Giuseppe. "What is Turing's Comparison between Mechanism and Writing Worth?" in: Cooper, S. Barry; Dawar, Anuj; Löwe, Benedikt (Eds.) *How the World Computes. Proceedings of Turing Centenary Conference and 8th Conference on Computability in Europe,* CiE 2012, Cambridge, UK, June 18-23, 2012.

NEWMAN, M. H. A. "Alan Mathison Turing. 1912-1954", *Biographical Memoirs of Fellows of Royal Society.*1955 1, 253-263.

PETZOLD, Jack. *The Annotated Turing: A Guided Tour through Alan Turing's Historic Paper on Computability and the Turing Machine*. Indianapolis: Wiley Publishing, 2008.

TURING, Alan. "On Computable Numbers, with an Application to the *Entscheidungsproblem*", *Proceedings of the London Mathematical Society*, Series 2, 42 (1936), pp 230 - 265.

_____. "Computing Machinery and Intelligence", *Mind*, New Series, Vol. 59, No. 236 (Oct., 1950), pp. 433-460.