https://doi.org/10.17163/soph.n39.2025.03

The Feasibility of Turing's Child-Machine

La factibilidad del niño-máquina de Turing

Rodrigo Alfonso González Fernández[*]
Universidad de Chile, Santiago de Chile, Chile
rodgonza@uchile.cl
http://orcid.org/0000-0001-9693-0541

Suggested citation: González Fernández, Rodrigo Alfonso. (2025). The Feasibility of Turing's Child-Machine. *Sophia, Colección de Filosofía de la Educación*, (39), pp. 111-133.

Abstract

According to the philosophers of Artificial Intelligence (AI), Turing Machines and the Imitation Game are the most important concepts proposed by Alan Turing. The Child-Machine Project, which projects learning machines via digital computers, is less known, although it is no less important. According to Turing's project, a programmed machine needs to be a Child-Machine to turn into an adult mind, one that understands, judges, and distinguishes. In this article, I argue that Turing's desideratum is not realizable only with algorithms. In the first section, I introduce the problem, while in the second I briefly analyze concepts such as algorithms, Turing Machines, and their relation. In the third section, I deal with Machine Intelligence and the Child-Machine Project. In the fourth section, I look at a form of understanding which is the basis of the Chinese Room Argument: introspection and reflective thinking, two factors that enable the process by which results are revised. In the fifth section, I analyze why those processes of revision are the stumbling block of classical AI or GOFAI; as I argue, introspection and reflective thinking are the cognitive faculties that prevent the child-machine from becoming a "thinking adult mind".

Keywords

Child-Machine, Thinking Adult Mind, Turing Machines, Algorithms, Introspection, Machine Intelligence.

[*] Associate professor at the Center for Cognitive Studies and the Department of Philosophy, Faculty of Philosophy and Humanities at University of Chile. His research interests are philosophy of mind and artificial intelligence, social ontology, and social epistemology. He has published a book entitled Experimentos mentales y filosofías de sillón: desafíos, límites, críticas (Mind Experiments and Armchair Philosophies: Challenges, Limits, Criticisms) and articles in journals such as Southern Journal of Philosophy, AI and Society, Anales del Seminario de Historia de la Filosofía, Isegoría, Aurora, UNISINOS, among others. Google Académico: https://scholar.google.cl/citations?user=YBcwekAAAAAJ&hl=es
Índice h: 8

Resumen

La máquina de Turing (MT) y el controvertido "juego de la imitación" son los aportes más reconocidos de Alan Turing en la filosofía de la inteligencia artificial (IA). El proyecto del niño-máquina, directamente ligado al aprendizaje de máquinas programadas o computadores digitales, no es tan reconocido, aunque no es menos importante. Según dicho proyecto, una máquina programada debe aprender como un niño, si ha de convertirse en una "mente adulta pensante", esto es, que juzgue, que entienda, que distinga. En este artículo se muestra que ese desiderátum de Turing no es realizable mediante algoritmos. Mientras que en la primera sección se introduce al problema, en la segunda se da un breve recuento histórico de los algoritmos, las máquinas de Turing y de su relación. En la tercera sección se aborda el concepto *machine intelligence* y el proyecto del niño-máquina. En la cuarta sección se muestra que una forma de entendimiento ("la habitación china" de Searle) da lugar a la introspección y reflexión crítica, que no son reductibles al funcionamiento de programas. Finalmente, en la quinta sección se argumenta que el proceso de la introspección y la reflexión crítica, son la "piedra de tope" de la IA clásica; en efecto, es la ausencia de ambas capacidades lo que impide que el niño-MT se convierta en una mente adulta pensante.

Palabras clave

Niño-máquina, mente adulta pensante, máquinas de Turing, algoritmos, introspección, inteligencia de máquina.

Our problem, then, is how to program
a machine to imitate the brain to make it think.
Alan Turing, BBC interview (1951)

Education is not the learning of facts, but the training of the mind to think.
Albert Einstein

Introduction

Artificial intelligence (AI) is a discipline that aims at creating programmed machines capable of imitating human intelligence. It is, then, an anthropocentric approach to intelligence. An interesting issue, linked to the discipline, is that Turing, who is considered one of the creators of AI, devised, in addition to the machine named after him, a method to establish whether it is justified to attribute mental states to digital computers. This method is based on a game, the famous and controversial «imitation game», described in detail below. But that was not Turing's only contribution to AI. Another that is less discussed is the child-machine project, i.e., the proposal that a computer program should be capable of learning just like a child. Such a project represents a remarkable challenge for AI, because it consists of projecting a program that learns like a child to eventually become a *thinking adult mind*. In what sense «thinking»? In that of

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

understanding and judging the true and the false, the clear and the ambiguous, and the solvable from the unsolvable.

In view of Turing's optimism, the aim of this article is to examine the following problem: can the child-machine project be carried out? But the question is, why could it not be carried out? As Turing himself points out, the child-machine must become an adult mind through learning. And this, as is well known, includes a series of key faculties for the development of intelligence: memory, reasoning, introspection, critical reflection, etc. It is precisely these last two capacities that represent a crucial stumbling block for Turing's project, as examined here. Indeed, the idea defended in this paper is that the mind's capacity for introspection and critical reflection *is not reducible* to the functioning of computer programs, which are only based on algorithms with automatic information processing. Precisely, the proposed argument is that, since the algorithms are mechanical and automatic, and do not require any *insight* or introspection, they do not allow for critical reflection, which is key in a thinking adult mind. This problem is a current and controversial issue in AI, due to *deep learning*. For this reason, a conceptual analysis method is used to test whether this type of AI with its machine learning achieves *insights* and critical reflection, which are key for an adult mind to learn effectively.

The article is divided into five sections. The first section gives a brief historical account of algorithms and Turing machines; it is intended to give readers a thorough understanding of the implications of the algorithmic operation of a program. The second revolves around two problems: on the one hand, how Turing elaborated the concept of *machine intelligence*, which is fundamental for understanding classical AI and *deep learning*; on the other hand, how his proposed method for testing mental states in programmed machines became a learning project based on algorithms: the child-machine project. The third part deals with a challenge to classical - or strong, in Searle's terms (1980, 1990) - AI based on a form of understanding: linguistics; as argued on the basis of the counterexample of «the Chinese room», introspection is not reducible to the operation of programs, on the contrary, they are not capable of generating it. Fourth, the idea is further developed as to why this capability, and the critical reflection to which it gives rise, are the *cornerstone* of classical AI; indeed, without introspection and critical reflection, the mind is not capable of revising the meaning of the rules that are automatically followed, so as to actually learn. Finally, the last section examines the most important conclusions of the analysis carried out in the paper.

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

From algorithms to Turing machines

The imitation of intelligent behavior is the central goal of classical AI. By classical I mean GOFAI (*good old fashioned artificial intelligence*)[1], AI that imitates human intelligence with the paradigm of rules and representations in mind. As Marvin Minsky would say: the goal of AI is the creation of machines programmed to perform tasks that require the same intelligence as if they were done by humans. Such tasks include simple activities such as playing checkers, or more complex ones, such as COVID-19 detection. In a way, GOFAI assumes that all problems can be approached algorithmically, i.e., that they can be solved by a set of finite steps, one of them being recursive. Therefore, classical AI builds machines that are able to mimic human intelligent behavior, the linguistic one, through algorithmic processing. But what are algorithms, a concept in vogue in our time?

Algorithms, despite their prevalence in today's world, are not new. They were popularized by the Persian mathematician Abu Ja'Far Mohammed ibn Mûsâ al-Khowâzarim around 825 A.D. (Penrose, 1989, pp. 41-44), but they were known much earlier. For example, Euclid's algorithm, which consists of a set of rules for finding the greatest common denominator (GCM) of two integers. Given this problem, the algorithm has finite rules and steps, with the third step being recursive:

i. Divide number and divisor, noting result and remainder (R);
ii. If R = 0, *halt*;
iii. If R ≠ 0, take previous divisor and remainder to execute step 1.

We can apply an example of this process with the numbers 99 and 15:

Number	Number divisor	Result	Remainder
99	15	6	9
15	9	1	6
9	6	1	3
6	3	2	0

Then, the algorithm yields the following result: the GCM between 99 and 15 is the integer 3. Just like that, a human can *automatically operate* with Euclid's algorithm (I return to the importance of Euclid's algorithm in the last section).

Another concept that is linked to that of algorithm is that of «recursive decomposition», i.e., the simplification of a complex operation

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

into mechanical steps. For example, one can decompose multiplication recursively like this, into addition and subtraction, respectively (Block, 1990, p. 256):

M x N = A
The rules of this new algorithm are as follows:
i. Add 1 times M to A and subtract 1 to N;
ii. Si N = 0, *halt*;
iii. If N ≠ 0, execute step i.

For example, 3 x 3 can be recursively decomposed into additions and subtractions, until a result is reached, algorithmically, that stops information processing.

M	x	N	=	A
3		3		0
3		2		3
3		1		6
3		0		9

Note that this new algorithm is a *multiplication machine*: it implements a program automatically and thus operates recursively by decomposing a more complex problem, multiplication, into simpler and more mechanical steps, addition and subtraction.

However, an algorithm is not exclusively about mathematics. To find the key to the lock on a key ring, an algorithm can be applied in which one recursively moves to the next key to the left if the one in one's hand does not fit. And so it can operate until it finds the key at which point the process stops. If one night, a drunken neighbor tried to find the key, he could execute the same algorithm and would do so despite being semi-conscious, because to operate algorithmically does not require consciousness, nor any *insight*. On the contrary, these are completely absent in serial processing.

Precisely, algorithms are machines because they are operated automatically, based on a mechanism that is recursive. It is worth noting that no introspective effort, *insight* or ingenuity is required to implement an algorithm. For example, the inebriated neighbor could search for the key while semi-conscious (*i.e.*, sleepwalking) and still implement the «find the key» algorithm mechanically.

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

The presence of algorithms is impressive today (*cf*. Kowalkiewicz, 2024). They are everywhere because they are the essence of what computing is and of the enormous facilities it provides. Such action processes information, transforming *inputs* into *outputs* automatically and mechanically, on the basis of a program that includes rules for the transformation expressed in conditional formulas of «If... then...». Therefore, running a program consists of the implementation of an algorithm capable of processing information. This is why the concepts of algorithm and computer program overlap. A program operates algorithmically, while an algorithm is a program for solving a problem.

Turing (1936) was —from a historical point of view—the one who clarified the definition of computation. He did so by introducing a definition in the form of an abstract machine, the so-called Turing machine (TM). In particular, TMs were postulated as a way of addressing the *entscheidungsproblem* posed by the mathematician David Hillbert. In view of this problem, we try to determine whether an algorithm X enables us to infer by means of a computable function all the theorems of first-order logic (formal language with quantifiers that reach variables of individuals, with predicates and functions whose arguments are constants or variables of individuals). This section does not delve into the *entscheidungsproblem* itself, suffice it to say that, thanks to it, Turing devised TMs as abstract devices that define precisely what it is to compute. Moreover, as a result of this problem, Turing aimed to find a method to characterize all computable functions.

Computing is, then, transforming an *input* into *output* on the basis of a set of rules or a program (and, therefore, an algorithm). In fact, a TM simply implements a computable function, which is linked to two fundamental notions: «machine» and «mechanical procedure» (we return to these notions below, in the light of the analysis of the *machine intelligence* concept). An TM is precisely a mechanical device that implements definable computational procedures by means of finite steps, i.e. it implements an algorithm. The TM has a discrete set of possible states that are finite in number (although potentially huge). This gives such a machine a very large (but finite) number of possible computations.

The *inputs* (*i. e.* numbers) are unlimited in size and the external storage capacity of the TM, where it writes, or the tape, is unlimited, as are the *outputs.* The TM, by definition, does not internalize external data or computations, but operates on the given data or computations in the most immediate operations. This idea is crucial to understand why a TM does not require *insights* and introspection (discussed again below).

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

As Penrose (1989) points out, the unlimited size of *inputs* and *outputs*, and the unlimited storage capacity of the tape, which is infinite, account for the highly idealized, abstract and mathematical character of a TM: «It is the unlimited nature of the *input*, the computational space, and the *output* that indicates that we are dealing with a mathematical idealization rather than something that could be constructed in practice» (p. 35) (own translation).

Thus, an TM usually characterizes as an infinite tape, with a head that reads and writes symbols according to a program (the rules of the algorithm). The head «remembers» some of the symbols: it is in an internal state $_{q1}$, $_{q2}$, $_{qn}$, etc., at a time $_{tn}$. Then, by reading a symbol (the *input*) and being in an internal state, it will generate an *output* depending on the program, which will lead the TM to delete or keep the symbol read, and move to a new state, if so stipulated by the rules.

For example, it is possible to make a TM add 3 + 2, with integers (Kim, 2006, pp. 125-128), given the following tape state (in this notation the number *n* is represented by the sequence of *n*, where each one occupies one square and only one):

#	#	1	1	1	+	1	1	#	#
		↑q_0							

It is important to note that all the instructions of a TM, which is discrete states[2] or clicks, are expressed in a machine table. In this case, the actions are described by the rules of the following program:

	q_0	q_1
1	1D q_0	# Halt
+	1D q_0	
#	# I q_1	

What the machine will do is that, given the internal state $_{q0}$ of the head, it will read *input* 1 and generate *output:* not write, move to the right and continue at $_{q0}$:

#	#	1	1	1	+	1	1	#	#
			↑q_0						

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

Then, it will read the next 1, not write, move to the right and continue at $_{q0}$:

#	#	1	1	1	+	1	1	#	#
				↑q_0					

Then, when reading the third 1, it will not write, it will move to the right (and continue in $_{q0}$), as we see below:

#	#	1	1	1	+	1	1	#	#
					↑q_0				

Subsequently, something «novel» will happen: it will read the *input* +, change it to a 1, move to the right and continue at $_{q0}$:

#	#	1	1	1	1	1	1	#	#
						↑q_0			

When it finds the next 1, it will leave it untouched, move to the right and continue at $_{q0}$. And it will do exactly the same with the next 1:

#	#	1	1	1	1	1	1	#	#
							↑q_0		

Again something new will happen when it reads # and is at $_{q0}$. Then, the *output* it will generate will be: it will not write, will move to the left and go to $_{q1}$:

#	#	1	1	1	1	1	1	#	#
								↑q_0	

Finally, being at q1 and reading 1, it will write # and *halt,* as illustrated*:*

#	#	1	1	1	1	1	#	#	#
							↑q_0		

Consequently, the TM processes algorithmically, based on the rules of a program, indicated in the machine table, with the rules of this one, which are expressed in conditionals of the form «if... then...». And

Sophia 39: 2025.

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

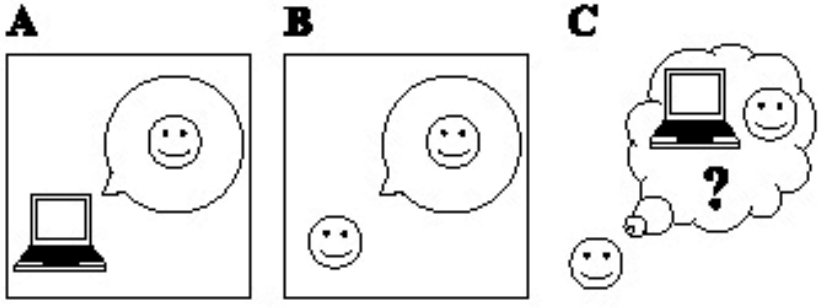
so, supposedly, it solves all problems requiring algorithmic solution. It is important to note that all problems, according to Turing, can have an algorithmic approach, even when learning is required. This is precisely what inspired him to conceive the child-machine project, as discussed in the next section. This project was born precisely as a consequence of eliminating the question «can machines think» (Turing, 1950).

Machine intelligence and Turing's machine-child project

Despite its presence today, the concept *machine intelligence* was originally conceived by Alan Turing in the 1950's. Bearing in mind the problem of linguistic understanding and how it represented a key to assign mental states to machines, Turing devised a way to evade the question «can machines think?». He did so for the following reason: the terms «think» and «machine» can generate dissent, in the face of people's alternative usages, which can lead to a sort of Gallup-type poll. In effect, Turing wants to move away from definitions, by leading to the uses of the concepts.

As a way of evading the analysis of the terms «thinking» and «machine», Turing proposed an empirical method to gather evidence of the existence of mental states in programmed machines: the imitation game. Although there are at least two versions of the game[3], tradition has interpreted it as a kind of standard version, as illustrated in Figure 1. In this version, the game consists of a programmed machine impersonating a person, fooling interrogators (Saygin *et al.*, 2000), while a person answers truthfully from a second room. The judges or interrogators, after five-minute rounds of questions, are tasked with discerning whether they are in the presence of a machine or a person, based on the typed answers alone.

Figure 1
Standard illustration of the «imitation game»



Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

Now Turing describes precisely: who can be interrogators and what kind of questions can be asked. Regarding the first point, he argues that the interrogators have to be "average", i.e., they cannot be specialists in computer science or similar, because this would give them an advantage with respect to discovering the machine. Regarding the second point, he specifies that the questions have to be average as well, so as not to give an obvious advantage to humans. Consequently, the interrogators and the machines must be average so as not to give humans an advantage in discovering the machines. But which ones can participate in the game?

Turing states that the machines in question are digital computers (programmed machines). These have the basic architecture of processing, drive and control in their operation. Moreover, they are digital because they operate by manipulating symbols according to the rules of a program. Hence, digital computers process information syntactically, given the algorithms they implement (we return to the importance of syntax in symbolic manipulation below).

Keeping in mind the imitation game -in its standard version- and the participants in it, Turing (1950) made a prediction:

> I believe that in about fifty years it will be possible to program a computer, with a memory capacity of[109], to participate in the Imitation Game so efficiently that a lay interrogator will have no more than a 70% chance of making the correct identification after five minutes of interrogation. I think the original question, can machines think, is too absurd for further analysis (p. 49).

Of course, Turing's prediction is as controversial as his test and is so for two important reasons. First, it fails to realize that the context of the prediction involves the philosophy of mind and AI. Second, the discipline of AI philosophy begins with a particular description by Babbage (in Swade, 2000) of one of his machines:

> Babbage speaks of *teaching the machine to foresee* [...] At other times he states that the machine *knows*. [...] The analogy between these acts and mental processes forced me to the figurative use of such terms. The use of these was found to be economical and expressive, and I prefer to continue to use them rather than substitute long circumlocutions (pp. 103-104) (own translation) (emphasis added).

Babbage doubts that his machine can be assigned mental states in an entirely literal way. Instead, he proposes that such adjudication is done by conceptual economy, or as a way of avoiding long circumlocutions. In this way, Turing's prediction is relativized in the field of philosophical

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

reflection, which attempts to examine and argue regarding the existence of mental states in programmed machines. But Turing (1950) faced two even more serious objections, as detailed below:

> Could the machine do anything different from what man does? This objection is very powerful, but we can say that, nevertheless, if a machine can be built to play the game of imitation successfully, we need not take it upon ourselves (p. 42) (own translation).

He then proceeds to ask the following question, with an objection that has important philosophical overtones because of the type of situation posed, clearly hypothetical: «We are not asking whether digital computers can perform well in the game, or whether there are in the present, but *whether there are imaginable computers that could perform well*» (p. 43) (own translation) (original emphasis).

As can be seen, Turing proposes a sort of thought experiment in relation to the intellectual capacity of digital computers. There must be *possible digital computers* that can pass the test. If, optimistically, it is assumed that there can be such digital computers, the question is: in what sense must they be intelligent in answering the questions?

After anticipating a number of possible objections to the prediction[4], Turing concludes that the best way to conceive of a digital computer capable of performing well in the imitation game is by learning. But not just any learning, but that of a child-machine. With this, Turing puts on the table that it is through education and learning that we will be able to have programmed machines that can *imitate intelligent human behavior*. In what sense of «imitate»?

A question that is important to clarify is the type of functionalism to which Turing finally ascribes (Putnam, 1967, 1968). In view of the game of imitation, it is not necessary to replicate the physical or biological properties of the brain, since the game allows differentiating mental capacities from such properties[5]. According to this type of TM functionalism, it is better to imitate the mind of a child, since it is like a notebook, with simple (programmable) mechanisms and blank pages. This is key to understanding how Turing argues that the mind can be mechanized and that any successful imitation of the machine-brain will replicate it and its intelligent capacity. Thus, the mind of a child-machine can be transformed into an adult mind and will do so in terms of how one who thinks, reasons and understands learns (Feldman, 2009, pp. 70-72).

Turing's (1950) words regarding his project of the machine-child are as follows:

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

> In the process of trying to imitate a human mind we are conditioned to think quite a bit about the processes it was subjected to in order to arrive at its present state. We can note three components:
>
> 1. The initial state of mind at birth;
>
> 2. The education it has received;
>
> 3. Other experience, not describable as education, that it has had.

Turing (1950) proposes, then, in the following way a specific type of learning, to imitate what an adult mind does. For that purpose, he even assumes that a child's mind has mechanisms that may be, by all accounts, algorithmic:

> Instead of trying to produce a program to simulate the human mind, why do not we rather try to simulate one that is like that of a child? If subjected to proper education, one would get an adult mind. Perhaps the child's brain is something like a notebook one buys at an office supply store. Little mechanisms and a lot of blank pages (p. 62) (own translation).

The question remains, however, whether the child-machine project is viable as Turing originally conceived it. Precisely, in the following sections it is shown that, since there are parts of learning that depend on introspection, the child-machine project is not feasible in the way Turing thought (*i. e.* only by means of algorithms). There are difficulties in principle for a child-machine to transform into an adult mind, which is precisely what Turing proposed to conduct the long-term AI project.

«The Chinese room» and its Cartesian puzzle

One issue worth emphasizing is that, in addition to the behaviorism that is attributed to it, the Turing test is underpinned by a kind of Cartesianism. This is because, as we know, Descartes (1994, pp. 112-113) proposed that there are two unequivocal signs of intelligence: the use of conventional linguistic signs and intelligent action, guided by reasons. In relation to the first point, the French philosopher states that only human beings are capable of using conventional signs, since animals are only capable of using natural signs. For example, John is able to say «I love Mary» or «Mary is loved by me», two sentences that mean the same thing. In contrast, animals are only able to react to stimuli in a natural way. If a dog's leg is pricked, it will howl in pain and it will do so because of its own nature, i.e., because of the disposition of its organs. On the other hand, John can give

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

reasons for why he loves Mary, but Bobby, John's dog, only acts according to the propensity of his organs when he howls in pain.

Turing inherits a Cartesian prejudice, namely, that the use and handling of language is a sign of intelligence (Descartes, 1994, pp. 112-113). In this sense, the game of imitation is nothing more than a sequel to Descartes' ideas regarding reason and intelligence. Indeed, the fact that the digital computer, with adequate programming, storage and speed, is capable of responding as a person would, deceiving interrogators, indicates that language or linguistic behavior is that which makes it possible to *verify the existence of intelligence*. Note that, although Turing argues that *there should only be possible computers* that pass the test, doing so indicates that we are in the presence of an acid test of intelligence, at least with respect to the existence of mental states in programmed machines.

But there is another Cartesian element that must be emphasized, and that is the question of linguistic understanding. This can only be detected internally. John Searle (1980) is emphatic when he speaks of his thought experiment against GOFAI, which he calls "strong AI" (p. 67). The Chinese room thought experiment is intended against the thesis that the mind is a computer program or *software*. In the room, Searle is a native English speaker, who speaks some degree of French and no Chinese. How does he know this? Internally: if an Englishman is presented, he will understand everything he says; if a Frenchman is presented, he will understand part of what he says; if a Chinese person is presented speaking to him, he will understand absolutely nothing. Consequently, it is examined and determined whether a language is understood internally, thanks to introspection (more on this point below). How could programmed machines understand a language, then?

Two GOFAI researchers, Roger Schank and Robert Abelson (1977), conceived of linguistic understanding in an anti-Cartesian way, i.e., by appealing to the recursive decomposition of elements. As will be recalled, this goes from the complex to the simpler, until a level of mechanical simplicity is reached. For example, what do you do when you have to understand a word? Understanding a word is possible because there are three clearly differentiated stages: that of bringing it in, that of matching it with a list to make it fit with some other word, and that of retrieving the associated syntactic and semantic information. It is important to note that these three levels are mechanical, because they bring in, collate and retrieve information, respectively. According to this model of understanding linguistically, there is no sort of Cartesian *self* that thinks and corro-

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

borates what the word means. On the contrary, there are only mechanical, algorithmic steps that process information.

Now, Schank and Abelson's theory can be extrapolated to understanding stories and answering questions about information that is not explicitly developed in them. Such a question is asked by a programmed computer: SAM (*script applier mechanism*)[6]. The system operates in a similar way to understanding a word: there are a number of *scripts* or stories stored in memory, there are stories that come in, these are matched with the first ones, and then questions can be answered about information that is not explicitly stated. One point worth noting is that an ideal version of SAM would certainly pass the Turing test. That is, depending on the *scripts*, and the stories, the computer might be able to provide information that is not directly alluded to in them, which certainly counts as intelligent behavior. However, Searle does not think that SAM understands stories and is intelligent. That would only occur if strong AI were true.

On the basis of what Schank and Abelson (1977) advocated, Searle (1980) is determined to show that strong AI is a false theory. One way to show the falsity of a theory is to ask what would happen if the mind operated in accordance with it, i.e., what would happen if the mind operated in accordance with the postulates of strong AI; consequently, he proposes to falsify it by indicating what would happen if the mind operated in accordance with this theoretical approach.

Thus, we return to the scenario of the thought experiment. As known, in Chinese there is no alphabet, but ideograms, i.e., pictorial representations of events, things, and so on. Chinese speakers know the meaning of ideograms by virtue of their form, which is essential to the thought experiment and to the fact that computing is manipulating symbols based on that form and rules. The speaker is locked in a room, which has an *input* slit, an *output* slit, a data bank of Chinese *scripts* (which he only sees as meaningless symbols) and a rule book for manipulating the symbols. Now, there are a number of native Chinese speakers outside the room sending ideograms through the *input* slit. The subject takes those ideograms, compares them to the data bank, and proceeds to manipulate those symbols by virtue of their shape, thanks to the rule book, which is written in English. The rulebook stipulates that if, for example, ideograms 81, 99 and 100 are together, the subject must send ideograms 1 and 7 through the *output* slit.

Of course, the subject has no idea what he is doing with the ideograms, except manipulating them. In fact, he only manipulates symbols syntactically by virtue of their form, and thanks to the rules of the book.

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

He does not understand, then, what the ideograms mean, let alone understand that native Chinese speakers are sending him stories and questions to get answers through the *output* slit. The speakers could even be insulting the subject of the experiment and the subject would not notice in the least. One thing is clear, given the Chinese room scenario, the individual has no linguistic understanding of Chinese whatsoever, nor are there any associated mental states. He just manipulates symbols without knowing what they mean and despite the fact that the Chinese speakers believe that there is another Chinese speaker locked in the room. In other words, the symbols are just forms manipulated by virtue of the rules of the book, in the case of Searle, who does not understand what all this symbolic manipulation is about. The attentive reader will recall that the scenario described is analogous to the Turing test, but there is an important difference with it. In the case of the imitation game, Turing's prediction is that in the year 2000 the average interrogator will have no more than a 70% chance of discovering the computer. However, in the Chinese room all the interrogators (*i. e.,* the native Chinese speakers outside the room) are fooled.

There are a number of objections to the Chinese room thought experiment[7]. Two are the most popular and seductive. The first is that Searle is only part of a system. The whole of it possesses linguistic understanding of Chinese, even though there is no clear *locus* of that understanding. Searle, then, cannot claim that the totality of the system has no linguistic understanding, since it is likely that the room, plus the slits, the data bank, and the rule book are capable of understanding Chinese. This philosopher defends himself against the system objection by emphasizing that if the room understood, all sorts of subsystems could have mental states without our knowing it. He even asserts that he could internalize all the elements of the room, such as the *input* slit, the *output* slit, the data bank and the rule book (which he could memorize). All symbolic processing could then be done internally. Given the importance of this point, I return to it below.

The second important objection is that of the robot. According to this objection, if there were a robot that causally anchored symbols in the environment through the use of transducers, it could understand the meaning of Chinese symbols. Clearly, this objection adds a novel element: that symbols have meaning to the extent that they are anchored in the environment. Searle answers this objection with a new thought experiment: he is now in the robot's head, receiving Chinese symbols from the transducers, and sending Chinese symbols to the robot's moving elements, so

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

as to produce the appropriate response. Surprisingly, the Chinese room scenario is replicated again, because it receives symbols whose meanings it does not know, and sends symbols, whose meanings it also ignores. Consequently, the robot does not seem a sufficiently convincing argument, at least as far as linguistic understanding is concerned.

In this regard, it is clear that both the system and the robot are based on a Cartesian element. Think of Searle's scenario internalizing the elements of the system. However, there is only one element that he cannot internalize: himself. Thanks to introspection he can realize that he understands English and not Chinese. In other words, if Searle were to internalize all the elements of the system, there would be only one that he could not internalize, namely himself, who is the one running the experiment. This is why the Chinese Room has, despite Searle, a Cartesian bias (Gonzalez, 2012). It has it because, like most mental experiments about the nature of the mind, it is introspection that is in charge of indicating in what way what is described by a theory is true or false. In the case of the Chinese room, introspection would show, according to Searle, in what way the strong AI is false[8].

Introspection cannot be reduced to the operation of an algorithm. This Cartesian element of introspection involves a bias that cannot be characterized algorithmically, as discussed below, and which is linked to critical reflection.

Introspection and critical reflection are the «stumbling blocks» of classical AI

In this final section, we show in what sense introspection cannot in principle be reduced to the operation of algorithms. If this is so, the child-TM learning project falters. It does so because there is a good portion of such learning that, analogous to linguistic understanding, depends on introspection, which leads to the realization that an algorithmic process is, for example, wrong to achieve a result. And the reverse is also true of algorithms: it is not necessary to have *insight* or introspection to reach a result. Recall Euclid's algorithm: to determine the MCD of 99 and 15 a series of steps were followed, in a mechanical way, until the result was reached, 3 is the MCD of both numbers. While it could be argued that the machine could understand that 3 is the MCD of 99 and 15, such an assumption would be doubtful if one assumes an additional element: that the machine could employ a faulty algorithm, with rules that lead to a *loop*, to recursion without stopping, when trying to find various MCDs

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

with wrong rules. In such a case, the machine could not be said to understand the result. Then, there is a crucial difference between understanding a problem and solution, and not doing so.

For example, if a programmed machine were instructed to follow finite rules to find an odd number by adding two odd numbers, the programmed machine would not find the solution to the problem, because the problem simply has no solution. Indeed, would the machine understand that the *loop* without stopping is the product of an algorithm that seeks the solution to a problem that does not have one? Discovering such a difficulty is part of a process involving introspection, which is necessary in order not to follow meaningless rules. In view of this difficulty, one could conclude that algorithms are too serial and linear, and that since they do not resort to a process of introspection, which facilitates critical reflection, they are not able to establish that there are some rule-followings that simply do not make sense.

The automatism of the algorithmic steps does not lead to any conscious psychological experience, as is the case with the Chinese room. In the case of 99 and 15, and their highest common denominator, there is no psychological experience in relation to mathematics. In fact, the algorithm could be followed by someone ignorant of mathematics, as well as by an expert in it. Neither would have any interesting psychological experience associated with the execution of the algorithm, and for this reason no conscious mental states are produced in any sense. Thus, no agent following the rules of Euclid's algorithm will have a conscious psychological experience associated with following those rules. The same, *mutatis mutandis*, is true of the algorithm for obtaining an odd number by adding two odd numbers. Just as there are no conscious states associated with the execution of Euclid's algorithm, there are no conscious states associated with the execution of a failed algorithm, which is key to the machine not understanding that the problem is nonsense.

As Penrose (1989, pp. 141-143) emphasizes, the *halting problem* is insoluble if there is no conscious introspection and experience associated with mathematical *insights* (mathematical intuitions) that indicate whether, for example, a mathematical problem is unsolvable. To some extent, classical AI pays a high price, because of the nature of the algorithms. Since they do not require *insights* or introspection, all the negative consequences follow in relation to solving problems whose solution does not have an algorithmic approach. In relation to what has been examined in this section, it turns out that the child-machine could not become an adult mind, because thanks to introspection, only introspection is able

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

to make an agent realize if an algorithm will stop. The TM-child, lacking introspection, will fail to realize and understand the non-solvability of some mathematical problems, which makes it unable to become a conscious adult mind, i.e., capable of critical reflection and understanding in the face of the meaninglessness of a problem.

Such critical reflection follows a pattern very similar to Socratic reflection, key to certain educational processes, as it is a constant search for truth through irony and *elenchos* (*i. e.*, refutational questions that make sense only in the context of certain statements). Just as critical reflection in the case of algorithms, *elenchos* illuminates the typical critical reflection of an adult, who doubts, who understands, who affirms, and who is key to applying the Socratic method, which questions the meaning of some statements. Contrary to what Corballis (2007) believes, with his thesis about recursion as a unique characteristic of the human species, the educational process that leads to a thinking adult mind cannot rely solely on such recursion. It leads to automatic and serial cognition, which rightly lacks critical reflection, and the typical process of the adult mind that doubts and returns to itself through introspection. In fact, this cognitive process can be related to the imaginative, interpretative and generative capacity of thinking, which, despite Corballis, is not the product of a mechanical process. Thinking is, in a Socratic sense, questioning, imagining and interpreting contextually. Precisely in relation to the introspection and critical reflection typical of an adult thinking mind advocated here, Bailin and Siegel (2002) emphasize that:

> Thinking that is primarily driven by the evaluation or critique of ideas or products *is not algorithmic, but has a generative and imaginative component.* The application of criteria is not a mechanical process, but involves the interpretation of circumstances, and an imaginative judgment regarding the applicability of criteria in different circumstances, and whether the criteria are satisfied (p. 187) (own translation) (emphasis added).

Certainly, the so-called Socratic method is intimately linked to this generative, imaginative dimension of thought, which consists precisely in not following rules automatically, and in questioning the meaning or meaninglessness of some of them. In fact, from the point of view of empirical evidence, there are studies that suggest that the critical reflection typical of adults is key in relation to positive results in higher education. Despite Turing, there is empirical evidence showing how humans become thinking adult minds, and how this unique feature of the human mind enables academic achievement. In fact, there are studies that rightly ex-

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

plore the relationship between reflective thinking, critical thinking, self-monitoring, and positive outcomes in college education (Ghanizadeh, 2017). Another study comes to similar conclusions: critical, reflective, and creative thinking is critical to academic achievement (Akpur, 2020).

In summary, algorithms enhance learning, but they also have important limitations. By not requiring introspection, they make the associated problem solving simply non-existent. An old and simple algorithm, such as Euclid's, shows precisely how introspection is completely absent in information processing, with all the negative consequences that follow. The questions are: Would Turing have the same enthusiasm for the child-machine learning project if he had realized the limitation of algorithms? Would he have defended the idea that the child-machine can learn based on them and become an adult-mind, without the capacity for critical reflection? An intuitive answer to both questions is that the philosopher and mathematician might not have argued so passionately for the possibility that programmed machines can learn without developing a kind of Socratic reflection. Or, at least, that a child-machine can eventually become a genuine adult mind by learning *only with the help of algorithms*, which is the purpose of his controversial AI learning project.

Conclusion

This paper has developed an unavoidable problem for classical AI or GOFAI: that algorithmic processing leaves out introspection and critical reflection. In particular, it has been examined how negative consequences follow from the absence of both of these cognitive processes, which are key for a child's mind to transform into a truly thinking adult mind.

To show the problems of classical AI or GOFAI, the operation of simple algorithms, such as Euclid's, or of some that do not have detention, such as the aforementioned search for an odd number by adding two odd numbers, has been described. In both cases, what was stated above is confirmed: that the absence of introspection and critical reflection represents a stumbling block for genuine learning, especially in the child-TM project mode. Indeed, this lack limits what he projects, particularly in terms of how a mind can fail to follow rules automatically.

To reach such a conclusion, five sections have been developed. The first consisted of presenting the problem to be examined in the essay. The second, on the other hand, gave a historical account of algorithms and their relation to TMs. The third section addressed the concept of *machine*

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

learning, as Turing understands it, and how this concept relates to the child-machine learning project. The fourth section discussed the Chinese room problem and how the Chinese room leaves out understanding, introspection and critical reflection, all key cognitive processes for learning. The fifth and final section addressed why these processes are the cornerstone of classical AI. In particular, it was examined in what sense the algorithms leave out introspection, which has negative consequences for the child-TM learning project. Indeed, such a project is limited to pure information processing and, consequently, difficulties for the child-machine to transform into a thinking adult mind.

Following similar guidelines to Weizenbaum (1984) and Smith (2019), who explore the shortcomings of AI in terms of developing the ability to judge without ethical commitment and responsible action, it has been criticized how the child-TM project responds to *an exclusively algorithmic conception of intelligence*, i.e., which only relies on the operation of a programmable machine that calculates and processes information by manipulating symbols through the rules of a program. Because of its algorithmic character, introspection is left out, and difficulties such as the *halting problem* remain unsolved. All problems that are solved by appealing to introspection and reflective, imaginative and critical thinking are also left out. It is worth asking, then, can the adult thinking mind be mechanizable in purely algorithmic terms, as Turing claims? Can such a thinking mind be reached by a child who only learns on the basis of algorithms? Here the answer has been negative: the child-machine project is not feasible as Turing conceived it, i.e., only in terms of algorithms, since these are incapable of generating introspection and reflective-critical-imaginative thinking. The algorithmic learning project lacks these fundamental processes in education, despite philosopher-mathematician Alan Turing's determined defense of his machine-child.

Notes

1 In this paper, GOFAI and classical AI are used interchangeably. Later, it refers to how cognitivism is embodied in what Searle calls strong AI. In a way, all these terms mean the same thing, because the basic assumption is that the mind is, as a matter of fact, a programmed serial and algorithmic processing computer. That is, GOFAI, classical AI, cognitivism, and strong AI all rely on one theory, namely, the computational theory of mind (as described by Block, 1990).

2 By discrete states it is meant that a TM cannot be in degrees, but in limited, precise, definable states. For example, 1.5 is graded between 1 and 2, whereas 0 and 1 are discrete states.

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

3 For a more thorough examination (*cf.* Gonzalez, 2015). This essay shows why the identification of the gender of the participants is not casual in the Imitation Game: to have the female linguistic behavior does not require having the female brain, which shows the crucial difference between the physical properties of the brain and its intellectual capabilities.

4 Turing advances nine possible objections to his Imitation Game: the theological, the heads-in-the-sand, the mathematical, the consciousness argument, the multiple disabilities argument, the Lady Lovelace argument, the continuity of the nervous system, the informality of behavior, and the extrasensory perception argument (Turing, 1950, pp. 49-60). For reasons of space only these objections are mentioned here.

5 Turing, despite the comments of some (*i. e.* Block, 1990, pp. 248-253) is not a behaviorist, but a functionalist. One can review how TM functionalism is antibiological (Putnam, 1967, 1968; Block, 1990, 1995; Heil, 2004; Gonzalez, 2011).

6 For a review of how SAM works one can review the works of another *chatbot*: ELIZA (Weizenbaum, 1984).

7 *Cf.* Block (1995) and Preston and Bishop (2002).

8 For reasons of space we only record that, given the objections to the thought experiment, the Chinese room raises reasonable doubt that strong AI is true. But it is less clear that Searle's thought experiment definitively disproves strong AI. Thus, downgrading the result of the thought experiment to reasonable doubt rather than refutation saves the objections of the system, the robot, among many others.

Bibliography

AKPUR, Uğur
2020 Critical, reflective, creative thinking and their reflections on academic achievement. *Thinking Skills and Creativity*, *37*. https://doi.org/10.1016/j.tsc.2020.100683

BAILIN, Sharon, & HARVEY, Siegel
2003 Critical thinking. En N. Blake, P. Smeyers, R. Smith, & P. Standish (eds.), *The Blackwell Guide to the Philosophy of Education* (pp. 181-192). Oxford: Blackwell.

BLOCK, Ned
1990 The computer model of the mind. En D. N. Osherson y E. E. Smith (eds.), *Thinking: An Invitation to Cognitive Science* (vol. 3, pp. 247-289). Cambridge, MA: MIT Press.

BLOCK, Ned
1995 The mind as software of the brain. En J. Heil (ed.), *Philosophy of Mind: A Guide and Anthology* (pp. 267-274) Oxford: OUP.

CORBALLIS, Michael
2007 The Uniqueness of human recursive thinking: The ability to think about thinking may be the critical attribute that distinguishes us from all other species. *American Scientist*, *95* (3), 240-248. https://doi.org/10.1511/2007.65.240

DESCARTES, René
1994 *Discurso del método*. Madrid: Alianza.

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

FELDMAN, Richard
2009 Thinking, reasoning and education. En H. Siegel (ed.), *The Oxford Handbook of Philosophy and Education* (pp. 67-82). Oxford: OUP.
GHANIZADEH, Afsaneh
2017 The interplay between reflective thinking, critical thinking, self-monitoring, and academic achievement in higher education. *Higher Education*, *74*(1), 101-114. https://doi.org/10.1007/s10734-016-0031-y
GONZÁLEZ, Rodrigo
2011 Máquinas sin engranajes, cuerpos sin mente. *Revista de Filosofía Universidad de Chile*, *67*, 183-200. http://dx.doi.org/10.4067/S0718-43602011000100012
2012 La pieza china: un experimento mental con sesgo cartesiano. *Revista Chilena de Neuropsicología*, *7*(1), 1-6. https://bit.ly/4kZC7Wb
2015 ¿Importa la determinación del sexo en el test de Turing? *Aurora*, *27*(40), 277-295. http://dx.doi.org/10.7213/aurora.27.040.AO02
KIM, Jaegwon
2006 *Philosophy of Mind*. Cambridge, MA: Perseus Books.
KOWALKIEWICZ, Marek
2024 *The Economy of Algorithms: AI and the Rise of the Digital Minions*. Bristol: Bristol University Press. https://doi.org/10.2307/jj.10354686.14
HEIL, John
2004 Functionalism. En *Philosophy of Mind: A Guide and Anthology* (pp. 139-49). Oxford: OUP.
PENROSE, Roger
1989 *The Emperor's New Mind*. Oxford: OUP.
PRESTON, John, & BISHOP, Michael
2002 *Views into the Chinese Room*. Oxford: OUP.
PUTNAM, Hilary
1967 Psychological predicates. En J. Heil (ed.), *Philosophy of Mind: A Guide and Anthology* (pp. 158-167). Oxford: OUP.
1968 Brains and behaviour. En D. Chalmers (ed.), *Philosophy of Mind: Classical and Contemporary Readings* (pp. 45-54). Oxford: OUP.
SAYGIN, Ayse Pinar, CICEKLI, Ilyas, & AKMAN, Varol
2000 Turing test: 50 years later. En J. H. Moor (ed.), *The Turing test: The Elusive Standard of Artificial Intelligence* (pp. 23-78). Dordrecht: Kluwer Academic Publishers.
SCHANK, Roger, & ABELSON, Robert
1977 *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Erlbaum.
SEARLE, John
1980 Minds, brains and programs. *The Behavioral and Brain Sciences*, *3*, 417-457. https://bit.ly/45XNcTg
1990 Is the brain's mind a computer program? *Scientific American*, 1 de enero, 26-31. https://bit.ly/3FS93kA
SMITH, Brian
2019 *The Promise of Artificial Intelligence: Reckoning and Judgement*. Cambridge, MA: MIT Press.
SWADE, Doron
2000 *The Difference Engine: Charles Babbage and the Quest to build the First Computer*. Londres: Penguin.

Sophia 39: 2025.

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.

TURING, Alan
1936 On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, *s2-42*(1), 230-265. https://doi.org/10.1112/plms/s2-42.1.230
1950 Computing intelligence and machinery. *Mind*, *59*(236), 433-60. https://bit.ly/3ZXWafy
WEIZENBAUM, Joseph
1984 *Computer Power and Human Reason: From Judgement to Calculation*. Harmondsworth: Pelican.

Acknowledgments

I thank the anonymous reviewers. I also thank Felipe Morales Carbonell, Felipe Álvarez, Felipe Tobar and María Soledad Krause for the discussion of this article. This research was funded by ANID Project FONDECYT 1230128 Mistrust: a Causal Factor of Searlean Institutional Crises.

Statement of Authorship - CRediT Taxonomy	
Author/s	**Contributions**
Rodrigo Alfonso González Fernández	Since this is a single-authored paper, the total contribution corresponds to the same author. The content presented in the article is the sole responsibility of the author.

Declaration of use of artificial intelligence
Rodrigo Alfonso González Fernández **DECLARES** that the elaboration of the article entitled "The feasibility of the Turing child-machine" was not supported by artificial intelligence (AI).

Date of receipt: July 14, 2021
Date of revision: September 15, 2021
Date of approval: April 20, 2025
Date of publication: July 15, 2025

Print ISSN:1390-3861 / Electronic ISSN: 1390-8626, pp. 111-133.