Minds & Machines (2010) 20:203–212
DOI 10.1007/s11023-010-9193-z

Can Machines Think? An Old Question Reformulated

Achim Hoffmann

Received: 11 June 2009 / Accepted: 23 May 2010 / Published online: 13 June 2010

Abstract This paper revisits the often debated question *Can machines think?* It is argued that the usual identification of machines with the notion of algorithm has been both counter-intuitive and counter-productive. This is based on the fact that the notion of algorithm just requires an algorithm to contain a finite but arbitrary number of rules. It is argued that intuitively people tend to think of an algorithm to have a rather limited number of rules. The paper will further propose a modification of the above mentioned explication of the notion of machines by quantifying the length of an algorithm. Based on that it appears possible to reconcile the opposing views on the topic, which people have been arguing about for more than half a century.

Keywords AI debate · Turing Test · Kolmogorov complexity · Algorithmic

Introduction

Can machines think? Is intelligence algorithmic? Are we humans, after all, just machines that follow a set of rules? These and similar questions express the wonderment of people whether it is possible to build a fully-fledged Artificial Intelligence system which at least behaves just as intelligent humans do.

The question on the general possibility of Artificial Intelligence (in short AI) attracted considerable interest even before Artificial Intelligence's founding Dartmouth conference was held in 1956. Numerous articles have been published in both the philosophical literature as well as the rather technical literature on Artificial Intelligence.

A. Hoffmann (✉)
School of Computer Science and Engineering, The University of New South Wales,
Sydney, NSW 2052, Australia
e-mail: achim@cse.unsw.edu.au

In 1950 Alan Turing discussed in his famous *Mind* paper arguments against the possibility of Artificial Intelligence (Turing 1950). Turing basically rejected all the arguments he mentioned. Reason for his rejection was the fact that the arguments could be applied to other human beings as well. The *behavior* of human beings, or intelligent systems in general, appeared to Turing to be the only, or at least the only relevant, observable aspect of intelligence. Consequently, instead of accepting any of the arguments he mentioned, he proposed an *empirical* test whether a machine *behaves* intelligently or not. The test, in which a human judge is supposed to differentiate between a human and a machine based on their respective behavior which can only be observed through communication via a tele-type became later known as the *Turing Test*. Since 1990 an actual contest is being held regularly and known as the Loebner contest in which a group of machines try to get mistaken for humans by the human judges, i.e., the machines try to pass the Turing Test. See e.g. (Floridi et al. 2009] for a discussion of the 2008 Loebner contest.

Interestingly, it has been argued quite vehemently that the notion of algorithm is too limited to allow the development of a machine that would be as intelligent as humans, see e.g. Lucas (Lucas 1961) or Penrose (Penrose 1989) for prominent examples. Various machine models have been suggested to be super-Turing machines, i.e., which go beyond the limitations of Turing machines, which are usually considered to explicate the notion of algorithm, see e.g. Shagrir (1997, 2002) for a discussion. Copeland (2002) provides an excellent review of ideas on super-Turing computation. Or see (Sekanina 2007) for an argument why evolved systems, such as humans or animals, may not be regular computing devices, simply, since we don't really know and understand what physical processes underlie their behavior. Piccinini (2007) reviews the debate on the idea that natural neural systems may be computational systems that are more powerful than a Turing machine. However, Piccinini suggests that natural neural systems might just not perform computations at all—whether in the sense of a Turing machine or some other super-computation sense.

Without intending to review or even debate those works, this paper makes the case that even Turing machines are already a too powerful formalism to deal with the intuitive question of whether human thought is just like what a machine can do.

Turing already noted in (1950) that there is the somewhat paradoxical situation that for a given Turing machine T we can specify a problem instance for which T will be unable to provide the correct answer (e.g. an instance of the Halting problem). Yet, on the other hand, for every finite set of problem instances or questions that might be posed to a machine, one can construct a Turing machine which provides the correct answer to each and every of those listed questions. See also Abramson (2008) for an extended discussion of Turing's view.

Lucas (1961) published an argument based on a purely (meta-)mathematical theorem. He used Gödel's incompleteness results in order to argue that minds have more powerful computing capabilities than algorithms. Other researchers argued along with Lucas. Probably best known is Penrose with "The emperor's new mind" (Penrose 1989), or see in later follow-ups such as Penrose (1994, 1996).

Even without going into too much detail of the various arguments for and against AI, it appears intriguing that so far no ultimate agreement had been achieved. One

side, the proponents of the possibility of AI, argue that an AI is in principle possible. The other side, the opponents, claim basically that due to the fact that an AI system has to be algorithmic, it cannot behave intelligently in every possible situation. Since the system has to follow pre-determined rules which cannot be changed in order to produce behavior which may be called intuitive, creative, etc., i.e., behavior which is appropriate for situations which were not foreseen by the programmers. Abramson (2008) argues that Turing himself considered learning capabilities of machines as crucial to overcome the limits of a set of pre-determined rules. Somehow, both viewpoints appear plausible which is the reason for the still ongoing debate on the general possibility of an Artificial Intelligence.

This article argues that the AI debate so far failed to address a sensible question! It failed because it referred to the *too general* notion of algorithm. While the notion of algorithm is useful for mathematical purposes it is inappropriate for a discussion on the possibility of Artificial Intelligence.

The following Section will discuss this claim in more detail. A reformulation of the AI debate's question will be proposed as a result resolving the contradiction in the intuitions of both camps of the AI debate. The final Section contains concluding remarks.

Reformulating the Question on the Possibility of Artificial Intelligence

In 1937, Alan Turing introduced his Turing machine (1937) as explication of the intuitive notion of algorithm.[1] Moreover, a conjecture was stated by Church and Turing, known as the Church-Turing thesis, that the intuitive notion of an algorithm is actually explicated by the formal notion of the Turing machine. Probably most mathematicians and logicians would agree with the Church-Turing thesis. However, the thesis is not entirely uncontroversial. See e.g. Piccinini (2007) for a recent discussion.

The Confusing Notion of Algorithm in the AI Debate

Most arguments which claim limitations for an Artificial Intelligence refer to one of the following:

- An infinite class of problems for which it has been proven that there does not exist an algorithm, e.g. the decision problem of first order predicate logic.
  Actually, there is no empirical evidence available for the claims that humans do better than algorithms. Despite the fact, that this kind of objections to the enterprise of AI demand more from machines than from the human mind it appears also quite irrelevant for the practical aspect of AI systems whether a system is able to respond properly to an infinite number of conceivable requests with which the system will never be confronted. Since in practice always only a

[1] Around the same time other approaches like Church's λ-**calculus** (Church 1936) or Post's production systems (Post 1943) had been developed. Subsequently, however, it had been proven that all these different formalisms are equivalent with respect to the set of functions which they describe.

limited number of requests will be addressed to a system it is sufficient if the system can respond to those as desired.
On the other hand, if we restrict the considerations to a finite number of responses, it is obvious that there exists an algorithm which delivers all desired responses. Even a simple table look-up from a sufficiently large but finite table would do.

- The claim that at least occasionally human thought transcends any given rule scheme, e.g. that deviations from universal patterns of thought occur in creative thinking. See e.g., Abramson (2008) and Piccinini (2003), for a recent discussion of the issue.
  A significant problem with this objection is that there is actually no such rule scheme explicitly given. If a particular rule scheme is considered it might be possible to devise human thoughts which will not be properly described by the rule scheme. On the other hand, after showing where a given rule scheme fails, it is easy to fix the rule scheme in order to cover the respective gap.

The above mentioned problems with substantiating the claim of a non-algorithmic nature of the human mind are due to the fact that the notion of algorithm is *too general*. It just requires an algorithm to be of any but finite length. Though, intuitively, a huge number of rules probably 'feels' to be very close to an infinite number of rules.

Because the notion of algorithm allows any finite number of rules, for any finite number of observations there is a spectrum of algorithms possible which ranges from a plain and trivial listing of all desired responses for each of the possible requests to a set of rules which cover the required responses in some more general 'algorithm-like' style.

However, a plain listing of all desired responses contradicts our intuitive idea of an algorithm. An algorithm is supposed to generate a response (the result of the computation) by applying a number of 'general' rules. If we try to impose that intuition as a formal requirement, it would be easily satisfied by finding the mentioned table as a 'subtable' of an even larger collection of possible responses. That larger table would contain many entries which will never be recalled since they are beyond the initially assumed and in practice required responses. Thus, those table entries can be treated as 'don't cares' when describing the larger table by a possibly very large number of rules. Due to this, any formally imposed requirement for algorithms being rule-like would be undermined. As a consequence considerations on the limitations of algorithms per se have always to refer to infinite sets of problems. Referring to the infinite imposes a clear distinction between a mere lookup table and a 'real' algorithm - a rule scheme which subsumes a particular case under a more general class of cases.

While this reference to the infinite is daily practice and sensible in theoretical computer science, it is not sensible for the AI debate since the AI debate compares machines with humans. Humans have only a finite life span and, hence, cannot cope with an infinite set of expected actions or reactions anyway. Even more limiting, humans have only a rather limited capacity of absorbing information. For example, it appears unlikely that any human would in practice be able to determine whether a

set of, say, 100,000 propositional clauses contains a contradiction. And even 1,000 clauses will be a serious challenge.

By using the classical (Turing) notion of an algorithm, the AI debate simply shifts the debate away from the original intuition towards metaphysical speculations on what the human mind might in principle be able to do as opposed to actually doing it. While people will generally concede that no human will be able to deal with an infinity of questions or problems, they still tend to debate what a human being is in principle able to do or what their 'real' nature is. The latter being highly speculative as it implies an idealization of what humans actually do. For example it is being speculated whether a human could potentially give a correct answer to a problem instance from an infinite class of instances. Among those problem instances are also those which are described on so many pages that all actually living human beings would be making mistakes or would even pass away before they could arrive at an answer.

Restricting the Notion of Algorithm

In the following, the notion of algorithmic information is presented, which allows the consideration of restricted classes of algorithms.

Algorithmic Information Theory

In algorithmic information theory the amount of information necessary to construct a certain finite or infinite string s is considered. The amount of necessary information is measured as the minimal length of a program for a given universal Turing machine U that yields U to print exactly string s. Usually and without loss of generality, only *binary* strings of finite or infinite length consisting of '0's and '1's not containing any blanks or other symbols are considered.

Only programs are considered which do not receive any input.[2] The length of a binary encoded program p for some universal Turing machine U is denoted by $|p|$.

Definition 1 The length of the shortest program for constructing s is called its **Kolmogorov complexity** $K(s)$.

According to the Invariance Theorem (see e.g. Li and Vitányi 2008) the particular kind of considered universal Turing machine U makes only a difference of a certain constant c. Moreover, there are 2^n different binary strings of length n. Since each string requires a different program there are strings of length n whose Kolmogorov complexity $K(s)$ is at least n. Note: One program produces exactly one string—there is no input to the program. Moreover, *most* strings of length n have Kolmogorov complexity $K(s)$ of magnitude n.

Examples Strings as '11111111111111...1' or '00000000000...0' or '1010101010 101010...' etc. are presumably[3] simple strings, since their description by encoding a

[2] Input can be simulated by a subprogram which prints the required input onto the tape.

[3] Of course, the exact complexity depends on the considered universal Turing machine U and may be different for a 'nonstandard' universal Turing machine.

program which outputs the string under consideration is short. Mainly, only the length of the string is required for finite lengths, i.e., $\log_2(\textit{length of string})$ bits are sufficient. In addition, an indication whether '0's , '1's or alternating '1's and '0's should be printed is necessary. (This can be done with two further bits.)

In contrast to the strings above, strings like '10110010011101100101110101 0...' are more complex, i.e., require a longer program for being printed. The length of an appropriate program depends on the primitive instructions of the program-interpreting machine. By the Invariance Theorem mentioned above, it is shown that this dependence on the program-interpreting machine makes only a difference of a constant in the program size. This holds because for every universal Turing machine U' there is a finite program which can be run on a 'standard' universal Turing machine U and simulates U'.[4]

The Algorithmic Information of Intelligence

According to the Turing Test, we assume that for considering intelligent behavior of agents our considerations can be restricted to agents which communicate with their environment through finite strings of symbols from a finite alphabet. In other words, we can say that an agent behaves intelligently, if it shows a certain appropriate output to the input supplied to it. Furthermore, all observable behavior—intelligent or not—is a mapping from some finite input to some finite output. This may include control sequences to effectors of a robot as well as the input from visual, acoustic, or other sensors. Only a single finite input and a finite output stream is assumed for the complete lifetime of the intelligent agent. Learning behavior fits into that framework as follows:

To each part of the input stream there is a corresponding part in the output stream which can be viewed as the response to the input supplied. Thus, the response to the same pattern in the input stream may change or adapt over time.

It appears to be no restriction, if, in addition, it is assumed that the length of the input to an agent throughout its lifetime is finite. Thus, the required I/O-function that models the *intelligence* of any particular agent is simply some function from a finite number of input symbols to a finite number of output symbols. This implies that there definitely is a TM that models the intelligent behavior. Let us assume this I/O function is encoded as a binary string $s(f_{Int})$, e.g. the binary encoding of an appropriate TM table.

Then also f_{Int} has a certain Kolmogorov complexity $K(s(f_{Int}))$. In other words, the goal of AI is the development of a physical implementation of such a function f_{Int}.[5] It is clear that f_{Int} can be represented, e.g. as a binary string for feeding some universal Turing machine U. Say, the number n_i of possible binary input symbols a human agent will receive per second is not more than 1,000,000. Then the total

[4] To add another technicality to the considerations, we may assume that the considered universal Turing machine has less than say 1,000 lines in its Turing table. An example of such a universal Turing machine can be found, e.g., in Minsky (1962). This would result in the fact that the Kolmogorov complexity depends only to a very limited extent on the respectively considered universal Turing machine.

[5] This holds at least for the engineering approach of AI.

number of input symbols through a lifetime of, say 100 years, is upper bound by $n_i \leq 1,000,000 \times 8,640,000 \leq 10^{13}$. The number of output symbols through the agents lifetime n_o is also limited by the same amount, i.e., by $n_o \leq 10^{13}$. Then the length $|s(f_{Int})|$ of the binary encoded function f_{Int} is upper bound as follows:

$$|s(f_{Int})| \leq 2^{(10^{13})} \times 10^{13} \approx 10^{30,000,000,000,000}$$

Certainly $|s(f_{Int})|$ is astronomically large. However, the function f_{Int} does not need to be represented explicitly as a binary string. Instead it may be *compressed* as much as practical by describing the function by rules for entire classes of input strings or input substrings. The size of the most compressed form of *any* representation, whatsoever, is *lower bound* by the Kolmogorov complexity $K(s(f_{Int}))$.

It is important to note, that the Kolmogorov complexity of $s(f_{Int})$ provides a *strict* lower bound on the representation size independently of the considered interpreting machine. As the name 'algorithmic *information* theory' suggests, the particular kind of f_{Int} bears a certain amount of *information* in some absolute sense that is measured as the Kolmogorov complexity $K(s(f_{Int}))$.

In AI the notion of Kolmogorov complexity has mainly been applied in the context of inductive inference and generally in learning. See, e.g., Solomonoff (1964), who is considered to be one of the founders of the field of Kolmogorov complexity, or later work by Hutter (2005). See also Legg and Hutter (2007) for a presentation of those ideas in the context of the development of tests for artificial as well as natural intelligence.

The Revised Question on the Limitations of AI

The notion of Kolmogorov complexity allows to differentiate the discussion. The qualitative question of *whether intelligent behavior is algorithmic* should be replaced by a new, rather quantitative, question allowing a more fruitful discussion. The intuitive notion of algorithm in the context of the AI debate is then identified as being a matter of degree. I.e. is a relatively small set of rules capable of producing intelligent behavior that can pass the Turing Test? If we ask a question like the following, the debate becomes a much more productive one:

> *Can a given task*[6] *be accomplished by a given Universal Turing machine U running a program of length at most k*?[7]

or similarly, the question

> *What is the minimal Kolmogorov complexity of intelligent behavior capable of passing the Turing Test?*

While the exact number is of no importance, the rough order of magnitude is of importance and seems also to reflect the intuition behind those who argue that intelligence is non-algorithmic: i.e., is a rather small set of rules, e.g. some 10,000 or 100,000 rules sufficient in order to at least allow a system to bootstrap into an

[6] that can be accomplished by an intelligent human.

[7] For some particular k and some given universal Turing machine U.

intelligent system through learning from its environment, similarly to human babies? Or does even a learning system require a rather large algorithm say, of the order of 1,000,000,000 rules in order to allow a human-like development of an intelligent system that can pass the Turing Test? While 1,000,000,000 rules may seem excessive to some, a rough estimate of the possible Kolmogorov complexity of the human brain from what is known so far allows for much more. For example, it is believed that the human brain consists of some 10^{10} to 10^{11} neurons. Each of these neurons has up to some 10,000 to 100,000 connections to other neurons. Many of those connections appear rather irregular. I.e. this alone could potentially account for a descriptional complexity of the topology of the human brain of some

$$10^{10} \times 10^4 \times \log 10^{10} \approx 10^{15} = 1,000,000,000,000,000 \text{ bits.}$$

On top of that comes the functionality of each individual neuron and each individual connection between neurons—again there appear to be many different types and even two neurons of the same type would need to be described by many extra parameters to account for apparent differences. Just to mention a few quite well-known aspects: the length and thickness of the physical connections between neurons (the dendrites) are known to vary a lot and show quite complex shapes. The thickness, however, is known to strongly influence the speed of the signal propagation. Since neurons appear to operate asynchronously such that the exact timing of signal propagation is critically important, it is likely that the exact shape of those connections play a crucial role in the overall functioning of the brain. What is not well understood, though, is how much of that final complexity of an adult human brain is determined by learning experiences and how much is determined by genes and complex biological processes.

In any case, the above proposed reformulations should just indicate the direction in which the debate should move. Some more technical details are actually involved: e.g. the problem of practical feasibility of computing a certain behavior from a program of limited length. Some technical progress to attack these issues has been achieved in theoretical computer science. In particular, the notion of *resource-bound Kolmogorov complexity* has been developed which asks for the minimal program length for computing a given string efficiently. See Adleman (1979) for original work or Levin (1984) or Li and Vitányi (2008) for a more accessible publication on the use of the notion.

Whether the human mind is 'really' able to create infinite structures of infinite descriptional complexity seems to belong eternally to the domain of metaphysical speculations. Finite structures, e.g. the description of any finite number of symbolic results of intelligence and creativity, like poems, patents, scientific publications leading to the award of Nobel prizes etc. are obviously of finite Kolmogorov complexity and therewith algorithmically describable. And such finitely describable structures seem to be the only structures whose human origin can in principle be supported by empirical evidence. In addition, such finitely describable outputs of an intelligent being seem also to be the only one of practical importance.

Conclusions

This paper argued that the debate on the general possibility of Artificial Intelligence so far has failed to address a meaningful question. The reason is the usual reference to the notion of algorithm when the capabilities of machines are considered. The notion of algorithm has been explicated in the mathematical context in order to illuminate which mathematical objects can effectively be constructed. From a mathematical point of view, this is an important epistemological achievement. However, it has been taken for granted, that the same notion of algorithm is appropriate for the debate about the general possibility of an Artificial Intelligence. As discussed in this paper this is not the case. The notion of the Turing machine, i.e., of an algorithm, is inappropriate because it does not allow to draw the line between a rule-governed system and a mere look-up table for a finite set of expected behaviors. Hence, the fact that we can only ever experience a finite number of utterances etc. from humans renders the classical (Turing) notion of an algorithm quite useless in the context of the AI debate.

Instead this paper proposed to consider the quantitative aspect of the question. I.e. to consider how long an algorithm has to be in order to allow passing the Turing Test or to show any other desired intelligent behavior. With this reformulation the intuitions of those on opposing sides of the AI debate can be reconciled as follows: The opponents of the possibility of AI are in essence just saying that AI is not possible because human-level intelligence requires an impossibly long algorithm. This is no longer a direct contradiction with the proponents who are saying it that human-level intelligence is algorithmic, though there may still be disagreements over the actual order of magnitude of the length of an algorithm that can pass the Turing Test.

References

Abramson, D. (2008). Turing's responses to two objections. *Minds and Machines, 18*, 147–167.

Adleman, L. (1979). Time, space and randomness. Technical report, Massachusetts Institute of Technology. MIT/LCS/79/TM-131.

Church, A. (1936). An unsolvable problem of elementary number theory. *American Journal of Mathematics, 58*.

Copeland, B. J. (2002). Hypercomputation. *Minds and Machines, 12*, 461–502.

Floridi, L., Taddeo, M., & Turilli, M. (2009). Turing's imitation game: Still an impossible challenge for all machines and for some judges—an evaluation of the 2008 Loebner contest. *Minds and Machines, 19*, 145–150.

Hutter, M. (2005). *Universal artificial intelligence: Sequential decisions based on algorithmic probability*. Berlin: Springer.

Legg, S., & Hutter, M. (2007). Universal intelligence: A definition of machine intelligence. *Minds and Machines, 17*(4), 391–444.

Levin, L. A. (1984). Randomness conservation inequalities: information and independence in mathematical theories. *Information and Control, 61*.

Li, M., Vitányi, P. (2008). *An introduction to Kolmogorov complexity and its applications*. Text and Monographs in Computer Science, 3rd edn. Berlin: Springer.

Lucas, J. R. (1961). Minds, machines, and Gödel. *Philosophy, 36*, 112–117. (rpt. in: *Minds and Machines*, ed. Alan R. Anderson, Englewood Cliffs, NJ, Prentice-Hall, 1964).

Minsky, M. (1962). Size and structure of universal Turing machines using tag systems. In: *Recursive function theory, proceedings, symposium on pure mathematics*, Vol. 5, pp. 229–238. American Mathematical Society.
Penrose, R. (1989). *The Emperor's new Mind*. Oxford: Oxford University Press.
Penrose, R. (1994). *Shadows of the Mind*. Oxford: Oxford University Press.
Penrose, R. (1996). Beyond the doubting of a shadow. *Psyche, 2*, 89–129.
Piccinini, G. (2003). Alan Turing and the mathematical objection. *Minds and Machines, 13*, 23–48.
Piccinini, G. (2007). Computationalism, the Church-Turing thesis, and the Church-Turing fallacy. *Synthese, 154*, 97–120.
Post E. L. (1943). Formal reduction of the general combinatorial decision problem. *American Journal of Mathematics, 65*.
Sekanina, L. (2007). Evolved computing devices and the implementation problem. *Minds and Machines, 17*, 311–329.
Shagrir, O. (1997). Two dogmas of computationalism. *Minds and Machines, 7*, 321–344.
Shagrir, O. (2002). Effective computation by humans and machines. *Minds and Machines, 12*, 221–240.
Solomonoff, R. J. (1964). Complexity-based induction systems: Comparisons and convergence theorems. *Information and Control, 7*, 1–22 and 224–254.
Turing, A. M. (1937). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, 2*(42), 230–265 and (43) 544–546.
Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 59*, 433–460.