DE GRUYTER OPEN

STUDIES IN LOGIC, GRAMMAR AND RHETORIC 48 (61) 2016

DOI: 10.1515/slgr-2016-0058

Kazimierz Trzęsicki
University of Bialystok

CAN AI BE INTELLIGENT?

Abstract. The aim of this paper is an attempt to give an answer to the question what does it mean that a computational system is intelligent. We base on some theses that though debatable are commonly accepted. Intelligence is conceived as the ability of tractable solving of some problems that in general are not solvable by deterministic Turing Machine.

Keywords: artifical intelligence, intelligence, Turing machine, mind, brain, Church-Turing thesis, computability, Cobham-Edmonds thesis, the invariance thesis, NP=P?, NP-hardness assumption.

1. Introduction

The idea of artificial intelligence (AI) has a long history. Already in Greek myths we meet the golden robots of Hephaestus. Pygmalion fell in love with Galatea, a statue he had carved and which was animated by Aphrodite. Since the Middle Ages we have known Paracelsus' homonoculus and Golem, the creation of rabbin Jehudy Löw ben Bekalela. In the 19th century Mary Shelley created the fictional Frankenstein monster. Due to Karel Čapek, a Czech writer, the name "robot" is commonly used in everyday as well as scientific languages. Robots are equipped with intelligence, with AI.

The contemporary idea of AI came into being as digital Information and Communication Technology (ICT) developed. There were various afflatus, also with emotional background as was the case with Alan Turing. From letters to the mother of his dead friend Morcom, we know that at least for three years after his death Turing was overwhelmed by the problem of the interweaving of body and mind. He asked whether death releases the mind from the body. He looked for an answer in contemporaneous physics. He studied "The Nature of the Physical World" (1948) by Arthur Stanley Eddington, where on the base of quantum mechanics the traditional metaphysical problems of mind and matter were discussed. He speculated that

ISBN 978-83-7431-512-8 ISSN 0860-150X

quantum mechanics underpinned free will (Hodges, 1983, p. 63). The argument for the possibility of AI developed by Turing in *Computing Machinery and Intelligence* (1950) is one of the most cited in modern philosophical literature.

Turing imagined a machine that could mimic human reasoning. He stated the question "Can machines think?" (1950, p. 433). The question has obsessed computer and cognitive sciences. It grows more important every day as computers grow more powerful all the time. The possibility of a thinking computer, a computer apart from software, a machine with human-like (or super-human) intelligence is frightening, or at least thought-provoking. The risks are as enormous as the potential benefits.

In 1948 Norbert Winer published *Cybernetics, or Control and Communication in the Animal and the Machine*, a classic work that started contemporary cybernetics. Relations between information science and cybernetics, especially in Poland, were both disciplines tied as well theoretically as personally: Henryk Greniewski, the first leader of the Grupa Aparatów Matematycznych, the team that started Polish ICT, was one of the prime movers and co-founder of the Polish Association for Cybernetics (Polskie Towarzystwo Cybernetyczne).

The idea of creating an information system that could think inspires the study of brain and mind. One of the first and most known steps in this direction were the works of McCulloch and Pitts, in particular their *A logical calculus of the ideas immanent in nervous activity* (1943). For Turing, the human brain had ever been both inspiration and challenge to his work on computing machines. The association of the idea of universal computation and the idea of the brain as a computer resulted in the inception of AI as a distinct scientific discipline. AI was founded as an academic discipline in 1956 at a Dartmouth conference organized by Marvin Minsky, John McCarthy and two senior scientists: Claude Shannon and Nathan Rochester of IBM.

Sciences that investigate diverse aspects of acquisition and processing, and storage and transmission of data by the brain and mind, have been integrated under the common name "cognitive science". As birds inspired the idea of moving humans in the air, human cognitive acts as well give the base of the vision of artificial cognitive systems.

The question whether computers could be as intelligent as humans, or even more, is interesting for practical reasons, too. From day to day new achievements of ICT excite us and make our everyday life easier.

20 years ago Deep Blue won over Garry Kasparov, the Chess world master. In March 2016 AlphaGo won over Lee Sedol, the Go world vice-

championship.[1] There is a great difference between Chess and Go: at the opening move in Chess there are 20 possible moves; in Go the first player has 361 possible moves. This wide latitude of choice continues throughout the game (Bozulich, 2015). In the case of Go the number of possible steps is higher than the number of atoms in the universe. AlphaGo to win has been trained for two years.[2] Chess is considered to be more of a tactical game than a strategic one. Go is generally considered to be a game in which both strategy and tactics are equally represented. In June 2016 ALPHA, developed by Nick Ernst, in a simulated air fight beat Gene Lee, an experienced Air Force pilot.[3] In USA legislation is being prepared to allow autonomous cars to use public roads.[4]

IBM's Watson helps in health care. 90% of nurses accept its advice. Klaus-Peter Adlassnig, editor of *Artificial Intelligence in Medicine* suspects that Watson's medical knowledge is not deep and sweeping.[5] An information system trained in recognizing breast cancer by researchers from Beth Israel Medical Center and Harvard Medical School operated with 92% accuracy. In the case of diagnoses made by human pathologists it is 96%. In the case of diagnoses supported by the system it is 99.5%.[6]

Not only benefits but also threats posed by the development of AI are acknowledged. By 2040–50 computers as intelligent as man are forecast. A superintelligent computer should be constructed 30 years later. We have been warned by several high-profile voices that we should be more concerned about possible dangerous outcomes of supersmart AI. According to Hawking:[7]

> Success in creating AI would be the biggest event in human history. Unfortunately, it might also be the last, unless we learn how to avoid the risks. In the near term, world militaries are considering autonomous-weapon systems that can choose and eliminate targets.

In an interview for BBC he continues:[8]

> humans, limited by slow biological evolution, couldn't compete and would be superseded by A.I.

A similar opinion is shared by Steve Wozniak, one of the first creators of ICT.

Elon Musk, the founder of Tesla, Paypal, and SpaceX, decided to participate in the project DeepMind in order to:[9]

> just keep an eye on what's going on with artificial intelligence.

In January 2015 Bill Gates wrote:[10]

> I am in the camp that is concerned about super intelligence. First the machines will do a lot of jobs for us and not be super intelligent. That should be positive

> if we manage it well. A few decades after that though the intelligence is strong enough to be a concern. I agree with Elon Musk and some others on this and don't understand why some people are not concerned.

Nick Bostrom, Swedish philosopher and the director of the Future of Humanity Institute, Oxford University, sees an analogy between automobiles and computers (2014):

> Horses were initially complemented by carriages and ploughs, which greatly increased the horse's productivity. Later, horses were substituted for by automobiles and tractors. [...]
>
> When horses became obsolete as a source of labor, many were sold off to meatpackers to be processed into dog food, bone meal, leather, and glue. These animals had no alternative employment through which to earn their keep. In the United States, there were about 26 million horses in 1915. By the early 1950s, 2 million remained.

The fundamental question arises: is it possible to build an informational system that would be at least as intelligent as a human mind? We will analyse an argument for the thesis that the positive answer to the question can be seen as equivalent to the positive answer to the question, whether $NP = P$.

We believe that the intelligence of any physically realizable computational system cannot be higher than the intelligence inherent to the human brain. We argue that any physically accomplished computational system as well as the brain is able efficiently to solve merely problems that could be computed by deterministic Turing Machine (TM). Human minds are able efficiently to solve some problems that in general are not solvable by deterministic TM. Thus if $NP \neq P$, no physically realizable computational system as good as a brain is able to be as intelligent as a human mind. In polynomial time, deterministic TM computes any solution of NP problems, but deterministic TM is not able to solve some problems that in general are not solvable by deterministic TM, if $NP \neq P$. Hence deterministic TM only mimics intelligence, if it computes solutions of such problems.

2. Is the brain able efficiently to compute *NP* problems?

A comparison of computers and human brains shows the tremendous technological distance we have to overcome to build computers that would be almost as technologically advanced as human brains are. Many generations of engineers will have opportunities to invent, innovate, and improve ICT

to build computers that would be technologically more similar to human brains and their computational resources. We ask about the computational abilities of brains or in general of nature.

The question how nature computes was stated by Turing in the paper *The Chemical Basis of Morphogenesis* (1952). In many ways this is one of his most original and maybe visionary forays into the world of computation.

2.1. The Church-Turing Thesis

In 1936, Alan Turing (Turing, 1936–37) invented a theoretical computational model, the Turing Machine TM). It is a theoretical abstract object that provides a precise characterization of algorithmic solvability (Gandy, 1982):

> Both Church and Turing had in mind calculation by an abstract human being using some mechanical aids (such as paper and pencil). The word 'abstract' indicates that the argument makes no appeal to the existence of practical limits on time and space.

Computation, however complex, can be decomposed into simple atomic steps. It is proved that any other known models of computation, e.g. lambda calculus and partial-recursive functions, are equivalent to TM. The Church-Turing thesis (CT) claims that the class of well-defined computations is exhausted by those of TM. CT states that any other thinkable models of computational processes will be equivalent to TM. Until now there has not been found any such process that computes in an intuitive sense, and that is not equivalent to TM. The assertion of validity of CT is based on the lack of counterexamples. There are distinguished various TM: deterministic, probabilistic, nondeterministic, quantum etc. These "machines" are theoretical abstractions that do not, and can not, exist in the physical world. An all-purpose computer that could execute any algorithm is subject only to limitations of space and time. Nevertheless the fact remains that everyone who taps at a keyboard, opening a computer program, is working on an incarnation of a TM.

The concept of computability which CT seeks to analyze is an idealized one which is divorced in certain respects from our everyday computational practices: CT will classify a problem as effectively computable even if it is computable by a TM with resources that are astronomically large.

CT is also sometimes understood as making a prediction about which functions are physically (and biologically) computable — i.e. are such that their values can be determined by measuring the states of physical systems which we might hope to use as practical computing devices.

The physical CT, an analogical thesis to the abstract CT, states that any physical process whatsoever can be modelled by one of TMs (to an arbitrary degree of precision), provided only that the system to be modelled is governed by the laws of physics. The physical CT is not obviously evident but there are good reasons to accept it. To falsify the physical CT it is enough to point out a natural process that computes and cannot be modeled by TM. If the physical CT is false, it follows that there exists a physical process that effectively computes functions that aren't Turing computable.

Everything that humans are able to know is information. Any physical process can be represented as information processing. The idea that everything that is, is information, was conceived by Leibniz for whom the world is built by 1 (God) or 0 (nothing). He said:

> *Cum Deus calculat et cogitationem excercet, fit mundus* — when God thinks things through and calculates, the world is made.

Chaitin (2010, chapter 3, pp. 39–43) for both questions:
- Is the world built out of information?
- Is everything software?

answers "yes".

Notwithstanding any behaviour of a physical (and biological) process is a computational process, it can be questioned that the process can be modelled by TM. It is an empirical question whether the physical CT is true. Proponents of hypercomputation argue that there are physical processes — and so, potentially, machine-operations — whose behaviour conforms to functions not computable by Turing machine (Copeland, 2015). Others maintain that any physically computable function is computable by TM (Piccinini, 2007). According to Davis (M. Davis, 2006, p. 4)

> So, on what basis can someone claim that some device is indeed a "hypercomputer"? It can only be on the basis of physical theory. Such a theory would have to be certified as being absolutely correct, unlike any existing theory, which physicists recognize to be only an approximation to reality. Furthermore, the theory would have to predict the value of some dimensionless uncomputable real number to infinite precision. Finally, the specifications of the machine would have to guarantee that it exactly follows the demands of this supposed theory. Needless to say, nothing like this is even remotely on the horizon.

Davis rejects also the possibility of "computation beyond the Turing limit" by the nervous system and brain (M. Davis, 2004, pp. 6–10). The efforts of researchers (Siegelmann, 1995, 1999; Copeland, 1998, 1999) hoping to

construct a physical or biological "hypercomputational" device are by him (M. Davis, 2004, p. 1) compared to amateurs aiming to devise a construction for dividing a given angle into three equal parts using only straight-edge and compass or to the search for a perpetual motion machine (Park, 2000).

Davis (2004, p. 14) admits that quantum algorithms can provide exponential speed-up. However, they can only compute computable functions. Brains are biological objects, thus the physical CT is applicable to it. Hence, according to the physical CT, any brain computational process can be modeled as TM. It is the basic thesis of computational theory of the brain.

Theoretical CT supports physical CT and vice versa. If TM is a model of any computational process, then it is also a model of any physical computational process. If any physical computational process can be modelled by TM, then it is an argument for theoretical CT.

We are very far from understanding the workings of our mind, but there is every reason to believe that one of the things our mind does is to execute algorithms. Lucas (1961) on the basis of Gödel's Incompleteness Theorem and Penrose (1989, 1995, 1994, 1996, 2000) referring to quantum mechanics argues that the mind is not restricted by CT. He holds that no truly intelligent behaviour will ever be simulated by a computer since the function of brain cannot be simulated by a computer program because of its quantum mechanical physical basis.

Turing has formulated an argument from human 'mistakes' to explain why Gödel's theorem did not show the existence of an uncomputable human intuition. If it is accepted, as was already claimed by Turing (1950), that a discrete state machine is the appropriate level of description for mental states, i.e. that the mind is a TM, then CT is applicable to the human mind, and thus the psychological CT is acknowledged: any computable function by the human mind is computable by TM. The psychological CT is supported by the idea of reasoning as computation: *cogitatio est computatio* as the motto of Hobbes was cited by Leibniz in *Dissertatio de arte combinatoria*. Nevertheless there are good reasons for the conviction that it is certainly possible that psychology will find the need to employ models of human cognition that transcend TM.

Let us add that in his last published paper (Turing, 1992b, 1954) Turing, referring to the pure mathematics of computability maintained, unlike in (1950), that Gödel's theorem showed that 'common sense' was needed in interpreting axioms, and the intuitive 'common sense' was not asserted to be something a machine could show as well as a human being.

If we suppose that mind is merely a function of brain, the psychological CT is equivalent to the physical CT. If mind has at least one ability

that is not completely dependent on the brain, the psychological CT is not dependent on physical CT.

To summarize these considerations on TM let us cite Martin Davis (2004, p. 15):

> The great success of modern computers as all-purpose-algorithm-executing engines embodying Turing's universal computer in physical form, makes it extremely plausible that the abstract theory of computability gives the correct answer to the question "What is a computation?" and, by itself, makes the existence of any more general form of computation extremely doubtful.

2.2. Computational complexity

According to Chaitin (2010, p. 10) complexity theory is one of three, besides computability and information, theories, hot new topics in 20$^{\text{th}}$ century mathematics.

Computation by TM is a sequence of moves defined by a transition relation. In the case of deterministic TM the relation is a function, i.e. it is one-valued relation: for a given input there is one output. In the case of indeterministic TM there are possible more outputs; the transition relation is not a function. The deterministic TM computes sequentially. The indeterministic TM is a theoretical construction. It says little about the physical procedure of computation. Nevertheless the concept of indeterministic TM is an important idea in the theory of computational complexity. Indeterministic TM defines the computational tree.

The theory of computational complexity investigates the resources needed to accomplish computation (Dean, 2016). The origins of the theory lie in the work of Gödel, Church, Turing, Kleene, and Post undertaken in an attempt to answer Hilbert's *Entscheidungsproblem* (Trzęsicki, 2006). Central to the theory is the notion of a decision problem. Its primary goals are to classify and compare the practical difficulty of solving problems about finite combinatorial objects. This theory provides tools of classification of problems and methods of measurement of computational resources. It explains why some problems are intractable and provides measures of anticipation of difficulties of their solution. Complexity theory attempts to provide a formal criterion for what it means for a problem to be feasibly computed. The classification is quantitative and aimed at investigation of required, i.e. the lower limit, and sufficient, i.e. the upper limit, resources to accomplish computation. A problem is considered to be complex in proportion to the difficulty of carrying out the most efficient algorithm by which it may be decided. Decision problems are commonly categorized into complexity

classes based on the fastest known machine algorithms. Decision problems may change class if a faster algorithm is discovered.

The most important and usually applied measures of computational complexity are time (temporal-complexity), which is needed to accomplish computation, i.e. the number of basic steps required by a machine to halt and return an output; and space, the amount of memory used in computation (memory-complexity). The basic definitions of time (one time unit is assigned to every transition) and space (the number of cells in the memory which have been affected by the computation) complexity were formulated by Hartmanis and Stearns (1965), (Hartmanis, 1981). One of the aims of computational complexity is to distinguish problems that are feasibly computable.

The efficiency of a machine is measured in terms of its time-complexity. Other measures of computational complexity are also applied, e.g. the number of processors. Time-complexity refers to the increasing number of machine basic steps needed by an algorithm relative to the size of the problem, where a basic step takes a fixed amount of time to perform. The focus on time of computation is natural and justified: any other parameters influence the time of computation; e.g. the larger the space, the amount of memory, the more time is needed (Garey & Johnson, 1979). The amount of time taken and the number of elementary operations performed by an algorithm differ by at most a constant factor.

P (deterministic Polynomial time) is the class of decision problems that are algorithmically solved in polynomial time. An algorithm is polynomial time iff its running time is upper bounded by the value of the polynomial expression in the size of the size of the input: $\mathcal{O}(n^k)$, where n is the length of input data and k is a constant that depends on the problem, but not the particular instance of the problem. Big-$\mathcal{O}$ is an expression of how the execution time of a program scales with the input data. The concept of big-$\mathcal{O}$ can be used for more than runtime, e.g. it is used to describe how much memory an algorithm uses.

Any problem of P and only a problem of this class is solvable by a conventional MT, i.e. a deterministic MT, in a number of steps which is proportional to a polynomial function of the size of its input.

NP (Non-deterministic Polynomial time) is the class of decision problems solvable in polynomial time by a theoretical non-deterministic Turing machine. It consists of those problems which can be correctly decided by some computation of a non-deterministic TM in a number of steps which is a polynomial function of the size of its input. Equivalently it means that the instances of the decision problems where the answer is "yes" are veri-

fiable by deterministic TM in polynomial time. The algorithm consists of two phases which consist of:

1. a guess about the solution, which is generated in a non-deterministic way,
2. a deterministic algorithm that verifies or falsifies the guess as a valid solution to the problem.

P can be characterized as a class of problems, membership in which can be decided efficiently, whereas NP can be characterized as the class of problems for which membership can be verified efficiently once an appropriate certificate is provided. P describes the class of feasibly decidable problems. NP are easy to check but impossibly hard to computationally solve.

Any solution to a problem of P also in polynomial time verifies the correctness of the solution, thus any problem of P is also a problem of NP. In other words, any problem solvable in polynomial time by deterministic TM is also solvable by non-deterministic TM in polynomial time. The complexity class P is contained in NP. The most important open question in complexity theory, the P versus NP ($P = NP$), asks whether NP is contained in P. It is widely believed that this is not the case (Mole, 2016, p. 20), (Feinstein, 2003). In a 2002 poll of 100 researchers, 61 believed the answer is "no", 9 believed the answer is "yes", 22 were unsure, and 8 believed the question may be independent of the currently accepted axioms, and so impossible to prove or disprove.[11] It is one of the seven Millenium Problems.[12] The Clay Mathematics Institute declares a prize of million dollars for solution to it.

For the first time the problem today known as $NP =?P$ was remarked by Kurt Gödel in a letter to John von Neumann (Hartmanis, 1989). The letter is translated and published by Sipser (1992). The rigorous formulation of NP versus P was done by Stephen Cook in the paper *The complexity of theorem proving procedures* (1971). Cook proved that the SAT problem is NP-hard. The SAT problem eventually assumed the role of paradigmatic "hard" problem.

Let us make a distinction between an instance of NP problem and NP problem or a problem that in general is NP (Thagard, 2000). To explain the difference let us take into account SAT. SAT is NP-complete. The problem that a given formula, an instance of SAT, is satisfiable, is a problem which in general is NP-complete.

NP-complete problems are the hardest NP problems. The solution to at least one NP-complete problem in polynomial time is sufficient to deal with any other NP problem in polynomial time. To find at least one instance of an NP-complete problem is enough to prove that $NP \neq P$.

Since the P versus NP problem is unresolved, no algorithm for an NP-complete problem is currently known to run in polynomial time and no instance of a NP-complete problem is currently known that could not be solved in polynomial time. NP-complete problem is such that there is no polynomial time algorithm for its solving for the worst case.

NP-intermediate problems are those that are between P problems and NP-complete problems, i.e. they are neither in the class P nor NP-complete. NP-hard problems are those at least as hard as NP problems, i.e., all NP problems can be reduced (in polynomial time) to them. NP-hard problems need not be in NP, i.e., they need not have solutions verifiable in polynomial time.

Humans are able efficiently to solve some problems that in general belong to $NP \setminus P$, i.e. they are NP-intermediate or NP-complete. Humans are able efficiently to perform computations that are not modelled by deterministic TM, if $NP \neq P$. For example, mathematicians are able efficiently to formulate and prove theorems, though it is a task that in general belongs to $NP \setminus P$, or — in other words — is modelled by indeterministic TM but not by deterministic TM, if we suppose that $NP \neq P$. Hackers efficiently crack passwords though these are problems that in general are not solvable by deterministic TM, though they need superpolynomial time. Generally, in the case of NP-complete problems, humans are able to solve:

1. special cases
2. small problem sizes,

or they can give:

3. approximate solutions
4. probabilistic solutions.

2.3. The Cobham-Edmonds Thesis

The attempt to develop a general theory of feasible computability was accompanied by a systematic exploration of the relationships between different models of computation. These investigations resulted in distinguishing between the class of feasibly solvable problems, i.e. those which are solvable in practice by an efficient algorithm, and the class of intractable problems, i.e. those which are regarded as intrinsically difficult to solve.

The Cobham-Edmonds thesis (Cobham, 1965; Edmonds, 1965b, 1965a) (CET) states — let us omit the technical formulation and details — that a problem can be efficiently computed iff it can be computed in polynomial time (Mole, 2016, p. 23). According to the thesis only algorithms that can be performed in polynomial time — those which lie in the complexity class P — are efficient (fast, tractable, feasible, easy, practical), while algorithms that

are not in P — that require superpolynomial time — are inefficient (slow, intractable, unfeasible, hard, impractical). Exponential time-complexity has been taken as the touchstone of intractability. CET relies on the informal notion of a reasonable model of computation.

CET has some limitations in application to determination of efficiency of computation, because the thesis abstracts away some important variables that influence the time of computation. It ignores:

- constant factors and lower-order terms,
- the size of the exponent,
- the typical size of the input.

CET is more debatable than CT. The incomputability of a problem can be proved purely theoretically as is the case with the Halting Problem (Turing, 1937). The grid colouring-task is an example of problems that could not be solved, since the universe does not provide sufficient time or space in which to get the thing done. In a twenty by twenty grid there are four hundred squares. There are 2^{400} possibility of colourings of theses squares either white or black. Even if colouring of one square will last the shortest possible time, i.e. Planck's time, t_P,[13] the number of these units from the beginning of the universe (about $4,34 \times 10^{26}$) is a tiny fraction of number of t_P needed to reach the output (Mole, 2016, p. 15). The completion of this task in general requires an exponentially large number of operations which goes beyond the bounds of the resources of the universe.

On the one hand, according to CET, for example, a computer program requiring n^{100} steps would be tractable, though even for $n = 10$ the time needed to execute this algorithm is greater than the age of the universe (Cook, 2006). On the other hand, an algorithm requiring $2^{0.00001n}$ steps could be executed for e.g. $n = 10^6$, though as belonging to exponential time problems, is intractable (Rotman, 2003). From a mathematical angle it is clear that for enough big input, any polynomial time algorithm will beat any exponential time algorithm. CET is considered to be a good rule of thumb for real-life problems. A proof that $P \neq NP$ would provide additional evidence that CET gives a correct analysis of the pre-theoretical notion of feasibility.

As a consequence of acknowledging the physical CT, we have to accept that laws of complexity theory, as pure theoretical laws, are applicable to information processing in nature, in particular in the brain. Physical computational processes require resources that are required by TM. In other words, the principles of the assessment resources needed by TM are applicable to assess resources needed by a natural process. The phys-

ical CT says nothing about tractability. The physical CET, as a consequence of physical CT and (theoretical) CET, maintains: a natural computational process is tractable only if it can be accomplished in polynomial time. In particular, brain computational processes are tractable only if they could be accomplished in polynomial time. According to the physical CET any tractable cognition problem is solvable in polynomial time. Any cognitive problem that requires superpolynomial time is not efficiently solvable by brain; it is practically unsolvable, intractable for the brain (Mole, 2016, p. 25).

Are there any physical computational processes at all that need superpolynomial time? If the universe is never-beginning, all the natural processes have had enough time to be accomplished. Any problem, regardless of time-complexity, would be already solved by natural processes. The idea of a never-beginning universe is not confirmed by contemporary cosmology. The beginning of the universe is estimated at approximately 13.8 billion years ago. Thus at least processes that require more than 13.8 billion years are not yet accomplished.

Cognitive ability can be described as the skills of information processing in order to reason, decide, intend, react etc. According to psychological CT any mind process could be modeled by TM. The human mind is a finite system with limited computational resources. The cognitive abilities of the human mind are limited to processes that can be practically executed. Some cognitivists maintain that the only tractable processes are these that could be accomplished in polynomial time. The thesis — i.e. psychological CET — is subject of many interesting discussions (van Rooij, 2008). Such a thesis is a consequence of psychological CT and (theoretical) CET.

If we suppose that mind is merely a function of brain, the psychological CET is equivalent to the physical CET. If mind has at least one computational ability that is not completely dependent on the brain, the psychological CET is not dependent on physical CET.

According to psychological CET, for a cognitive agent, tractable processes are solely those solvable in polynomial time. The class P approximates the class of feasible mind problems.

2.4. The Invariance Thesis

We suppose that any computational process such as theoretical (Church-Turing thesis) or physical (physical Church-Turing Thesis) and psychological (psychological Church-Turing Thesis) is modelled by TM. From CET (Cobham-Edmonds Thesis) as applied to theoretical devices, and the physical CT (Church-Turing thesis) it follows that any physical computational

process is tractable only if it can be executed in polynomial time (physical Cobham-Edmonds Thesis). The same is true in the case of the psychological computation process (psychological Cobham-Edmonds Thesis).

We are interested in comparing the complexity of algorithms implemented in different effective models of computation. We ask if the computational complexity of a problem can change if the computing device is changed. The time of execution of an algorithm depends on the calculating device. Some devices are able to solve a problem in shorter time than other devices. Can devices differ in complexity of the same problem; in particular are there any devices that in polynomial time solve NP problems that are not P? The thesis of invariance (TI) maintains that (van Emde, 1988, p. 5) (van Emde, 1990, p. 2):

> There exists a standard class of machine models, which includes among others all variants of Turing Machines [and] all variants of RAM's. ... Machine models in this class simulate each other with Polynomially bounded overhead in time, and constant factor overhead in space.

'Reasonable' models of computation can simulate each other within a polynomially bounded overhead in time and a constant-factor overhead in space. A 'Reasonable' machine is a machine that could be constructed (van Emde, 1990, p. 2).

IT states that all standard models of computing devices are equivalent in the sense that the fundamental complexity classes do not depend on the precise model chosen for their definition. Changing of a (theoretical) computational device can, beside the constant space coefficient, polynomially change the time of execution but the possibility of changing the time complexity of a problem is excluded (Slot & van Emde, 1984). In particular, if a problem is executed in superpolynomial time on one device, it is also, maybe in a shorter or longer time, executed in superpolynomial time on any other computational device. For example, some computation by indeterministic TM can be modelled as probabilistic computation, but it does not mean that probabilistic TM has the same ability as indeterministic TM. It is proved that NP-complete problems are not computable in polynomial time, if $NP \neq P$ (Implagliazzo & Wigderson, 1997).

P is the smallest time-complexity class on a deterministic TM which is robust in terms of machine model changes. Any deterministic TM will have a complexity class corresponding to the problems which can be solved in polynomial time on that machine.

The TI, as pertaining to physical computational devices, is a consequence of physical CT and (theoretical) TI. Thus we claim that the time

of execution of an algorithm on one device, regardless of whether a theoretical or physical one, can, besides the constant space coefficient, differ merely polynomially from the time of execution of the algorithm on another device, regardless of whether theoretical or physical. In particular it means that any NP problem that is not P is NP that is not P independently of the used TM or physical computational device.

From TI it follows that any physical process that is modelled by deterministic TM is not able to solve any NP-intermediate or NP-complete problem in polynomial time, if $NP \neq P$. From this it does not follow that some natural processes do not exist that are modelled by indeterministic TM.

Of course, the psychological IT follows from the psychological CT and (theoretical) TI: the time of execution of an algorithm by any psychological computational process can, besides the constant space coefficient, differ merely polynomially in time of execution of this algorithm on any other computational device. From TI it follows that any psychological process that is modelled by deterministic TM is not able to solve any NP-intermediate or NP-complete problem in polynomial time, if $NP \neq P$. A computation by mind as modelled by deterministic TM can only be executed polynomially faster or slower than is the case of another TM.

TI and CET are not dependent on one another. The CET does not exclude that there are machines which in polynomial time solve problems that for other machines are superpolynomial time problems. TI says nothing about the tractability of computation. TI, similarly as CT and CET, is only a practically confirmed thesis, for which no counter-example has been found.

2.5. *NP* Hardness Assumption

Let us ask if there are any tractable natural processes that could not be modelled by deterministic TM. Any natural computational process is tractable, as stated by CET, only if it is accomplished in polynomial time. According to IT this is only possible for problems of the class P. Let us remember that all the theses taken into consideration are not proven but they are merely practically confirmed, and such that for which no counter-example is found. Thus the possibility that some natural process computes in polynomial time some problems that are NP-complete for (theoretical) TM should not be excluded. We ask if some NP-hard problems can be solved in polynomial time using the resources of the physical universe, i.e. we ask (Aaronson, 2005, p. 1):

> can NP-complete problems be solved in polynomial time using the resources of the physical universe?

The most known proposals of natural processes that supposedly execute in polynomial time exponentially many steps are soap bubbles, protein folding, quantum computing, quantum advice, quantum adiabatic algorithms, quantum-mechanical nonlinearities, hidden variables, relativistic time dilation, analog computing, Malament-Hogarth spacetimes, quantum gravity, closed timelike curves, and "anthropic computing" (Aaronson, 2005). To better understand what we speak about let us mention some of these processes.

The time of execution of a calculation depends on the interval of time that is needed for one step. If any step could be executed in an arbitrarily small interval, any process despite time-complexity could be tractable, i.e. it could be accomplished in polynomial time. This idea has to be rejected because in quantum mechanics the smallest interval of time is determined by Planck's constant. Thus for any sufficiently big number any exponential algorithm could not be accomplished in polynomial time since it could not be possible to shorten the interval of execution beyond 5.39×10^{-43} seconds.

In relativity theory the measure of time is dependent on the speed of the agent, thus the agent could expect a result of computation in reasonable time: it is enough to move him with a velocity near to the speed of light. But the acceleration of the agent is possible only if there is enough energy. Since for some calculation it could be greater than the amount of energy in the universe, the idea has to be rejected, too.

If the universe would be infinite, the number of parallel computation processes could not be limited, and thus any computation regardless of time-complexity could be executed in polynomial time. But contemporary physics claims that the universe is finite in size (though limitless). Some maintain that arbitrarily many parallel processes are allowed in quantum mechanics. But at least till now it is rather a phantasy. The number of parallel computations in the human brain is tremendous but limited (Litt, Eliasmith, Kroon, Wienstein, & Thagard, 2006). Abilities of our brains are not decided by their sizes. It is known that:[14]

> the distance from "village idiot" to "Einstein" is tiny, in the space of *brain designs*.

An international research group at Peking University is "working together to understand the unified theory of cognition and to provide a coming generation of a reasoning machine, beyond the current model of the von Neumann machine."[15]. The Klar (Knowledge Learning And Reasoning), an algorithm elaborated by the group, in its opinion, is able to perform

the NP task efficiently.[16] In this case it is enough to repeat Davis's remark on hypercomputation.

In conclusion we may accept that the hardness of NP-complete problems is a basic fact about the physical world. A physical process in either an atomic universe or a quantum mechanical universe that could cross the border of IT is questionable in contemporary science (M. Davis, 2001, pp. 677–679). There is no natural computational process that can accomplish exponentially many operations in polynomial time. Alternative models of computation are dreamt about in Science Fiction (van Emde, 1990, p. 1). The answer to the question if NP-complete problems can be solved efficiently in the physical universe, is "no". The NP Hardness Assumption says — loosely speaking — that NP-hard problems are intractable in the physical world (Aaronson, 2005, p. 17)

TI could cover psychological computational processes. It would do so if any mind computational process was a brain computational process. If there are some mind computational processes that are not completely exhausted by brain processes, it would not be the case. There are some arguments in favour of the thesis that mind is able efficiently to solve some problems that are not computed by deterministic TM, e.g. the Lucas argument based on Gödel's theorem and many others. Of course, the arguments are not indisputable (Krajewski 2003).

3. Computability and intelligence

Although $P \neq NP$ is widely believed to be true, let us ask about the consequences of $NP = P$.

Humans can — as finite beings — know only that which is computable in polynomial time. Thus humans can know what is solvable by non-deterministic TM only if it is computable by mind in polynomial time. Physical computers — this is a consequence of the NP-hardness Assumption — can efficiently solve only P problems. Hence, if $NP = P$, all knowledge that could be achieved by human beings would be achievable by the human brain, and by computers.

The great consequence of $NP = P$ was already suggested by Gödel who in a letter to von Neuman[17] wrote:

> If there actually were a machine with [running time] $\sim Kn$ (or even only with $\sim Kn^2$) [for some constant K independent of n], this would have consequences of the greatest magnitude. That is to say, it would clearly indicate that, despite the unsolvability of the *Entscheidungsproblem*, the mental effort of the mathematician in the case of yes-or-no question could be completely [added in

> a footnote: apart from the postulation of axioms] replaced by machines. One would indeed have to simply select an n so large that, if the machine yield no result, there would than also be no reason to think further about the problem. (Sipser, 1992, p. 612) (Aaronson, 2006, p. 3)

A world in which there would be no fundamental gap between solving a problem and recognizing the solution once it is found, would be a profoundly different place than we usually assume it to be.

According to Cook (2006, p. 94):

> Although a practical algorithm for solving an NP-complete problem (showing $P = NP$) would have devastating consequences for cryptography, it would also have stunning practical consequences of a more positive nature, and not just because of the efficient solutions to the many NP-hard problems important to industry. For example, it would transform mathematics by allowing a computer to find a formal proof of any theorem that has a proof of reasonable length, since formal proofs can easily be recognized in polynomial time. Such theorems may well include all of the CMI [Clay Mathematics Institute] prize problems. Although the formal proofs may not be initially intelligible to humans, the problem of finding intelligible proofs would be reduced to that of finding a recognition algorithm for intelligible proofs. Similar remarks apply to diverse creative human endeavors, such as designing airplane wings, creating physical theories, or even composing music. The question in each case is to what extent an efficient algorithm for recognizing a good result can be found. This is a fundamental problem in artificial intelligence, and one whose solution itself would be aided by the NP-solver by allowing easy testing of recognition theories.

We have argued that physical (and biological) systems are able efficiently to solve only P problems or, in other words, only physical systems that can be modeled as deterministic TM compute efficiently. We do not exclude that there are NP-intermediate, NP-complete, or NP-hard natural processes. We have excluded only that these processes are tractable, i.e. according to CET, that they are computational processes that can be accomplished in polynomial time.

Are there any reasons to maintain that a physical device is not able to solve efficiently a problem that in general is not P, provided that $NP \neq P$? If the answer is "yes", there is an important difference between the computational abilities of physical processes and mind processes. The difference can be taken as a characteristic of intelligence. To the question "what does it mean to be intelligent?" we can answer: a system is intelligent if the system is able efficiently to solve problems that in general are not P.

Intelligence conceived in such a way could be measured by the amount of solved problems that in general are not P.

No physical (or biological) device is able in polynomial time to solve (general) problems that are not P. Does it follow that there are no instances of such problems that are efficiently solvable by such a device? Let us try to answer the question.

Any solution of an NP problem is verifiable in polynomial time. Hence, any solution of such a problem can be verified by deterministic TM. Correctness of a mathematical proof can be verified. For example, such is the aim of MIZAR, http://mizar.org/, a project conceived by Andrzej Trybulec, University of Białystok. Any problem that is NP in general, if solved by human, can also be solved by a computer with the appropriate software. It means that the intelligence of computers can grow as the intelligence of the human grows (McCarthy & Hayes, 1969, p. 4):

> A machine is intelligent if it solves certain classes of problems requiring intelligence in humans, or survives in an intellectually demanding environment.

There is a whole range of intractable foundational questions across a wide range of research areas in science and the humanities. The limits of human knowledge could be reached if all the NP problems would be solved. If $NP \neq P$, the limits of knowledge can never be reached. If so, our intelligence will grow and we will be able to build ever more intelligent computers but never will computers be more intelligent than humans. The improvement of AI is the direction in the development of computing devices. Thus AI, the part of computer science concerned with designing computer systems that exhibit the characteristics we associate with intelligence in human behavior (Barr & Feigenbaum, 1981–84), will remain a subject of deeper and ever more advanced technological research.

Turing's very first written mention of 'intelligent' machinery is in *Proposal for Development in the Mathematics Division of an Automatic Computing Engine* (?). In his famous *Computing Machinery and Intelligence* (1950), Turing astutely narrowed down what one can sensibly say about human intelligence, and discussed in some detail his observer-based test for a thinking machine. About the prospect of a thinking machine he wrote (1950, p. 442):

> I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted. I believe further that no useful purpose is served by concealing these beliefs. The popular view that scientists proceed inexorably from established fact to established fact, never being influenced by any unproved conjecture, is quite mistaken. Provided it is made clear which are proved facts and which are conjecture, no harm can result. Conjectures are of great importance since they suggest useful lines of research.

Now we do not ask if machine can think, but we ask about what the machine can think and if there are some limits of thinking by the machine.

The result of our considerations can be put simply: human intelligence will be always a step ahead of machine intelligence. Machine intelligence depends on human decisions and the abilities to construct machines and write software, even if such machines are able to imitate such human abilities as: learning, teaching, training, searching (Turing, 1950). Machine intelligence is a result of endowment of machines with intelligent software by a human. Though, for example, in the case of genetic algorithms, the final designer is no longer a human but a computer, it is the human designer who has to design the fitness function. We are the only educator of computers. Such an opinion agrees with Post's conviction expressed in a discussion of the thesis that man is not a machine as a consequence of the incompleteness properties of formal systems (Post, 1965, p. 423):

> All we can say is that man cannot construct a machine which can do all the thinking he can.

Turing says the same; for him computers (1950, p. 438):

> can in fact mimic the actions of a human computer very closely.

Turing was well aware of the paradox of expecting intelligence from a machine capable only of obeying orders. But he believed that, with sufficient complexity, machines need not appear 'mechanical' as in common parlance (Turing, 1948, 1969, p. 10):

> if we are trying to produce an intelligent machine, and are following the human model as closely as we can

a good approach would be to allow the machine to learn just like humans. Today the leading approach to AI is machine learning. Deep learning is an extremely effective technique for training computers to recognize patterns in images or audio, enabling machines to perform with human-like competence useful tasks such as recognizing faces or objects in images.

Can computers thus built following the human model as closely as we can, be more intelligent than humans? Are we racing towards the Singularity — a point at which AI outstrips our own and machines go on to improve themselves? If intelligence is conceived as the ability to solve in polynomial time instances of problems that in general are not P, the ultimate answer to these questions depends on solving the problem of whether $NP = P$.

4. Conclusion

In the summary of *Intelligent Machinery* (1948, 1969, p. 20) we read:

> The analogy with the human brain is used as a guiding principle. It is pointed out that the potentialities of the human intelligence can only be realised if suitable education is provided.

In the conclusion of our consideration let us ask, what/who has educated humans to be intelligent?

From our point of view, if intelligence is conceived as the ability to solve efficiently problems that in general are not P, the source of intelligence could be in participation with an intelligent being in processes which can not be modelled by deterministic TM. All living beings are subject to evolution. Evolution seems to be a process that is not in P. Evolution as a physical computational process is not tractable. Thus evolution will never achieve its ultimate goal; it will never end. It will last as long as life will. Key to Chaitin's notion of evolution is something he calls creativity. One theorem of his (2010, p. 11) metabiology is that evolution will continue indefinitely, that biological creativity is endless, unceasing. Due to biological creativity, evolution causes "Darwinian" fitness to increase faster than any computable function.

Calculations done by the process of evolution are more or less advanced. It depends on time. According to Moravec's paradox (1988, pp. 15–16):

> Encoded in the large, highly evolved sensory and motor portions of the human brain is a billion years of experience about the nature of the world and how to survive in it. The deliberate process we call reasoning is, I believe, the thinnest veneer of human thought, effective only because it is supported by this much older and much more powerful, though usually unconscious, sensorimotor knowledge. We are all prodigious olympians in perceptual and motor areas, so good that we make the difficult look easy. Abstract thought, though, is a new trick, perhaps less than 100 thousand years old. We have not yet mastered it. It is not all that intrinsically difficult; it just seems so when we do it.

It can be said that humans to adapt to survive do not wait for genetic evolution but instead replace it by intelligence.

Evolution begets intelligence. Any living being is thus intelligent in a degree proper to its own specie. (We assume that animal intelligence is human-like to the extent that the animal itself seems human-like.) Intelligence allows organisms to adapt to their environment continually during life and allows organisms to survive. It allows humans to defy generational

selection and develop intelligences external to our own, making use of computational techniques. The development of AI is similar to biological evolution. Computers can be trained. Turing even remarked on an analogy of training with evolution (Turing, 1969). The human mind is largely a bundle of hacks and heuristics, cobbled together over a billion years of evolution (Baum, 2006) and living in intelligent society. Currently popular approaches to teach computers include biologically inspired algorithms such as swarm intelligence and artificial immune systems, which can be seen as a part of evolutionary computation, image processing, data mining, and natural language processing.

Humans beings are the most intelligent due to the creation of and participation in another process that is not modelled by deterministic TM, namely society (Marciszewski, 2003, 2004). Social life, relations, and the cooperation of people and communities are not tractable processes. Communication and cooperation stimulate the speed of processes of ever growing intelligence at an enormous rate. In contrast to the slow rate at which advantageous evolutionary adaptations spread, ever-increasing advances in information technologies like the invention of language, Gutenberg's printing press, and contemporary ICT, have resulted in an ever-increasing speed of successful adaptations to reality and harnessed the environment to our advantage. It is the principal reason why information sharing is beneficial for humans. Some people believe that the internet will develop an entirely new form of intelligence (Gelertner, 2016).

The development of ICT seems the most important for the growth of the intelligence of the human race. The growth of the volume of the human brain is limited by biology. Life, in general, evolves to become more complex. Human brains are too small, as computational devices, to compute the process at the human stage of evolution. The development of ICT can be seen as something enhancing the volume of the brain. Human intelligence begets artificial intelligence. Due to cooperation and support by 'thinking' machines we are able to solve some problems with the ever-increasing computational complexity that the future brings.

The difference between human intelligence and machine intelligence is seen as the difference between reasoning and thinking. From the point of view of computational complexity there is a difference between solving P problems and solving problems that in general are not P. Humans are able to think. Machines are only able to reason. Reasoning is modelled by deterministic TM. Thinking comprises reasoning, as all P problems are NP problems, but is creative and imaginative. Thinking is the youngest human ability from an evolutionary point of view (Moravec's paradox). It can

not be modelled by deterministic TM. AI need not replicate human cognition directly, but a better understanding of human commonsense might be a good place to start (E. Davis & Marcus, 2015). For McCarthy (1980) not only:

> humans use ... 'non-monotonic' reasoning, [but also] it is required for intelligent behavior.

The same is true about using non-exact and non-complete knowledge. Much of what we call "insight" or "intelligence" simply means finding succinct representations for our sense data. Free-associations are an important part of human thought. No computer will be able to think like a man unless it can free-associate. No computer will be creative unless it can simulate all the nuances of human emotion. The question of aesthetics, of intuition, of instinct, of judgement is highly subjective. But — as Chaitin maintains (Chaitin, 2010, p. 15):

> There is nothing more important than experiencing beauty; it's a glimpse of transcendence, a glimpse of the divine, something that fewer and fewer people believe in nowadays. But without that we are mere machines.

Uncovering the relationship between thinking and learning seems central to understanding the nature of intelligence. To make computers more intelligent they must use:

- fuzzy logic — to enable understanding of natural language
- artificial neural networks — to permit the system to learn experiential data like the biological one
- evolutionary computing — which is based on the process of natural selection
- learning theory
- probabilistic methods — which help dealing with uncertainty imprecision.

Even if one day all these problems will be solved and artificial thought will be achieved, an artificially intelligent computer will experience nothing and be aware of nothing. It will say "that makes me happy," but it won't feel happy. Still: it will act as if it did. It will act like an intelligent human being. And then what?

Would it ever be possible for computers to do the same as humans do? Yes, if they achieve a life instinct and evolve to achieve the ability to create a society of cooperating individuals. In Chaitin's (Chaitin, 2010, p. 9) opinion DNA is a universal programming language which is rich enough

to express any algorithm. The fact that DNA is such a powerful programming language is a more fundamental characteristic of life than mere self-reproduction, which anyway is never exact for if it were, there would be no evolution. DNA is our software; it's the programming language for life.

NOTES

[1] http://www.wired.com/2016/03/googles-ai-wins-first-game-historic-match-go-champion/?mbid=nl_3916

[2] http://techcrunch.com/2014/01/26/google-deepmind/

[3] http://arstechnica.com/information-technology/2016/06/ai-bests-air-force-combat-tactics-experts-in-simulated-dogfights/ [29.06.2016]

[4] http://cyberlaw.stanford.edu/wiki/index.php/Automated_Driving:_Legislative_and_Regulatory_Action, http://spectrum.ieee.org/transportation/advanced-cars/how-googles-autonomous-car-passed-the-first-us-state-selfdriving-test [11.09.2016]

[5] http://www.bloomberg.com/news/articles/2014-01-10/ibms-artificial-intelligence-problem-or-why-watson-cant-get-a-job

[6] https://www.engadget.com/2016/06/19/ai-breast-cancer-diagnosis/ [21.06.2016]

[7] http://www.independent.co.uk/news/science/stephen-hawking-transcendence-looks-at-the-implications-of-artificial-intelligence-but-are-we-taking-9313474.html

[8] http://www.bbc.com/news/technology-30290540

[9] https://www.washingtonpost.com/news/morning-mix/wp/2014/11/18/why-elon-musk-is-scared-of-killer-robots/

[10] https://www.reddit.com/r/IAmA/comments/2tzjp7/hi_reddit_im_bill_gates_and_im_back_for_my_third//:

[11] https://www.artofproblemsolving.com/wiki/index.php?title=P_versus_NP (viewed October 31, 2016)

[12] http://www.claymath.org.millenium

[13] In the Planck scale system of physical measurements it is approximately 5.39×10^{-43} seconds. Contemporary physics suggests that this is an in principle lower bound on the duration of a primitive step in a concretely embodied computing process.

[14] http://lesswrong.com/lw/ql/my_childhood_role_model//

[15] http://cognitivelogic.org/id1.html

[16] http://cognitivelogic.org/index.html

[17] https://rjlipton.wordpress.com/the-gdel-letter/, http://cognitivelogic.org/id8.html (viewed 03.11.2016)

REFERENCES

Aaronson, S. (2005, March). NP-complete problems and physical reality. *ACM SIGACT News 36*(1), 30–52. Retrieved from www.scottaaronson.com/papers/npcomplete.pdf (viewed October 31, 2016).

Aaronson, S.(2006). *Quantum computing since Democritus.* Retrieved from http://www.scottaaronson.com/democritus/ (viewed October 31, 2016) (Lecture notes, Fall 2006).

Barr, C. P., A. & Feigenbaum, E. (Eds.). (1981–84). *Handbook of artificial intelligence* (Vols. 1–4). Los Altos, CA: William Kaufmann.

Baum, E. B. (2006). *What is thought?* (Revised edition ed.). Cambridge MA: MIT Press.

Bostrom, N. (2014). *Superintelligence: Paths, dangers, strategies.* Oxford: Oxford University Press.

Bozulich, R. (2015). *Chess and go: A comparison.* Chigasaki, Japan: Kiseido Publishing Company.

Chaitin, G. J.(2010). *Metabiology: Life as evolving software.* Retrieved from https://www.cs.auckland.ac.nz/~chaitin/metabiology.pdf

Cobham, A. (1965). The intrinsic computational difficulty of functions In *Proceedings of the Third International Congress for Logic, Methodology, and the Philosophy of Science* (pp. 24–30). Amsterdam: North Holland.

Cook, S.(1971). The complexity of theorem proving procedures. *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, 151–158.

Cook, S. (2006). The P versus NP problem. In J. A. James A. Carlson, A. Jaffe, & A. Wiles (Eds.), *The millennium prize problems* (pp. 87–106). Cambridge Mass.; Providence Rhode Island: Clay Mathematics Institute, American Mathematical Society. Retrieved from www.claymath.org/sites/default/files/pvsnp.pdf (viewed October 31, 2016)

Copeland, J. B. (1998). Turing's O-machines, Penrose, Searle, and the brain. *Analysis*, *58*, 128–138.

Copeland, J. B. (1999). Alan Turing's forgotten ideas in computer science. *Scientific American*, *253*(4).

Copeland, J. B.(2015). The Church-Turing thesis. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* Summer 2015 (ed.). http://plato.stanford.edu/archives/sum2015/entries/church-turing/ (viewed October 31, 2016).

Davis, E. & Marcus, G. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, *58*(9), 92–103. doi: 10.1145/2701413

Davis, M. (Ed). (1965). *The undecidable: Basic papers on undecidable propositions, unsolvable problems, and computable functions* (1965, 2004 ed.). Hewlett, N.Y.: Raven Press. (An anthology of fundamental papers on undecidability and unsolvability, this classic reference opens with Gödel's landmark 1931 paper demonstrating that systems of logic cannot admit proofs of all true assertions of arithmetic. Subsequent papers by Gödel, Church, Turing, and Post single out the class of recursive functions as computable by finite algorithms. 1965 edition.)

Davis, M. (1982). Why Gödel didn't have Church' Thesis. *Information and Control*, *54*, 3–24.

Davis, M. (2001). *Engines of logic. Mathematicians and the origin of the computer.* New York: W. W. Norton & Company.

Davis, M. (2004). The myth of hypercomputation. In C. Teuscher (Ed.), *Alan Turing: Life and legacy of a great thinker. Turing Festschrift.* Berlin, Heidelberg: Springer-Verlag. Retrieved from https://www.researchgate.net/publication/243784599 doi: 10.1007/978-3-662-05642-4_8

Davis, M. (2006). Why there is no such discipline as hypercomputation *Applied Mathematics and Computation*, *178*, 4–7.

Dean, W. (2016). Computational complexity theory. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Winter 2016 ed.). http://plato.stanford.edu/archives/win2016/entries/computational-complexity/ (viewed October 31, 2016).

Eddington, A. S. (1948). *The nature of the physical world* (Electronic edition 2007 ed.; R. C. Henry, Ed.). Cambridge: Cambridge at the university press. Retrieved from henry.pha.jhu.edu/Eddington.2008.pdf (viewed October 31, 2016).

Edmonds, J. (1965a). Minimum partition of a matroid into independent subsets. *Journal of Reaserch of the National Bureau of Standards-B. Mathematics and Mathematical Physics*, *69*, 67–72.

Edmonds, J. (1965b). Paths, trees, and flowers. *Canadian Journal of Mathematics*, *17*(3), 449–467.

Feinstein, C. (2003). *Evidence that P is not equal to NP* (Tech. Rep. No. cs.CC/03 10060).

Gandy, R. (1982). Church's thesis and principles for mechanisms. In J. Barwise (Ed.), *The Klenee symposium.* Amsterdam: North-Holland.

Garey, M. R. & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness.* New York: Freeman.

Gelertner, D. (2016, 24. November). Ein Geist aus Software. *Frankfurter Allgemeine Zeitung.* Retrieved from http://www.faz.net/aktuell/feuilleton/debatten/digitales-denken/kuenstliche-intelligenz-ein-geist-aus-software-1607431.html

Hartmanis, J. (1981). Observations about the development of theoretical computer science. *Ann. Hist. Comp.*, *3*, 42–51.

Hartmanis, J. (1989). Gödel, von Neumann, and the $P =?NP$ problem. *Bulletin of the European Association for Theoretical Computer Science*, *38*, 101–107. Retrieved from https://ecommons.cornell.edu/handle/1813/6910 (viewed October 31, 2016).

Hartmanis, J., & Stearns, R. (1965). On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, *117*(5), 285–306.

Hodges, A. (1983). *Alan Turing: the Enigma.* London: Burnett (Polish translation (Hodges, 2002)).

Hodges, A. (2002). *Enigma. Życie i śmierć Alana Turinga*. Warszawa: Wydawnictwo Prószyński i S-ka (translation of (Hodges, 1983) by W. Bartol).

Impagliazzo, R. & Wigderson, A. (1997). $P = BPP$ if E requires exponential circuits: derandomizing the XOR lemma. In *Stoc '97 proceedings of the twenty-ninth annual acm symposium on theory of computing* (pp. 220–229). New York: ACM. doi: 10.1145/258533.258590

Krajewski, S. (2003). *Twierdzenie Gödla i jego interpretacje filozoficzne. Od mechanicyzmu do postmodernizmu*. Warszawa: Instytut Filozofii i Socjologii PAN.

Litt, A., Eliasmith, C., Kroon, F. W., Weinstein, S. & Thagard, P. (2006). Is the brain a quantum computer? *Cognitive Science*, *30*, 593–603.

Lucas, J. R. (1961). Minds, machines and Gödel. *Philosophy*, *36*, 112–127. http://users.ox.ac.uk/~jrlucas/Godel/mmg.html (viewed October 31, 2016).

Marciszewski, W. (2003). A comment on Hayek's ideas of free market and civilization. *Studies in Logic, Grammar and Rhetoric*, *19*(6), 107.

Marciszewski, W. (2004). Challenges for the logic of social research: To grasp rationality, to deal with complexity. *Studies in Logic, Grammar and Rhetoric*, *20*(7), 17.

McCarthy, J. (1980). Circumscription-a form of non-monotonic reasoning. *Artificial Intelligence*, *1–2*(13), 27–39.

McCarthy, J., & Hayes, P. J. (1969). *Some philosophical problems from the standpoint of artificial intelligence* (Tech. Rep.). Stanford, CA 94305: Computer Science Department, Stanford University.

McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* (5), 115–133.

Mole, C. (2016). *The unexplained intellect: Complexity, time, and the metaphysics of embodied thought*. New York: Routledge.

Moravec, H. (1988). *Mind children: The future of robot and human intelligence*. Harvard University Press.

Park, R. (2000). *Voodoo science. The road from foolishness to fraud*. Oxford, U.K. & New York: Oxford University Press.

Penrose, R. (1989). *The emperor's new mind*. Oxford: Oxford University Press. (Polish translatation (Penrose, 1995)).

Penrose, R. (1994). *Shadows of the mind*. Oxford: Oxford University Press. (Polish translatation (Penrose, 2000)).

Penrose, R. (1995). *Nowy umysł cesarza: o komputerach, umyśle i prawach fizyki*. Warszawa: PWN. (translated by Stefan Amsterdamski).

Penrose, R.(1996). Beyond the doubting of a shadow. *Psyche*. Retrieved from http://psyche.csse.monash.edu.au/v2/psyche-2-23-penrose.html (electronic journal).

Penrose, R. (2000). *Cienie umysłu. poszukiwanie naukowej teorii świadomości*. Poznań: Zysk i S-ka. (translated by Stefan Amsterdamski).

Piccinini, G. (2007). Computationalism, the Church–Turing thesis, and the Church–Turing fallacy. *Synthese* (154), 97–120.

Post, E. (1965). Absolute unsolvable problems and relatively undecidable propositions — Account of an anticipation. In M. Davis (Ed.), *The undecidable: Basic papers on undecidable propositions, unsolvable problems, and computable functions* (pp. 338–433). Hewlett, N.Y.: Raven Press. (submitted to the American Journal of Mathematics in 1941, but published only in (M. Davis, 1965, pp. 338–433)).

Rotman, B. (2003, August). Will the digital computer transform classical mathematics? *Philosophical Transactions of the Royal Society. Matehmatical, Physical and Engineering Sciences*, *361*(1809), 1675–1690. doi: 10.1098/rsta.2003.1230

Siegelmann, H. T. (1995). Computation beyond the Turing limit. *Science*, *268*, 545–5498.

Siegelmann, H. T. (1999). *Neural networks and analog computation: Beyond the Turing limit*. Boston: Birkhäuser.

Sipser, M. (1992). The history and status of the P versus NP question. In *STOC '92: Proceedings of the twenty-fourth annual ACM symposium on theory of computing* (pp. 603–618). New York: ACM.

Slot, C. F., & van Emde, P. (1984). On tape versus core: an application of space efficient perfect hash functions to the invariance of space. In ACM (Ed.), *STOC '84 Proceedings of the sixteenth annual ACM symposium on theory of computing* (pp. 391–400). New York.

Thagard, P. (2000). *Coherence in thought and action*. Bradford Book.

Trzęsicki, K. (2006). From the idea of decidability to the number Ω. *Studies in Grammar, Logic and Rethoric*, *9*(22), 73–142.

Turing, A. M. (1936–37). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, *42*(Series 2), 230–265. Retrieved from http://www.abelard.org/turpap2/tp2-ie.asp

Turing, A. M. (1937). On computable numbers, with an application to the *Entscheidungsproblem. Proceedings of the London Mathemactical Society*, *2*(1), 230.

Turing, A. M. (1948, July). *Intelligent machinery* (Tech. Rep.) National Physical Laboratory. Retrieved from http://www.AlanTuring.net/intelligent_machinery

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *49*, 433–460. (Available on-line: http://cogprints.org/499/00/turing.html. Reprinted in (Turing, 1992a)).

Turing, A. M. (1952). *Can automatic calculating machines be said to think?* (Typescript of broadcast discussion in BBC Third Programme, 14 and 23 January 1952, between M.H.A. Newman, A.M. Turing, Sir Geoffrey Jefferson, R.B. Braithwaite).

Turing, A. M. (1954). Solvable and unsolvable problems. *Science News* (31), 7–23.

Turing, A. M. (1969). Intelligent machinery. National Physical Laboratory Report, 1948. In B. Meltzer & D. Michie (Eds.), Machine intelligence 5 (pp. 3–23). Edinburgh: Edinburgh University Press. (Digital facsimile viewable at http://www.AlanTuring.net/intelligent_machinery)

Turing, A. M. (1992a). *Collected works of A. M. Turing: Mechanical intelligence* (D. C. Ince, Ed.). Amsterdam: North Holland.

Turing, A. M. (1992b). Solvable and unsolvable problems. In D. Ince (Ed.), Collected works of alan mathison turing (Vol. III, pp. 187–203).

van Emde, B. P. (1988, Aug.). Machine models and simulations (Revised version) (Tech. Rep. No. CT-88-05). University of Amsterdam: Institute for Language, Logic and Information. Retrieved from http://www.illc.uva.nl/Research/Reports/CT-1988-05.text.pdf (viewed October 31, 2016).

van Emde, B. P. (1990). Machine models and simulations. In J. van Leeuwen (Ed.), *Handbook of theoretical computer science* (Vol. A: Algorithms and Complexity, pp. 1–66). Cambridge: MIT Press/Elsevier.

van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science. A multidyscyplinary journal 32*(6), 939–984. Retrieved from http://onlinelibrary.wiley.com/doi/10.1080/03640210801897856/full (viewed October 31, 2016).

Wiener, N. (1948). *Cybernetics: Control and communication in the animal and the machine* (rev. ed. 1961 ed.). Cambridge Mass.: MIT Press.