Can Cognitive and Intelligent Systems Outperform Turing Machines?*

Jiří Wiedermann

Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
jiri.wiedermann@cs.cas.cz

"Everything that's worth understanding about a complex system can be understood in terms of how it processes information."
Lloyd's Hypothesis [10]

Abstract. We look for computational limits of artificial, natural and hybrid cognitive and intelligent systems. The common basis for such studies is offered by computationalism, i.e., the belief that cognitive or intelligent processes, respectively, are in essence computational processes. We show that in principle cognitive systems might exist whose computational power outperforms that of Turing machines and that even in practice we observe the rudiments of such systems. These results point to the fact that the so–called Church–Turing Thesis, dealing with the central position of Turing machines in the world of computations and algorithms, must be seen in the context of physical principles exploited by the cognitive systems, and in that of the communication scenario between the system and its environment.

1 Introduction

The term "cognition" usually denotes the activities by which the living organisms collect, process, store and utilize information. These activities especially include perception, learning, memorization, and decision making [11]. W.r.t. this definition intelligence can be seen as a part of cognition which is less interested in perception and focuses mainly to the quality of cognitive processes. Both cognition and intelligence are related to information processing. The so–called *computationalism* heralds the belief that human, or biological cognition and intelligence present a specific kind of computations (cf. [3]). The proponents of this school of thoughts claim that the computational modelling of cognitive abilities of living organisms is at least in principle possible and that in this way one can achieve if not a genuine than at least an approximative capturing of all mental faculties (inclusively of thinking and consciousness) and the explanation of the underlying algorithmic principles. In the sequel we will be interested in the efficiency of abstract computational devices used in computational cognition

* This research was carried out within the institutional research plan AV0Z10300504 and partially supported by grant No. 1ET100300419.

modelling of living organisms. We will measure this efficiency by the standard methods used in the computational complexity theory, i.e., we will compare it to the efficiency of standard basic models known within this theory. We will look for the computational limits of the cognitive models. We will be especially interested in their efficiently in processing the data in order to solve cognitive problems and, last but not least, whether there are cognitive problems which, in principle, cannot be solved by these models. In the rest of this abstract we will be simply speaking only about cognition which in the framework of its previous informal definition also seems to be a key notion for the definition of intelligence.

By the end of the past century computationalism has obtained an unexpected support both from the theoretical physics and computer science. In 1985 a paper by the theoretical physicist D. Deutsch appeared [5] showing that any real (dissipative) finite physical system can be efficiently simulated by a quantum computer. Since a quantum computer can be simulated by a Turing machine (albeit, as it seems, quite inefficiently) we have a proof of the computationalistic claim that, e.g., man can be genuinely simulated, at least in principle, by a quantum computer. Another result from the computational complexity theory asserts that this simulation will be efficient, indeed (it will be of polynomial time complexity w.r.t. the size of the simulated physical system [2]).

In our approach we will further generalize the scope of computationalism by proceeding beyond the cognitive abilities of living organisms *per se*: our considerations will include any organisms (such as humans) equipped by whatever device which will "strengthen" their cognitive capabilities, or allow their new quality. The Hubble telescope mounted on a satellite encircling the Earth may serve as an example of such a device of which the control and computing center on the Earth is also a part. No doubts that such a machinery will strengthen the cognitive abilities of an observer using this device. Clearly, using this device an observer gets an access to data inaccessible to him by his own senses. Moreover, these data are processed in a way which, without computers, is also beyond men's abilities. We will call the resulting system, i.e., an observer as well as his or her apparatus, the *cognitive system*. The resulting "hybrid" cognitive system is clearly endowed by a new quality of cognition which for a man without the respective devices is unattainable. Let us be broad–minded by not insisting on the cognitive device being really constructed and at one observer's disposal in his or hers experiments. We will be happy just with the gedankenexperiments, i.e., with the situation when the assumed existence of a "cognitive amplifier" does not violate any natural law. That is, we can be prevented in building such a device by its price, its size, its large energy consumption, etc., but not by the physical laws. We will also admit models comprising an unbounded, albeit always finite, number of cognitive subjects. It means that in our thought experiments we accept exploitation of all known properties of space–time, of quantum and relativistic effects, the emergence of new universes, cultivation of the evolving colonies of living organisms, "growing-up" a human society communicating over the Internet, etc. Standing firmly to the ground of computationalism, any kind of devices just mentioned can eventually be thought of as a data processing system. In the end, using this rather general approach everything reduces to the question about the limits of all "thinkable" cognitive (read: computational) systems based on whatever principles obeying natural laws.

The investigation of the computational limits of various computational machineries belongs among the fundamental issues in the computability theory and in the computational complexity theory. The *computational power* of a class of computing devices relates to the set of computational problems which the device at hand can in principle solve. The larger this set, the larger is the computational power of devices in the given class. Along these lines, the standard upper bound on the computational power of computational systems is offered by the so–called *Church–Turing Thesis* (cf. [1], [7]) which, for our purposes, can be formulated as follows[1]:

Church–Turing Thesis: The computational power of whatever class of computational devices controlled by finite algorithms does not extend beyond the Turing machines.

Of course, for the first time the Turing machine was defined by its inventor, A.M. Turing in 1936 [12]. The Thesis itself was formulated later after the first computers appeared. Note that what the Thesis claims is, in other words, that there are no computational devices which could outperform Turing machines as far as their computational power is concerned. The Thesis cannot be proven, since it does not specify exactly what is meant by *"whatever class of computational devices controlled by finite algorithms'* and henceforth one cannot prove how a Turing machine could simulate such a class of devices. However, provided we construct (or at least, we show how to construct) a device solving the problems insolvable by any Turing machine, the Thesis will be refuted. In the computability theory the problems which are in principle insolvable by any Turing machine are called *undecidable problems.* The past has witnessed many unsuccessful attempts to refute the Church–Turing Thesis. It is their failure which eventually has lead to the thesis formulation and to the belief of its validity. Only quite recently has it appeared that the Thesis is to a certain extent incompletely stated: the explicit assumptions under which the Thesis holds are missing. These assumptions must concern the physical theory in the framework of which the computational devices mentioned in the Thesis are realized (cf. [6]) and the way in which the system interacts with its environment [17]). As far as the first condition is concerned, nowadays it seems that, roughly speaking, as long as we stay within the world whose laws are well approximated by the laws of the classical, newtonian physics and we only consider finite computations, the Thesis holds. However, should the "modern" physics be considered, inclusively the quantum and relativistic physics, it appears that the Thesis need not hold true [6], [8]. An alternate way for trespassing the Thesis is to consider a more general scenario of computations than that assumed by Turing in its design of the Turing machine. This idea leads to the so–called *interactive evolutionary computing systems* characterized by potentially infinite computations interacting

[1] The Thesis is known in many forms, we have chosen the one which stresses the notion of the computational power.

with their environment and by a continuous hardware modification during the computations [16], [17]. Using these ideas one can capture the computational behavior not only of living organisms over many generations, but also of the Internet or similar evolving computing networks [17].

In the paper we concentrate on the cognitive systems of a certain type which possess the potential for going beyond the Church–Turing Thesis. In order to be able to speak about their deviations from the classical computations as realized by Turing machines referred to by the Thesis, we first mention briefly the computational systems whose computing mechanisms are akin to those of the Turing machine and which do not refute the Thesis. Nevertheless, these systems, modeling the cognitive subjects, will present a basis for construction of more complex systems opening the door for attacking the Thesis (cf. [18]). Based on works [6], [19] we then describe a relativistic cognitive system which outperforms the Turing machines by realizing infinitely many computational steps in a finite time when seen from a suitable observer's point of view. We will then proceed to the so–called *evolutionary interactive systems* which can be realized either by the artificial systems resembling the Internet or by evolving communities of living organisms [18]. All machines refuting the thesis will make heavy use of the external non–computational elements entering into the design of the underlying machines (cf. [4]). In the theory, the non–computational elements are modelled by so–called non-uniform algorithms. In fact, these algorithms are algorithms of infinite length which exist in two forms. First, they can be supported by infinite sequences of simple finite–size devices each of them being capable to process inputs of greater size than its predecessors (cf. [1], [14], [17]). Second, non–uniform computational device (or algorithms) can take form of universal (i.e., programmable) machines making use of oracles [13] or of their weaker relatives called *advices* [9]. The purpose of oracles or advices is to provide non–computable information on demand. We show that all devices mentioned above trespass the computational barrier imposed by the Turing machines and all are equivalent to the so–called *interactive Turing machine with advice* [16]. We conclude our overview by stating the extended Church–Turing Thesis capturing the computational potential of the contemporary computing technologies as well as that of information processing in biological systems.

Extended Church–Turing Thesis: The computational power of whatever class of interactive computational devices controlled by non–uniform algorithms does not extend beyond the interactive Turing machines with advice.

The purpose of this survey has been to bring the recent developments related to the Church–Turing Thesis to the attention of researchers working within the field of artificial intelligence and artificial life. Irrespectively, whether the computationalism is right in its approach to cognition and intelligence, the Thesis and its modern form is crucial for understanding the limits of the computational potential of any natural or artificial system whose behavior is based on data processing.

2 Should we rather model organisms by finite automata or by Turing machines?

The basic notion we shall be using for a while is the notion of *configuration* of some finite artefact, organism, or of a matter in a fixed space volume. All these categories will be termed as *devices*. A configuration of a device is a term relative w.r.t an observer of this device. Namely, an observer will say that in two successive times a device is in the same configuration if for this observer the device appears to be the same in these times. Of course, in order to be able to tell apart the two configurations the observer can be equipped by a special device. This means that for an observer the number of distinguishable configurations is determined by the resolution ability of that observer. The contemporary quantum physics sets a theoretical upper bound on the number of (quantum) configurations a device of a given mass and volume can enter. S. Lloyd has shown [10] that the number of quantum configurations which can be entered by 1 kg of a matter in a volume of 1 liter can be of order at most 10^{31}. This is a huge, but finite number which can bee seen as an upper limit on the memory capacity of thinkably the most efficient memory of a given size. Matter achieves such a capacity under extreme conditions existing perhaps during the big–bang. Nonetheless, within the framework of the previous consideration any finite device can serve as a memory of a capacity given by observable, or measurable physical parameters of this device.

Any device can be seen as a device computing in accordance with the given sequence of transition rules (i.e., with a program) if and only if it fulfills the following rather general conditions:

1. there must be a possibility to change the configuration of the device by the input data interacting in a predetermined manner with the device;
2. it must be possible to set the device into a distinguished initial configuration;
3. the device in a given configuration interacting with the given input must enter the next configuration; the dynamics of such a transition must correspond to the transition rules, i.e., the device must "all by itself" cause the transition from one configuration into the other in accordance with its program;
4. the computational dynamics depends on the input data.

The property ensuring that the device causes something "all by itself" means nothing else that there is a mechanism in the device working in the desired way: the device is "made" in this way. The transition rule need not be known — it is enough id it exists and it it finitely describable. The classical real computer can serve as the prime example of such a device; here the transition rules are known, similarly as in the case of of automatic teller machines, mobile phones, etc. The brain presents another example of a computing device with the unknown set of transition, but the computationalists believe that it does exist). A rock, a picture, a memory card, a mathematical model of a Turing machine are examples of devices which do not compute in the sense defined above. Note that we did not define neither the result of the computation, nor its termination. The has been done intentionally — our computing device should realize potentially *never ending computations.* Stated differently, the device transforms a potentially infinite stream of input data (which are called stimuli in the case of cognitive systems)into a potentially infinite stream of output data called action in the case of living organisms; the sequence of actions correspond to the behavior. In this case it is possible and the definition admits that some input data can represent reaction of the environment to some actions. Hence we can speak about interactive computations. Obviously, with the the device just described we can also realize finite computations — simply by artificially restricting the input stream. E.g., from a certain position the input stream will consist but of empty symbols and we will be interested only in terminating computations.

In the sequel we will only deal with classical (i.e., with the discrete, non–quantum) computational devices of a finite size. Formally, such a device is equivalent to so–called interactive finite–state automaton with output, which are also called interactive transducers [17].

Definition 1. *An* interactive finite–state transducer *(IFT)* $\mathcal{T}$ *is a finite–state automaton consisting of the finite–state control and the input and the output port.*

- *The* input port *serves as the entry point through which* $\mathcal{T}$ *reads the input data. These data are symbols from a finite* input alphabet Σ*. At each step there is a symbol from* Σ *on the input port (it can also be the* empty input symbol $\varepsilon \in \Sigma$ *denoting "no input"); once a symbol is read, in the next step a next input symbol will appear on the input port.*
- *The* output port *is the place where* $\mathcal{T}$ *sends in each move so–called output symbols — the elements from the finite* output alphabet Γ*.*
- *The finite–state control is defined formally with the help of a finite* set of states Q *and of a transition relation* Δ*. At each step the control is in one state of* Q*. Transition relation is of form* $\Delta \subseteq \Sigma \times Q \times \Gamma \times Q$*. An element* $\delta = (\sigma, q_1, \gamma, q_2) \in \Delta$ *has the following meaning:*"reading σ at the input port, $\mathcal{T}$ in state q_1 sends γ to its output port and enters state q_2".

Each IFT defines a relation between the input and output streams. In the deterministic case to each input stream there is at most one output stream. This is achieved by considering the transition function $\Delta : \Sigma \times \rightarrow \Gamma \times Q$ instead of a relation of the form as described in Def. 1. In this case we speak about the translation of the input stream to an output stream.

Thus, comparing an IFT with the commonly known finite–state automaton with output (so–called Mealy automaton) we see that the input data for an IFT are not given on an input tape before the start of a computation and there need not be a finite number of them. That is why the output of an IFT can be an infinite stream.

Some computing devices can have a specific ability to increase their memory capacity. This can be achieved either by an additional mechanism or by connecting several computing mechanisms together. This additional memory capacity enables these devices to create and explite a potentially unbounded set of configurations. A so–called *interactive Turing machine* (ITM) (cf. [17]) can serve as an example of such a device. An ITM is basically a standard Turing machine which has no input tape. Instead, it reads the input symbols via the input port and sends the output symbols to its output port. An ITM can be seen as an IFT which in order to increase its memory capacity (depending on the cardinality of Q) makes use of a potentially infinite tape. This tape alone cannot compute, but in a symbiosis with an IFT which is endowed by the ability to move along the tape while reading and rewriting the symbols on the tape, leads to a more powerful computational device than was the IFT alone. An ITM computes all what was computed by an IFT, but also more than that. This is because it can enter more than a finite number of configurations.

In fact, an ITM computes more that the classical Turing machine. From the viewpoint if its construction an ITM is the extension of a classical Turing machine for the case of infinite input streams; this is what enables to an ITM to compute "more" than the classical Turing machine. For instance, an ITM can process an infinite sequence of finite data segments. Of course, each such segment can also be processed by a standard Turing machine. However, the latter machine has no means for "transferring" information obtained from processing a finite segment in one run into the next run. This is simply because the standard Turing machine, after terminating its computation, cannot be restarted from the state in which it terminated its previous computation: according to its definition, the standard Turing machine must start a new computation from its initial state, with all its tapes empty. For instance, the standard Turing machine cannot realize the following translation: if a segment of a stream gets accepted (the machine produces 1), then the following segment will always be rejected (the machine produces 0.) The computational abilities of ITMs are studied in [15].

References

1. Balcázar, J. L., Diáz, J., Gábarró, J.: Structural complexity, Vol. I, 2nd Edition, Springer-Verlag, Berlin, 1995.
2. Bernstain, E., Vazirani, U.: Quantum Complexity Theory. Proc. of the 25-th Annual Symposium on the Theory of Computing, ACM, New York, 1993, pp. 11–20
3. Chalmers, D. J.: A Computational Foundations for the Study of Cognition. Minds and machines, Vol. 4, No. 4, 1994
4. Davis, M.: The myth of hypercomputation. In Teuscher C. (ed.) Alan Turing: Life and legacy of a great thinker. Springer-Verlag, Berlin, Heidelberg, 2004
5. Deutsch, D.: Quantum Theory, the Church-Turing principle and the universal quantum computer. Proc. of the Royal Society of London A 400, pp. 97-117, 1985
6. Étesi, G., Németi, I.: Non-Turing computations via Malament–Hogarth space-times, Int. J. Theor. Phys. 41, 2002, pp. 341-370, or http://xxx.lanl.gov/ps/gr-qc/0104023
7. Hopcroft, J., Motwani, R., Ullman, J. D.: Introduction to automata theory, languages and computation, 2nd Edition, Addison-Wesley, Reading, MA, 2000.

8. Hogarth, J.: Non–Turing computers and non-Turing computa bility, Proceedings of the 1994 Bienal Meeting of the Philosophy of Science Assocoation, PSA 1994, Vol. One, pp. 125–138, 1994
9. Karp, R. M., Lipton, R. J.: Some connections between non-uniform and uniform complexity classes, in Proc. 12th Annual ACM Symposium on the Theory of Computing (STOC'80), 1980, pp. 302-309.
10. Lloyd, S.: How Fast, How Small, and How Powerful: Moor's Law and the Ultimate Laptop. In: Rebooting Civilization, http://www.edge.org/3rd_culture/ rebooting/rebooting.html, 2001
11. Shettleworth, S.: Cognition, Evolution, and Behavior, Oxford University Press, 1998
12. Turing, A. M.: On computable numbers, with an application to the Entscheidungsproblem, Proc. London Math. Soc., Vol. 42-2, pp. 230-265, 1936; A correction, ibid., Vol. 43, pp. 544-546, 1937
13. Turing, A. M.: Systems of logic based on ordinals, Proc. London Math. Soc. Series 2, Vol. 45, pp. 161-228, 1939
14. Verbaan, P.R.A., van Leeuwen, J., Wiedermann, J.: Complexity of Evolving Interactive Systems. In J. Karhumäki et.al., Eds., *Theory Is Forever,* Essays Dedicated to Arto Salomaa on the Occasion of His 70th Birthday. LNCS Vol. 3113, Springer-Verlag, Berlin, 2004, pp. 268–281
15. J. van Leeuwen, J. Wiedermann: On algorithms and interaction, in: M. Nielsen and B. Rovan (Eds), Mathematical Foundations of Computer Science 2000, 25th Int. Symposium (MFCS'2000), Lecture Notes in Computer Science Vol. 1893, Springer–Verlag, Berlin, 2000, pp. 99–112.
16. van Leeuwen, J., Wiedermann, J: The Turing machine paradigm in contemporary computing, in: B. Enquist and W. Schmidt (Eds.), *Mathematics unlimited - 2001 and beyond*, Springer-Verlag, 2001, pp. 1139–1155.
17. van Leeuwen, J., Wiedermann, J.: Beyond the Turing limit: evolving interactive systems, in: Proc. SOFSEM'01, LNCS Vol. 2234, Springer-Verlag, Berlin, 2001, pp. 90–109.
18. Wiedermann, J., van Leeuwen, J.: The Emergent Computational Potential of Evolving Artificial Living Systems. Ai Communications 15, 4, 2002, pp. 205-216.
19. Wiedermann, J., van Leeuwen, J.: Relativistic Computers and Non-Uniform Complexity Theory. In: Unconvential Models of Computation (UMC'2002), LNCS Vol. 2509, Berlin, Springer 2002, pp. 287-299