Machines, computations and time; when **P** can be equal **NP**

KAMIL KULESZA
Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK[1],
Institute of Fundamental Technological Research, Polish Academy of Sciences, Warsaw, Poland,
e-mails: *K.Kulesza@damtp.cam.ac.uk, Kamil.Kulesza@ippt.gov.pl*

Abstract.
The theory of computability is a mathematical abstraction that has proved to be very handy in explaining and foreseeing many facts about the computers, as we know them today. Yet, existing computers are objects embedded in our physical word and as such have to behave according to the laws of physics. This observation has had stunning consequences for our understanding of computations. For instance, it allowed proposing universal quantum computer and hence extending Turing-Church Thesis. Physics of computation has become a very active field of research, not only limited to quantum computing but also including other branches of physics like statistical mechanics. So far, not much has been known about possible links between models of computations and relativistic effects. In the paper we fill this gap, by the discussing role of a relativistic time factor in computations. At first we limit our consideration to the Turing Machine and show that in suitably chosen reference frames $P = NP$. We also discuss how this result can be obtained in the more general framework. Towards the end of the paper we outline an idea for a new, energy based, computational complexity measure and touch upon the halting problem.

1. Introduction

In 1936 Alan Turing published his seminal paper *"On computable numbers with an application to the entscheidnungsproblem"* ([1]) and introduced the notion of computable function. Although at the time of publication there were no digital computers as we know them today, the paper described a computer model which underlies all present designs. The Turing machine is a generally accepted model in computability theory and the foundation of all the classical computational complexity theory.

It is not the most efficient model of computation; however it is highly regarded for the fact the ability to simulate more feasible models. Actually, even stronger assertion, known as Church-Turing Thesis, exists: *Every 'function which would naturally be regarded as computable' can be computed by the universal Turing machine.*

Any problem, that can described in the finite number of bits and in a way compatible with the Turing machine, can be presented to the machine for consideration. It is hoped that after a finite number of steps the machine should produce the answer. There is however no guarantee for that and the machine can loop on the problem forever. Information whether the machine will eventually stop is often difficult to obtain forehand and is know as the *halting problem*. This problem was central to Turing's considerations, who in the true mathematical spirit was mainly concerned with showing possibility of the existence of finite solutions (the machine stops after finite number of steps) than their efficiency.

The latter problem was addressed by his successors by linking complexity of algorithm required to solve a given problem to the size of the input in bits.

In general there are two main measures of complexity. The first is *space complexity* which describes memory requirements of the algorithm, the second *time complexity* which describes

[1] Part of the work described in this paper was done when the author was visiting scholar at DAMTP and participant in Logic and Algorithms program at Isaac Newton Institute for Mathematical Sciences in Cambridge.

the number of computational steps needed to tackle a problem. In this paper we will be mainly concerned with the latter.

The questions (and initial answers) on efficiency can be traced back at least to the 1950's and John von Neumann [2]. Further quest for answers stimulated research, which resulted in modern computational complexity theory and a whole zoo of computational complexity classes (for most recent treatment see [3]).

For the sake of our presentation we describe polynomial-time algorithms, a notion was first introduced in [4] and [5]. If the solution of a problem requires polynomial number of steps in relation to the size of an input, then the problem falls into this category. Polynomial-time algorithms are considered "good" or tractable (also see Feasibility Thesis by Stephen Cook [6]). In contrast exponential-time algorithms are infeasible and, hence often called intractable.

We informally describe two classes that are subject to the fundamental question in computational complexity. The notion of polynomial-time underlies definition of the class **P**, which in computational complexity is defined in terms of Turing machine and refers to decision problems that can be solved by deterministic algorithm in polynomial time. In contrast, the class **NP**, groups decision problems that require more powerful, nondeterministic machines in order to be solved in the polynomial time. Best know solutions for such problems on the Turing Machine have algorithms with exponential time complexity.

Although both classes seem distinct, still today it is not known whether they really are.

The **P** versus **NP** problem is central to modern computer science and mathematics and as such is one of seven problems included in the Millennium Grand Challenge in Mathematics announced by the Clay Mathematics Institute, see [7]. Finding feasible **NP**-solver (e.g. by constructive proof that $P = NP$) would transform computer science as we know it, for instance allowing jump progress in artificial intelligence. It would also change mathematics since such a result would enhance techniques for formal proofs and automated theorem proving.

However, the problem's importance stems not only from the theoretical considerations, but also from related practical applications. Many of **NP** class problems have numerous industrial and business applications in virtually every field that quantitative methods are used. A very good example is modern cryptography which is largely based on the assumption that $P \neq NP$.

For the present state of art on **P** versus **NP** problem the interested reader is referred to a recent paper by Steven Cook ([8]) who actually first outline the problem over 30 year ago.

So far, we have been concerned with mathematical theory of computation, neglecting the physical aspects of the process. This concern is nicely expressed in the following quote *"Though the truths of logic and pure mathematics are objective and independent of any contingent facts or laws of nature, our knowledge of these truths depends entirely on our knowledge of the laws of physics."* by David Deutsch, Artur Ekert and Rossella Lupacchini in [9].

In fact such way of reasoning has proven quite seminal over the years. The most notable field is quantum information and quantum computing, which were first discussed in their full extent by David Deutsch in [10]. Many people consider it to be a great change in computation theory with Turing's universal Turing machine being replaced by Deutsch's universal quantum computer.

So far the results closest to practical applications have been achieved in quantum cryptography with some fielded solution already available (for detail see [11], [12], [13]). It should be noted that quantum computing inspired links between other branches of physics and computational complexity; a very good instance being statistical mechanics, see [14]. As a result, various concepts taken straight from physics toolbox, like phase transition and the study of threshold phenomena, were applied in the field of computability.

When it comes to foundational issues in modern physics, often some of Richard Feynman's ideas, usually in the form of cute quotes, are applicable. This has happened so often that his sayings won him the status of the "Oscar Wilde of Physics", as was nicely summarized by Vladko Vedral [15]. Feynman had his share in fostering a quantum computer, [16]. However, it was not the only contribution to the field that he made. He was discussing computers' miniaturization and density of information storage in the late 1950s, see [17]. At the time his ideas were considered interesting but rather purely theoretical speculations, although he get a little more practical by stating his famous challenge *"Why cannot we write the entire 24 volumes of the Encyclopaedia Britannica on the head of a pin?"* (see [17]).

Since then we have witnessed about 50 years of phenomenon known as Moore's law. The computing power and capacity of computer memory has been doubled roughly every 18 month, resulting in continuous exponential growth. Since Moore's law is not a law of nature, this prompted questions about the limits of the exponential grow. Computers are physical systems, hence their limits arise from the laws of physics. Some of the studies go as far as investigating term logic for physically realizable models of information, [18].

As the physical boundaries of computations one usually considers: a) energy limits on speed of computation, b) entropy limits on memory space, c) size limits on parallelization. An excellent discussion of these issues is contained in Seth Lloyd's article [19]. It should be noted that such limits are usually studied with the help of quantum mechanics applied to an ultimate machine possible only in some *Gedankenexperiment* (thought experiment).

So far, the time in computational complexity has been predominantly considered in the Newtonian way and relativistic effects connected with time have been largely neglected. On the popular level the topic was briefly discussed by Roger Penrose in "Shadows of the Mind" ([20]), where he considered the Gödel universe with close time-like lines and argued that causal violations would help to tackle the halting problem. However, it was David Deutsch that first considered using universal quantum computer in such conditions. In [21] he investigated nature of quantum mechanic near closed time-like lines and discussed possibility of removing paradoxical constrains imposed by such space-times. While discussing novel quantum mechanical effects, he also elaborated on possibilities for further enhancing quantum computers, if operated under such conditions. More recent research results on quantum computers near closed timelike curves, can be found in [22]. We came across Deutsch's and his followers works while polishing the final draft of this paper. It was an interesting experience to read Deutsch work and compare our ideas.

Their results are interesting and deep, however, we feel that one does not need to go that far as quantum computers, to see whole range of interesting effect resulting from non-Newtonian perception of time. The advantage of such an approach stems from the fact that we do not need to use universal quantum computer as model of computation. The argument can still be carried our for the Turing Machine, which makes the our presentation palatable also to researchers supporting more traditional approach to the theory of computability.

In the following sections we are going to discuss consequence of time related relativistic effects on our understanding of models of computations.

2. Time factor and computability

"The theory of computation has traditionally been studied almost entirely in the abstract, as a topic in pure mathematics. This is to miss the point of it. Computers are physical objects, and computations are physical processes. What computers can or cannot compute is determined by the laws of physics alone, and not by pure mathematics." – attributed to David Deutsch.

So far, in computational complexity time has been very much Newtonian. Let's see what changes can be introduced by Einstein, even if we restricted our consideration only to the

Turing Machine. Taking the above quote seriously, in this section, we ask a question: what impact on the process of computation effects from the theory of relativity can have?

2.1 Twin paradox

We start from the famous *Gedankenexperiment* in special relativity known as the Twin paradox, for instance see [23]. In the original formulation, the experiment concerns a pair of twins. One of them (twin A) stays on the Earth, while the other (twin B) undertakes a long journey with a rocket at almost the speed of light (velocity v). The journey ends up on the Earth and when siblings compare the time that they measure (e.g. their age), they find that the measurements different. To be more precise, let T_A be time of the journey as measured by twin A and T_B the time measured by B respectively. Then they will find that

$$T_A > T_B . \tag{2.1.1}$$

This can be written as

$$T_A = \gamma T_B , \tag{2.1.2}$$

where γ is time dilatation factor and is described by the formula

$$\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}} \quad . \tag{2.1.3}$$

In our case we propose to replace twins by two identical Turing machines, M_A and M_B respectively. The machines are given exactly the same problem for computation. Let's further assume that it takes T_A to complete execution of the program and both machines are started exactly at the moment in time, when M_B starts its space journey. In such a case, when M_B returns to the Earth, it is still busy computing, but M_A has just obtained the result. This allows M_B to learn outcomes from M_A, instead of continuing computations on its own.
So, in principle it is possible that one machine can obtain result of computations before actually finishing them. The point is that the time complexity measure is not something absolute and should be connected with the frame of reference. Now, it is the time to examine what speedup in computations can result from this observation.

2.2 Collapsing NP to P

"Everything should be made as simple as possible, but not simpler." – Albert Einstein.

In this section we are still in the realm of special relativity and *Gedankenexperiment* as outlined in the preceding section.
Now, assume that we set up thought experiment in the same way as described in the preceding section. We have two identical machines and send one of them into outer space. However, this time our machines are fed two different programs.
Travelling machine M_B is now computing the problem B with time complexity described by the function $f_B(n) = n^y$, where y is some constant natural number. Under such condition one can quite comfortably place problem B into polynomial-time and tractable domain. If proper care is taken in the specific description of the problem B, it can easily fall into class **P**.
Earth bound machine M_A is assigned different task. It tackle the problem A with time complexity described by the function $f_A(n) = x^n$, where x is some constant natural number.

Hence problem A is an exponential-time and intractable. Problem A, if properly defined, can belong to class **NP**.

The equation (2.1.3) relates times registered by A and B by time dilatation factor γ. The value of γ depends only on v, the relative velocity of A and B. It does not depend on any other parameters; especially it is independent of n (number of bits in the input). In principle, time dilatation can be made arbitrarily large as relative velocity v approaches the speed of light c. Note that for any finite n, exist γ such that

$$f_A(n) = \gamma f_B(n). \tag{2.2.1}$$

This can be written as

$$\frac{1}{\gamma} f_A(n) = f_B(n). \tag{2.2.2}$$

So, time-complexity of both problems can be the same in suitably chosen reference frames. In other words problems from **NP** class can be solved in the polynomial time!

It means that in principle exponential time speedup in computations is possible. This results in exponential reduction between **NP** and **P**, so **NP** can be collapsed to **P**. So, with the proper choice of reference frames $P = NP$.

2.3 Beyond the twin paradox

Conclusions of the previous sections are strong, however they are applicable only to a very special case of twin paradox. Sceptics would point out that it is only one case from special relativity. Also one would claim that the provided example is not particularly useful, since it provides us with the relative "slowdown" of M_B, not a "speedup" of M_A.

Let's start from the last problem. Can it be doctored? Well, M_A and M_B are not inertial observers, so dilatation effect is not reciprocal, see [23]. Hence, some other example should be found, such that travelling machine M_B will "record" more time steps than the Earth bound machine M_A. According to our best knowledge, finding such motion is not possible on the grounds of the special theory relativity. However, author is very keen to be proven wrong on that point.

So what about general relativity then? The answer is that in some metrics such results are possible. One example would be a Gödel universe, as proposed by Roger Penrose and David Deutsch for tackling the halting problem, see [20], [21].

This is a good news, since large scale physics of our universe, as we perceive it today, is described rather by general relativity theorem than its special case. The last sentence would obviously provoke reaction from all research communities convinced that general relativity is not good enough. Dealing with this problem is beyond the scope of the paper, although we used the theory of relativity to demonstrate our reasoning. Our main point is that the effects from the theory describing time metric (say the string theory if needed) have to be taken into account while considering realistic models of computation. These effects are not negligible, because they might cause **NP** collapse to **P** with suitable choice of reference frames.

3. Discussion, conclusions and further research

"Good mathematicians see analogies between theories, the very best see analogies between analogies"– Stefan Banach.

Computers are physical systems, so in describing them we cannot neglect the laws of physics, even those resulting from relativity theory. The point is that the time complexity measure is not something absolute and should be connected with the frame of reference. Having shown that given proper choice of reference frames it is possible to have $P = NP$, we call on seasoned relativists to help in finding more examples of this kind. The motivation for this call comes from the awareness of complexity of problems tackled in general relativity and the knowledge that many of them require years of dedicated studies.

Being positive about the existence of possible solutions does not however answer questions on their feasibility. At this point we try to address some of them. Probably the majority of objections would relate to the "energy bill". So far, we discussed some *Gedankenexperiment* involving accelerating large bodies to relativistic velocities. Such activity is obviously very energy consuming. However, it should be pointed out that actual computing device can be made very small as recent developments show. This is especially true when it comes to specialized machines, not general purpose computers. One of the leading and often very well funded applications is the special-purpose hardware for attacking cryptographic systems. Such devices are often relatively simple, but have to perform countless iterations of the same simple routine. Such parameters are very handy for the proposed agenda. Simple machine means minimal mass that has to be accelerated. Assume scenario that a machine is sent in to outer space and we do not need to retrieve it as long as it communicates results of the computations. It is not very difficult to visualize applications, whether in cryptography or automated theorem proving, that such *one-time* machines can be used. In such a scenario the program is hardwired into the physical structure of a machine, machine is sent to outer space, computations are performed, result sent back to us and we can forget about a machine, if we wish.

It should be mentioned that obtaining relativistic effects in computational complexity does not place any particular requirements on the model of computation. Hence, in principle also universal quantum computer can benefit from this enhancement (e.g., [21], [22]), making computation one of the fields where long awaited meeting between quantum and relativistic worlds takes place ☺.

Actually, one can easily imagine computing machine consisting with small number of particles, for instance heavy-ion system ([19]) or the plasma that "computes itself" ([24]). In such case it might happen that there will be no need to send our machine for relativistic journey to outer space, since achieving proper conditions might be feasible on the Earth with help of powerful accelerators.

As we mentioned above in order to have exponential reduction between **NP** and **P**, one has to introduce relativistic effects. It seems that the energy required to do that might be increasing exponentially with the required level of such effects. This will be certainly the case, should it be required to accelerate particles to relativistic velocities.

Taking this fact into account results in two observations:

1. Although $P = NP$ (with suitable choice of reference frames), nonetheless obtaining solution for the problem from the **NP** class might require much more, in fact even exponentially more, energy in comparison to solving problems from **P**. As we have shown, once relativistic effects come to play, the time-complexity, or the number of steps, ceases to be a good measure for algorithm performance. Hence, we see the need for a new measure which links algorithmic complexity with the energy required to complete it. Although the role of thermodynamics in computation is quite well studied (e.g., [19], [14]), in the relativistic setup the author would be very cautious using a simple measure linking the number of steps and energy cost of bit flipping. So, we think that interplay between energy and information in computation, should be re-examined taking relativistic effects in to account. A proper design of the postulated measure is left as an open problem.
2. So far we were mainly concerned with complexity measures, however it seems that it might also be possible to address the halting problem. In order to see it, consider the twin paradox as described in Section 2.1. Machine M_A is left on the Earth, running

the algorithm about we wish to know whether it terminates. The observer undertakes a relativistic journey. From the equations (2.1.2) and (2.1.3) it follows that he can delay the time of return almost to infinity (in machine M_A frame of reference). Obviously in order to check whether M_A has stopped he has to return after a finite time. In principle, however, T_A can be as long as he wishes possibly limited only by the lifespan of the Universe. This opens theoretical possibilities for probabilistic testing, with the reasoning that links T_A, with the probability that M_A would ever stop. Assuming the finite lifespan of the Universe, checking M_A just before the end of it would be a good indicator. If M_A is still running, with very high probability it will not stop, at least in this Universe. There is one more thing that should be taken into account in this joyful *Gedankenexperiment* – the energy bill. The energy required to experience infinite time dilatation will be infinite, too. However, observer's return just before the Great End, will require very high, but always a finite amount of energy. To on a more serious note, we observe that an exponentially growing energy is required to attack the halting problem "by twin paradox" while increasing the probability of getting the right answer. The other solutions, for instance ones resulting from the general theory of relativity, might however have a more handy energy profile.

We conclude with reference to "Programming the Universe: A Quantum Computer Scientist Takes On the Cosmos", a recent book by Seth Lloyd [25]. In the book he postulates the Universe that is a large computationally universal machine, where everything that surrounds us is the result of quantum computation, a long interplay between information and energy. Even if he is right, taking into account likely energy requirements, it might be required that one has to use such a machine at "100% processor's capacity" to properly tackle the halting problem. However, there is hope that we can do better than that.

Acknowledgement

The author wishes to thank Artur Ekert, Maciej Dunajski, Ruth Williams, Adrian Kent, Mosche Vardi and Tim Pedley for interesting discussions and encouragement. Great thanks to Kiran Singh for reading this manuscript and providing useful remarks and suggestions. In addition special thanks go to the library staff at Isaac Newton Institute for Mathematical Sciences for their help and patience with a notorious "poacher".

Bibliography

[1] A. Turing. "On computable numbers with an application to the Entscheidnungsproblem". Proc. London Math. Society, ser. 2, 42, pp.230–265, 1936-7.

[2] J. von Neumann. "A certain zero-sum two-person game equivalent to the optimal assignment problem. In H. W. Kahn and A. W. Tucker (Eds.), Contributions to the Theory of Games II, Princeton University Press, 1953.

[3] O. Goldreich. "Computational Complexity: A Conceptual Perspective". To appear: Cambridge University Press 2008. Preliminary version available from http://www.wisdom.weizmann.ac.il/~oded/cc-book.html

[4] A. Cobham. "The intrinsic computational difficulty of functions". In Yehoshua Bar-Hillel (Ed.), Proceedings of the 1964 International Congress for Logic, Methodology, and Philosophy of Science, pp. 24–30. Elsevier/North-Holland, 1964.

[5] J. Edmonds. “Minimum partition of a matroid into independent subsets”. J. Res. Nat. Bur. Standards Sect. B, 69, pp. 67–72, 1965.

[6] S. Cook. “Computational complexity of higher type functions”. In Ichiro Satake, editor, Proceedings of the International Congress of Mathematicians, Kyoto, Japan, pp. 55–69, Springer–Verlag, 1991.

[7] A. Jaffe. “The Millennium Grand Challenge in Mathematics,” Notices of the American Mathematical Society, 53 , pp. 652—660, 2006.

[8] S. Cook. “The P versus NP Problem”. Official Problem Description at Clay Mathematics Institute website: http://www.claymath.org/millennium/P_vs_NP/pvsnp.pdf.

[9] D. Deutsch, A. Ekert, R. Lupacchini. “Machines, Logic and Quantum Physics”, Bulletin of Symbolic Logic 3 3, 2000.

[10] D. Deutsch. “Quantum theory, the Church-Turing principle and the universal quantum computer”, Proceedings of the Royal Society of London Ser.~A, vol.A400, pp. 97-117, 1985.

[11] A. Poppe, A. Fedrizzi, T. Lorünser, O. Maurhardt, R. Ursin, H. R. Böhm, M. Peev, M. Suda, C. Kurtsiefer, H. Weinfurter, T. Jennewein, and A. Zeilinger. “Practical quantum key distribution with polarization entangled photons”, Optics Express, vol. 12, 3865. 2004.

[12] N. Gisin, G. Ribordy, W. Tittel, H. Zbinden. “Quantum cryptography”, Reviews of Modern Physics, vol. 74, 145. 2002.

[13] D.G. Angelakis, M. Christandl, A. Ekert, A. Kay, S. Kulik (Eds.). “Quantum Information Processing. From Theory to Experiment”, Volume 199 NATO Science Series: Computer & Systems Sciences, IOS Press. 2006.

[14] A. Percus, G. Istrate, C. More (Eds.). “Computational Complexity and Statistical Physics”, Santa Fe Institute Studies in the Sciences of Complexity, Oxford University Press 2006.

[15] V. Vedral. “Quantum entanglement as a universal indicator of phase transitions”, talk at CQC, DAMTP, 1st of February 2006.

[16] R.P. Feynman. “Simulating physics with computers”. International Journal of Theoretical Physics, vol. 21 (1982), pp. 467-488.

[17] R.P. Feynman. “There's Plenty of Room at the Bottom” December 29th 1959 at the annual meeting of the American Physical Society, California Institute of Technology. Transcript: February 1960 issue of Caltech's Engineering and Science, available at http://www.zyvex.com/nanotech/feynman.html.

[18] S. Lindell. “A Term Logic for Physically Realizable Models of Information”. Chapter 8 in the book *The Old New Logic: Essays on the Philosophy of Fred Sommers*, MIT Press, 2005.

[19] S. Lloyd. “Ultimate physical limits to computation”, Nature, vol. 406, August 2000.

[20] R. Penrose. “Shadows of the Mind”, Oxford University Press, 1994.

[21] D. Deutsch. “Quantum mechanics near closed timelike lines”, Physical Review D, vol.44, pp. 3197 - 3218, 1991.

[22] D. Bacon. “Quantum computational complexity in the presence of closed timelike curves”. Physical Review A, vol.70, 2004.

[23] G. F.R. Ellis, R.M. Williams. “Flat and curved space-times”. Oxford University Press, 1988.

[24] H. Pagels. “The Cosmic Code: Quantum Physics as the Language of Nature”. Simon and Schuster, New York, 1982.

[25] S. Lloyd. “Programming the Universe: A Quantum Computer Scientist Takes On the Cosmos”. Alfred A. Knopf/Jonathan Cape, 2006.