CDMTCS Research Report Series

Counterfactual Effect, the Halting Problem, and the Busy Beaver Function

Cristian S. Calude
Michael J. Dinneen
Department of Computer Science
University of Auckland, New Zealand

Karl Svozil
Institut für Theoretische Physik,
University of Technology Vienna, Austria

CDMTCS-107
July 1999

Centre for Discrete Mathematics and
Theoretical Computer Science

Counterfactual Effect, the Halting Problem, and the Busy Beaver Function

(Preliminary Version)

C. S. Calude, M. J. Dinneen, K. Svozil

Abstract

Using the counterfactual effect, we demonstrate that with better than 50% chance we can determine whether an arbitrary universal Turing machine will halt on an arbitrarily given program. As an application we indicate a probabilistic method for computing the busy beaver function—a classical uncomputable function. These results suggest a possibility of going beyond the Turing barrier.

1 Introduction

One fundamental result of theoretical computer science is Alan Turing's proof (in [19]) that it is undecidable to determine whether a computer program will halt or not. This is formally known as the *halting problem*. We can restrict our attention to Turing machines since they are equivalent in computational power to any "conventional" computer [1, 4]. In this note we wish to illustrate that some recently observed properties of quantum physics, more precisely, the interaction-free measurements, can be appropriately used to *solve* the halting problem in a limited sense. Specifically, we demonstrate that with better than 50% chance we can determine whether an arbitrary universal Turing machine M will halt, where M's input tape contains an encoded program. As an application we show how to compute, in a probabilistic way, the busy beaver function, a well-known uncomputable function.

We are going to base our analysis on Elitzur and Vaidman interaction-free measurements [12] and the Quantum Zeno (sometimes also called watched–dog) effect discussed by Misra and Sudarshan [15]. With interaction-free measurements we can detect a "live-bomb" or "a piece of film" without even sending a photon to trigger the bomb or to expose the film (see for example Bennett [2] or Penrose [16]).

To illustrate this let us suppose that we are given a bomb which is either live or dead, and we want to determine, with some reasonable probability, this fact without detonating it. This problem classically has no solution (see Penrose

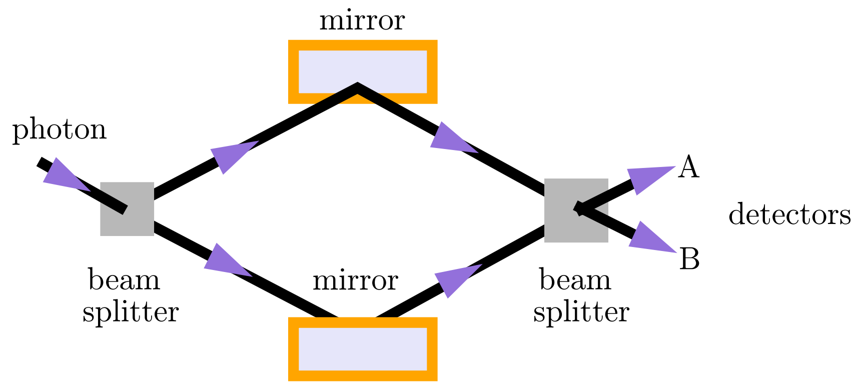

Figure 1: An interferometer.

[16]). However, there is a quantum mechanics solution! Consider a Mach-Zehnder interferometer, as illustrated in Figure 1, which is composed of two perfect mirrors and two 50-50 beam splitters. The upper and lower path lengths are set to be equal. Two detectors A and B determine the output superposition state of a photon. Under these hypotheses there is complete destructive interference to the lower exit port due to the fact that any incident light always exists the opposite port. The probability is 1 for a photon to reach detector B; see, for instance, Milburn [14] for a detailed explanation.

What happens when there is a bomb in one of the paths, e.g., as in Figure 2? First attach a special trigger device to the bomb which is activated by a photon. We then shoot a photon[1] through a beam splitter, where one possible path goes to the bomb and the other path goes to the deflecting mirror. In both cases the photon is reflected to a final switch splitter. A photon hitting detector B indicates that the photon was unaltered. A photon hitting detector A signals that a change has occurred in its quantum state (e.g., a bomb explosion).

If the bomb does not go off under the input photon, then we *know* that the photon took the upper path. At the second beam splitter the photon behaves as a single photon approaching a single beam splitter: in such a situation the photon takes a random choice between A and B.[2] We have got a 25% chance that the photon goes to B: this result contains no information since it would have happened also in the absence of the bomb.[3] There is a 25% chance that the photon is detected in A, and this is the *counterfactual miracle:* we can *see* a live bomb without any light reaching it.[4] Rephrased: whenever A goes "click" it signals the presence of an object in the interferometer. If we carefully send only

[1]Producing individual photons, one at a time, is not a problem; see for example [13].
[2]The non-explosion made two indistinguishable alternatives distinguishable!
[3]Note that this result of measurement does not disturb the bomb.
[4]The last situation never happens in the absence of the bomb.

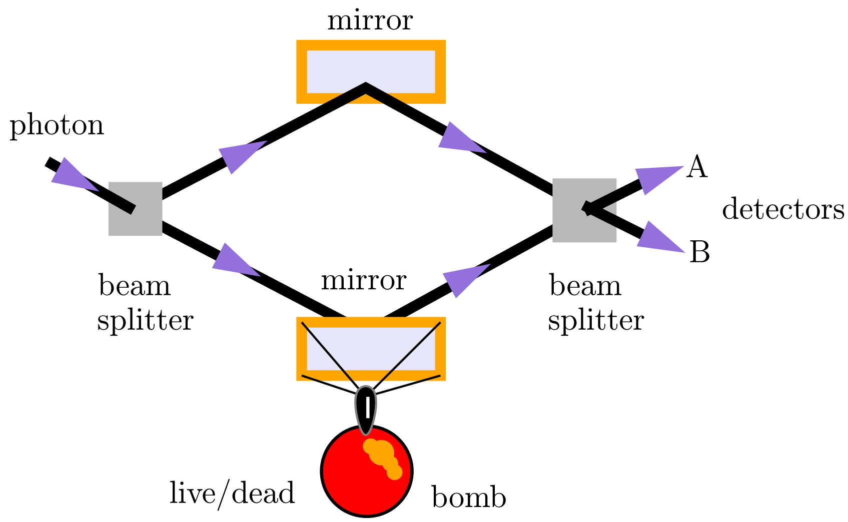

Figure 2: A counterfactual logic example: detecting live bombs.

one photon at a time, the detection of it in A means that it has not interacted with the object in the interferometer (we assume that a bomb which does not explode is non-transmitting). The price paid was to make B uncertain (compare with the scheme without a bomb).

Suppose now that the probability of having a live bomb is p. If the photon splits one half of the time towards the mirror and one half of the time towards the bomb, an explosion will occur with probability $p/2$. A live bomb will be detected in A and B with the same probability: $p/2$. A dead bomb will send the input photon to B with probability 1. So, with probability $\frac{2-p}{2}$ the photon is detected in B (and, of course, the photon will end up in A with probability $\frac{p}{2}$). The counterfactual effect happens with probability $p/2$. For the simplest case $p = \frac{1}{2}$ we have one chance in four to detect the photon in B, that is, we can find out that the bomb is live without actually detonating it. Of course, the result is only semi-reliable and semi-destructive! If p tends to 1, then A and B will have 50% chances to be reached. If p tends to 0, then the counterfactual miracle disappears.

The performances can be improved in a few of directions; for instance by utilizing the quantum Zeno effect.

The wave-particle duality of quanta guarantees that the above scheme works. When the interferometer is empty, the two possibilities to get to a detector are indistinguishable: we get interference, and consequently, the photon behaves like a wave. If one of the paths is blocked, then there is only one way to reach a detector, so there is no interference, and consequently the photon behaves like a particle: it can only show up at the bomb or at detector A or at detector B, but never at more than one of these.

The bomb used to illustrate the counterfactual effect can be in fact replaced by a device which is not required to signal to the outside world the reception or non-reception of the photon in such a dramatic way. In fact the device may not signal at all its reception or non-reception of the photon. A slightly wobbly mirror can play the role of the device if it were pivotable under the impact of the photon, so dispersing the motion as friction (cf. Penrose [16], p. 270). It is the mere potentiality for the mirror to wobble that allows the photon to reach detector B. This fact led to variants of the bomb model which are very useful in computer science applications. One possible way is to replace the live/dead bomb with two superposition states, on/off, of a computer. The simplest model uses a Turing machine and has the photon's state altered if and only if the machine accepts its input. The counterfactual effect saves computer resources by having the opportunity to turn off the computer after the photon has been split (and hoping that the photon goes to the mirror route).

2 Main Results

In order to tackle the halting problem we have to take into account the running time of our experiment. If the photon has to wait around for the result of a computation, then its 'parallel counterpart' will also be delayed on the mirror path. In what follows we are interested in the possibility that a computation halts, and if true, in a "sign" of it, **not** the completion of the (halting) computation.

Consider a universal Turing machine U running some programs P. Our machine U has an output classical one-bit register i (see Figure 3) which is initially zero. The value of i will toggle between 0 and 1, depending on whether the computation of U on P is progressing towards the halting state. For a given input program P define the halting function:

$$H_U(P) = \begin{cases} 0, & U \text{ loops on input } P, \\ 1, & U \text{ halts on input } P. \end{cases}$$

We are ready to state our first result: *There exists a computer and physical environment which computes probabilistically the halting function H_U.*

Here is the argument. Consider the device consisting of the universal Turing machine U, a program P to be executed on U, and the output register i; see Figure 3. Assume that P halts with probability p and consider the experiment illustrated in Figure 4. The photon going through the lower path starts the computation on U and is connected to the output register i; any modification in i affects the trajectory state of the photon. An input photon splits one half of the time towards the mirror and one half of the time towards the universal Turing machine; in the last case it will start the computation $U(P)$. If $U(P)$ never halts and the initial value for i was 0, then running P on U will have no effect on i. If $U(P)$ eventually halts, then running P on U will determine a finite computation at the end of which $i = 1$; however, in this situation we know nothing about the duration of the computation! To avoid this we just wait a predetermined infinitesimal amount of time, not for the completion of

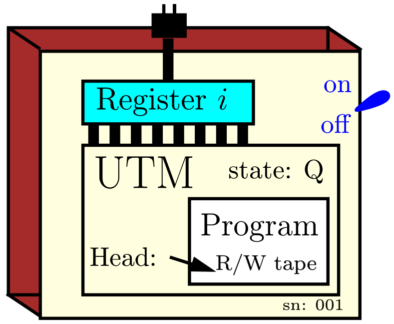

Figure 3: An universal Turing machine with a halting register.

the computation. If the computation stops within this interval we know for sure that $U(P)$ stops and i changes its value to 1; if the computation has not stopped in the given time, then we use the counterfactual effect: $U(P)$ stops with probability $\frac{p}{2}$.

Initially U is in state *off* and $i = 0$. The aim is to compute the halting probability of $U(P)$. If U does not halt on P, then we end up in B with probability $1 - \frac{p}{2}$. The fact that U halts on P can be detected on A with probability $\frac{p}{2}$. The quantum experiment is configured so that the final quantum state of the photon is based only on the potential value of register i. The mere potential starting of an eventually ending computation will be detectable by detector A. We do not have to wait until the *computation actually ends*! The "cause", i.e., the actual end of the computation does not have to be reached to produce the "effect", that is the hesitation detected by A.

We may ask ourselves the question: is this a realistic assumption? The computation of the halting probability of a universal self-delimiting computer (first done by Chaitin [9] [10]; see also Calude [5]) shows that the longer P becomes, the smaller its halting probability becomes; p tends to 0 as the length of P goes to infinity. So, only a tiny finite number of all programs may halt with a high probability.[5] Running the experiment repeatedly leads to the correct detection with probability about $\frac{p}{2} + \frac{p^2}{4} + \cdots (\frac{p}{2})^n + \cdots = \frac{p}{2-p}$. For $p = \frac{1}{2}$ we get about $\frac{1}{3}$; for p approaching 1 we obtain $\frac{1}{2}$. More importantly, for $p \geq \frac{2}{3}$ the asymptotic efficiency of the iterated procedure will be greater than $\frac{1}{2}$.

Of course, if p is pretty small, then we can run the same experiment using the hypothesis is that P eventually halts: $i = 1$. In this case the chance of computing the non-halting probability is now better: $\frac{1-p}{2}$.

Now that we have shown that the undecidable halting problem can be re-

[5]Davis [7] shows that if we tolerate one failure in a thousand of elementary instructions, and the probability of success in one such elementary instruction is $1 - \frac{1}{m}$, then we should not carry on more than $m/1000$ instructions. As the number of instructions executed by current computers is enormous, the chance of failure is far from infinitesimal in terms of lifetime probabilities.

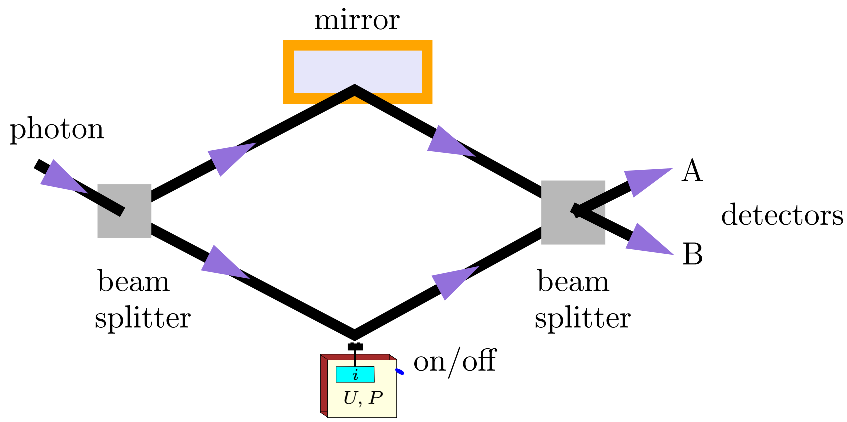

Figure 4: The schematics for (almost?) solving the halting problem.

alized with probability greater than $\frac{1}{2}$, we can easily apply the result to show that noncomputable functions can be computed in this physical environment. Consider the *busy beaver problem* as proposed by Rado in 1962 [18]. The busy beaver function $B(n)$ denotes the maximum number of 1's any n-state Turing machine over the two-symbol alphabet $\Sigma = \{0, 1\}$ may output on an initially blank (0-filled) tape.[6]

Our second result is: *There exists a computer and physical environment which computes the busy beaver function $B(n)$ with probability greater than $\frac{1}{2}$.*

Here is the argument. Assume that the halting problem can be computed with high probability $p_1 > \frac{1}{2}$. Let T_n denote the set of Turing machines with n states and alphabet $\Sigma = \{0, 1\}$. To compute $B(n)$ we simply decide whether each n-state Turing machine T in T_n halts with amplified confidence

$$p_2 > p_1^{\frac{1}{|T_n|}}.$$

This is obtained by using one or more of the performance techniques mentioned earlier. We then simulate on a conventional universal Turing machine each of those 'halting' T and note the largest output as the value of $B(n)$. This procedure is guaranteed to halt with probability greater than $p_2^{|T_n|} > \frac{1}{2}$ and will correctly compute the busy beaver function with probability greater than $p_2 \gg \frac{1}{2}$.

3 Conclusions

The computational procedures described in this paper are probabilistic (one can argue that all mathematical proofs or computer programs are ultimately

[6]The only known values are $B(1) = 1$, $B(2) = 4$, $B(3) = 6$, $B(4) = 13$, and $B(5) \geq 1915$.

probabilistic, see Davis [7], De Millo, Lipton, Perlis [8]), but go beyond the capability of any classical computation: even the best probabilistic algorithms are not able to achieve this computational power (by a classical result [11], probabilistic algorithms are equivalent to Turing machines).

References

[1] J. Barrow. *Impossibility—The Limits of Science and the Science of Limits*, Oxford University Press, Oxford, 1998.

[2] C. H. Bennett. Night thoughts, dark sight, *Nature* 371 (1994), 479-480.

[3] A. Berthiaume. Quantum computation, in L. A. Hemaspandra, A. L. Selman (eds.). *Complexity Theory Retrospective II*, Springer-Verlag, New York, 1997, 23–51.

[4] C. Calude. *Theories of Computational Complexity*, North-Holland, Amsterdam, 1988.

[5] C. Calude. *Information and Randomness — An Algorithmic Perspective*, Springer-Verlag, Berlin, 1994.

[6] C. S. Calude, J. Casti, M. J. Dinneen (eds.). *Unconventional Models of Computation*, Springer-Verlag, Singapore, 1998.

[7] P. J. Davis. Fidelity in mathematical discourse: Is one and one really two? *Amer. Math. Monthly* 79(1972), 252-263.

[8] R. De Millo, R. Lipton, A. Perlis. Social processes and proofs of theorems and programs, *Comm. ACM* 22(1979), 271-280.

[9] G. J. Chaitin. *Information, Randomness and Incompleteness, Papers on Algorithmic Information Theory*, World Scientific, Singapore, 1987. (2nd ed., 1990)

[10] G. J. Chaitin. *The Limits of Mathematics*, Springer-Verlag, Singapore, 1997.

[11] K. De Leeuw, E. F. Moore, C. E. Shannon, N. Shapiro. Computability by probabilistic machines, in C. E. Shannon, J. McCarthy (eds.). *Automata Studies*, Princeton University Press, Princeton, N.J., 1956, 183-212.

[12] A. C. Elitzur, L. Vaidman. Quantum-mechanical interaction-free measurements, *Foundat. of Phys.* 23 (1993), 987-997.

[13] P. Grangier, G. Roger, A. Aspect. Experimental evidence for a photon anticorrelation effect on a beam splitter: a new light on single-photon interferences, *Europhysics Letters* 1 (1986), 173-179.

[14] G. Milburn. *The Feynman Processor. An Introduction to Quantum Computation*, Allen & Unwin, St Leonard, 1998.

[15] B. Misra, E. C. G. Sudarshan. The Zeno's paradox in quantum theory, *J. Mathematical Phys.* 18 (1977), 756–763.

[16] R. Penrose. *Shadows of the Mind*, Oxford University Press, Oxford, 1994.

[17] K. Svozil. Halting probability amplitude of quantum computers, *J. UCS* 1 (1995), 201-203.

[18] T. Rado. On non-computable functions, *Bell System Technical Journal* 41 (1962), 877–884.

[19] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, Series 2, 42–43 (1936-7), 230-265, 544–546.