The Slowdown Theorem: A Lower Bound for Computational Irreducibility in Physical Systems

Jonathan Gorard
Department of Mathematics, King's College London
Algorithms R&D Group, Wolfram Research, Inc.
jonathan.gorard@kcl.ac.uk

Following from the work of Beggs and Tucker on the computational complexity of physical oracles, a simple diagonalization argument is presented to show that generic physical systems, consisting of a Turing machine and a deterministic physical oracle, permit computational irreducibility. To illustrate this general result, a specific analysis is provided for such a system (namely a scatter machine experiment (SME) in which a classical particle is scattered by a sharp wedge) and proves that it must be computationally irreducible. Finally, some philosophical implications of these results are discussed; in particular, it is shown that the slowdown theorem implies the existence of classical physics experiments with undecidable observables, as well as the existence of a definite lower bound for the computational irreducibility of the laws of physics. Therefore, it is argued that the hypothesis that "the universe is a computer simulation" has no predictive (i.e., only retrodictive) power.

Keywords: computational irreducibility; computational equivalence; computational complexity; Turing machines; physics; simulations; slowdown; experiments

1. Introduction

A system is described as being "computationally irreducible" if, loosely speaking, it is not possible to "shortcut" its behavior computationally, that is, if it is impossible to predict the behavior of the system using fewer computational steps than the system itself takes to evolve. It has been a longstanding conjecture of Wolfram that many systems in nature, including the physical universe, are computationally irreducible [1]. This conjecture arose from analyzing elementary cellular automata, such as rule 30, as a convenient idealization for physical systems, as in *A New Kind of Science* [2].

The goal of the present paper is to make this conjecture mathematically precise and to show that it can be proved rigorously for a large class of nonidealized physical systems, that is, systems that are not

https://doi.org/10.25088/ComplexSystems.27.2.177

based on cellular automata. We prove this very general result—the "slowdown theorem"—by means of a diagonalization argument, analogous to the one used by Turing in the proof of the undecidability of the halting problem [3].

Definition 1. If T is a Turing machine that computes some "definite" function $f : \mathbb{N} \to \mathbb{N}$ (i.e., T computes the value $f(i)$ for some fixed input i) in n steps, then we say that T's computation is computationally reducible if and only if there exists a Turing machine T^* that computes $f(i)$ in m steps, where $m < n$ [4, 5].

This algorithm T^* is a *speedup* algorithm of T [6]. In Definition 1, it is necessary for T and T^* to be the same *type* of Turing machine, since the relative efficiencies of different types of Turing machines are not always equivalent, as illustrated by Theorem 1.

Theorem 1. Given any k-tape Turing machine M, operating within time $f(n)$, we can construct a 1-tape Turing machine M', operating within time $O\left(\left[f(n)\right]^2\right)$, and such that for any input x, $M'(x) = M(x)$ [7].

For this reason, we shall henceforth assume (without loss of generality) that all Turing machines referenced in this paper are 1-tape Turing machines [8, 9].

Definition 2. If T is a Turing machine that computes some definite function $f : \mathbb{N} \to \mathbb{N}$ in n steps, and T^* is a Turing machine that computes f in m steps, then the degree of slowdown of T^*'s simulation of T is defined to be $m - n$.

It follows that, if a computation is irreducible, then any simulation of it must be subject to a non-negative degree of slowdown [10].

1.1 Background to the Slowdown Theorem

The slowdown theorem arose from the following thought experiment: suppose that the universe is a Turing machine, T. Now, construct a physical Turing machine T^* that simulates T. Since T^* is a physical system, we can think of it as encoding abstract computational states X as concrete physical states $f(X)$.

Since T computes the behavior of all physical systems in the universe, it must also compute the behavior of T^*. If T is computationally reducible, then it would be possible for T^* to shortcut the evolution of T, in order to determine what the physical state of T^* will be at some arbitrary point in the future.

Therefore, we can construct a simple diagonalization argument: if T^* predicts that its own physical state will be X at some point in the future, then we can program T^* to produce an output state corresponding to some different physical state Y, and so on. By construction, the system T^* contradicts any predictions that are made about its behavior by T. Thus, by contradiction, T cannot be computationally reducible.

The purpose of the remainder of this paper is to make this argument mathematically formal. We begin by presenting the argument in full generality for any physical system T^* containing a Turing machine T. Then, we analyze a particular type of physical system, first studied by Beggs and Tucker [11], consisting of a Turing machine T that consults a classical physics experiment E as an oracle (we also provide a specific analysis of such a system: namely a scatter machine experiment (SME) in which a classical particle is scattered by a sharp wedge). Finally, we discuss some philosophical implications of these results.

For the remainder of this paper, we shall assume that all physical systems are discrete, such that any encoding function that maps computational states onto physical states has the form $f : \mathbb{N} \to \mathbb{N}$.

2. The Slowdown Theorem

We consider some generic physical system T^* that contains a Turing machine T:

$$\mathrm{A} \longrightarrow \underbrace{\cdots \qquad \boxed{\mathrm{T}} \qquad \cdots}_{T^*} \longrightarrow \mathrm{B}$$

T can be connected to any finite collection of physical experiments E. T^* starts in some initial physical state A and ends in some final physical state B. T, on the other hand, takes an input C and produces an output D:

$$\mathrm{C} \longrightarrow \boxed{\mathrm{T}} \longrightarrow \mathrm{D}$$

The slowdown theorem states that:

Theorem 2. If T^* is the specification of a computation (acting on the input A and producing the output B), then there exists an instance of T^*, namely T_0^*, that is computationally irreducible.

Proof. We want to show that there exists a particular T^*, namely T_0^*, such that T_0^* is computationally irreducible. We suppose, conversely, that every T_0^* is computationally reducible; that is, that for every Turing machine T_0 and collection of physical experiments E_0, there exists a Turing machine T_0^{**} that, for any given A, correctly computes B, using fewer steps than T_0^* itself takes. Therefore, T_0^{**} can shortcut the physical evolution of T_0^* and predict what the final state will be before T_0^* itself reaches it.

Since T_0 is itself a physical system, C and D can be thought of as being the initial and final states of that system, respectively. Let $g : \mathbb{N} \to \mathbb{N}$ be the associated encoding function, such that $g(C)$ is the physical state of T_0 at the start of its computation, and $g(D)$ is the physical state of T_0 at the end of its computation.

Since T_0^* is a closed and deterministic classical system, information about T_0^*'s state must always be conserved (since Liouville's theorem implies that T_0^*'s behavior must be exactly reversible). Furthermore, since T_0 is a component of T_0^*, it follows that T_0's input state will be encoded within T_0^*'s input state, and T_0's output state will correspondingly be encoded within T_0^*'s output state. In other words, A must somehow encode all of the information about $g(C)$, and B must somehow encode all of the information about $g(D)$.

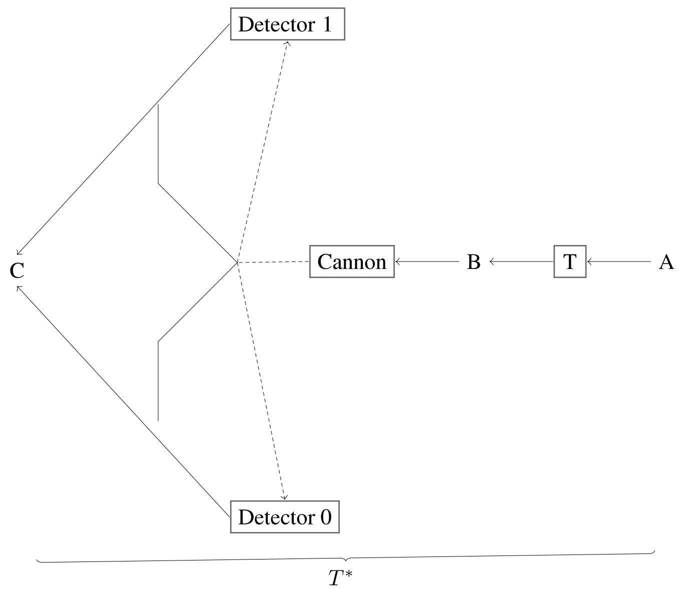
Thus, we let $f : \mathbb{N} \to \mathbb{N}$ be the associated decoding function, such that $f(A) = g(C)$, and $f(B) = g(D)$. Now, since we have assumed that T_0^* is computationally reducible for every T_0 and E_0, it follows that every T_0 that simulates T_0^* using the speedup algorithm T_0^{**} must also be reducible. This allows us to diagonalize over the space of Turing machine input pairs (T, A) and construct the following Turing machine T_0:

$$T_0(A) = \begin{cases} Y & \text{if } f\left(T_0^{**}(A)\right) = g(X), \\ X & \text{otherwise,} \end{cases} \tag{1}$$

where $X \neq Y$. In other words, T_0 uses T_0^{**} to preempt what the final state of T_0^* will be and then produces an output that forces T_0^* to end up in some different final state. Thus, by contradiction, the speedup algorithm T_0^{**} does not exist for T_0^*, and so there exists at least one such T_0^* that is computationally irreducible. □

3. The Scatter Machine Experiment

We consider a generic physical system T^* consisting of a Turing machine T that accepts an input A and outputs a natural number B. The value of B then determines the position of a cannon, which projects a particle at a sharp wedge:



After colliding with the wedge, the particle will either scatter upward (into Detector 1), or downward (into Detector 0), depending upon the position of the cannon. For the sake of simplicity, we shall henceforth assume that the experiment is configured in such a way that the particle never hits the point of the wedge, and so there is no value of B for which the experiment becomes nondeterministic.

We capture the details of the particle's trajectory with the single binary observable Y, which is either 1 (corresponding to the particle being scattered up) or 0 (corresponding to the particle being scattered down). We can think of T^* as being a computation (acting on the input A and producing the output observable C).

Beggs and Tucker proved Theorem 3 about the complexity of the SME experiment (in which the particle is allowed to hit the point of the wedge, and the experiment is therefore nondeterministic) in 2007.

Theorem 3. If T is a Turing machine with polynomial time, then T^* computes the nonuniform complexity class $P / poly$ 12.

We shall prove Theorem 4 regarding its computational reducibility (in which point collisions are not permitted).

Theorem 4. If T^* is Turing computable, then there exists an instance of T^*, namely T_0^*, that is computationally irreducible.

Proof. We want to show that there exists a particular T^*, namely T_0^*, such that T_0^* is computationally irreducible. We suppose, conversely, that every T_0^* is computationally reducible; that is, that for every T_0

there exists a Turing machine T_0^{**} that, for any given A, correctly computes the observable C, using fewer steps than T_0^* itself takes.

By the definition of the system T_0^*, the observable C is determined by the output of the Turing machine T_0, namely B. Therefore, we can define an encoding function $f : \mathbb{N} \to \mathbb{N}$, such that $f(B) = C$. Now, since we have assumed that T_0^* is computationally reducible for every T_0, it follows that every T_0 that simulates T_0^* using the speedup algorithm T_0^{**} must also be reducible. We diagonalize over the space of Turing machine input pairs (T, A) accordingly:

$$T_0(A) = \begin{cases} Y & \text{if } T_0^{**}(A) = f(X), \\ X & \text{otherwise,} \end{cases} \tag{2}$$

where $X \neq Y$. We have therefore produced an output of T_0, which forces the observable C to take a different value to that predicted by T_0^{**}. Thus, by contradiction, the speedup algorithm T_0^{**} does not exist for T_0^*, and so there exists at least one such T_0^* that is computationally irreducible. □

3.1 Implications

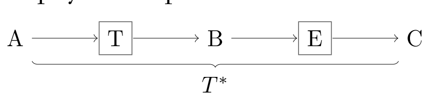
Clearly, the amount of time between the particle being fired and the particle being detected can be made arbitrarily large (for instance by moving the wedge further away from the cannon, moving the two detectors further apart, or by firing the particle at an arbitrarily low speed). Thus, it follows that:

Corollary 1. If T^* is Turing computable, then the number of steps required to compute the observable C, given the input A, can be unbounded (i.e., can be made arbitrarily large).

Therefore, T^*'s computation of C from A can be made to take an arbitrarily large number of steps, yet we have also shown that T^* can be computationally irreducible, which yields the following remarkable corollary:

Corollary 2. Determining the value of the observable C, given the input A, can (in general) be undecidable.

Note that these results would hold for any generic physical system T^* in which the output of a Turing machine T determines the input of some deterministic physical experiment E:



The SME is just a particularly elegant minimal example of such an experiment E for which computational irreducibility holds.

4. Some Philosophical Remarks

In order to evade the diagonalization argument presented in this paper, it is necessary to assert one of two things: either T^* is not Turing computable (i.e., T^* is a *hypercomputer*), or T^* can be computationally irreducible. Both of these possibilities seem to be, in many ways, philosophically objectionable.

We have established that the outcome of the experiment E is completely determined by the output of the Turing machine T. Yet, if T^* is a hypercomputer, then somehow the combination of T and E must be able to compute partial functions that could not have been computed by T alone. Though highly counterintuitive, this resolution would be consistent with the complexity-theoretic results of Beggs and Tucker.

On the other hand, it seems intuitively obvious that one could optimize T^* in the case of the SME system, thereby reducing the number of steps required to compute C (for instance, by making the particle travel more quickly, moving the wedge closer to the cannon, or moving the detectors closer together). However, if T^* is computationally irreducible, then it follows that no such optimizations can be possible (since they would imply reducibility of T^*). In other words, even if the experiment is made intentionally inefficient by adding arbitrary redundancies, it would not be possible to optimize it without somehow modifying the results: one would still need to simulate all of the redundancies and inefficiencies explicitly and in their entirety.

Furthermore, the fact that T^*'s (irreducible) computation can involve a theoretically unbounded number of steps implies that classical physics experiments can give rise to undecidable observables. More generally, it implies that there exist deterministic classical systems about which one can ask undecidable questions. One plausible candidate would be the n-body problem [2].

Conjecture 1. The question of whether a given body will ever escape from a gravitationally interacting n-body system is, in general, undecidable.

These arguments clearly do not apply to quantum-mechanical systems in which observables exhibit nondeterministic behavior following measurement.

If the universe is Turing computable, then the slowdown theorem implies that any simulation of it must be subject to a non-negative degree of slowdown. This immediately entails that the hypothesis "the universe is Turing computable" has no predictive power, since making predictions would require being able to simulate the universe in such a way as to shortcut its evolution, thereby determining the outcomes of events that will occur in the future. The arguments in this paper demonstrate that the construction of any such simulation would inevitably lead to logical contradictions, so such a hypothesis would only be testable through retrodiction.

Thus, the slowdown theorem implies that the only universes that are computationally reducible (and that can therefore be simulated without the effects of non-negative slowdown) are those universes in which the laws of physics are sufficiently restrictive that they do not permit the construction of universal Turing machines [13].

Acknowledgments

Thanks are due to Stephen Wolfram, John V. Tucker, Joseph McGovern and Jan Stańczuk for many illuminating conversations, at various stages during the development of these ideas. Thanks are also due to the reviewers and editors, whose comments and questions substantially improved the clarity, completeness and quality of the manuscript.

References

[1] S. Wolfram, "Undecidability and Intractability in Theoretical Physics," *Physical Review Letters*, **54**(8), 1985, pp. 735–738. doi:10.1103/PhysRevLett.54.735.

[2] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[3] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, **S2-42**(1), 1937 pp. 230–265. doi:10.1112/plms/s2-42.1.230.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco: W. H. Freeman, 1979.

[5] J. E. Hopcroft, R. Motwani and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, 3rd ed., London: Pearson, 2013.

[6] H. Zenil, F. Soler-Toscano and J. J. Joosten, "Empirical Encounters with Computational Irreducibility and Unpredictability," *Minds and Machines*, **22**(3), 2012 pp. 149–165. doi:10.1007/s11023-011-9262-y.

[7] H. Zwirn and J.-P. Delahaye, "Unpredictability and Computational Irreducibility," in *Irreducibility and Computational Equivalence: 10 Years after Wolfram's A New Kind of Science* (H. Zenil, ed.), Berlin, Heidelberg: Springer, 2013 pp. 273–295. doi:10.1007/978-3-642-35482-3_ 19.

[8] O. Goldreich, *Computational Complexity, A Conceptual Perspective*, New York: Cambridge University Press, 2008.

[9] C. H. Papadimitriou, *Computational Complexity*, London: Pearson, 1993.

[10] D. Reisinger, T. Martin, M. Blankenship, C. Harrison, J. Squires and A. Beavers, "Exploring Wolfram's Notion of Computational Irreducibility with a Two-Dimensional Cellular Automaton," in *Irreducibility and Computational Equivalence: 10 Years after Wolfram's A New Kind of Science* (H. Zenil, ed.), Berlin, Heidelberg: Springer, 2013 pp. 263–272. doi:10.1007/978-3-642-35482-3_ 18.

[11] E. J. Beggs and J. V. Tucker, "Experimental Computation of Real Numbers by Newtonian Machines," *Proceedings of the Royal Society, Series A: Mathematical, Physical and Engineering Sciences*, **463**(2083), 2007 pp. 1541–1561. doi:10.1098/rspa.2007.1835.

[12] E. J. Beggs, J. F. Costa and J. V. Tucker, "The Impact of Models of a Physical Oracle on Computational Power," *Mathematical Structures in Computer Science*, **22**(5), 2012 pp. 853–879. doi:10.1017/S0960129511000557.

[13] R. Rucker, "An Incompleteness Theorem for the Natural World," in *Irreducibility and Computational Equivalence: 10 Years after Wolfram's A New Kind of Science* (H. Zenil, ed.), Berlin, Heidelberg: Springer, 2013 pp. 185–198. doi:10.1007/978-3-642-35482-3_ 14.