

- [20] Wolpert, D. (1990). The relationship between many-to-one mappings and thermodynamic irreversibility. Los Alamos Laboratory report LA-UR-90-4108.
- [21] Wolpert, D. (1994). On the computability of the future. Los Alamos Laboratory Report LA-UR-91-3344.
- [22] Zurek, W. (1988). Algorithmic randomness and physical entropy. Los Alamos Laboratory report LA-UR-88-1547.
- [23] Moore, C. (1991). *Nonlinearity*, **4**, 199.
- [24] Pour-El, M., and Richards, I. (1982). *International Journal of Theoretical Physics*, **21**, 553.
- [25] Kanter, I. (1990). *Physical Review Letters*. **64**. 332.
- [26] Farmer, J., and Sidorowich, J. (1988). Exploiting chaos to predict the future and reduce noise, Los Alamos Laboratory report LA-UR-88-901.

ACKNOWLEDGMENTS This work was done under the auspices of the Department of Energy and the Santa Fe Institute. I would like to thank Bill Macready, Cris Moore, Paul Stolorz, Tom Kepler and Carleton Caves for interesting discussion. This work was originally presented at the Third Santa Fe Workshop on Complexity, Entropy, and the Physics of Information, 1994.

the real world we cannot set up and then perform arbitrary computation backwards in time. See [19].

REFERENCES

- [1] Bennett, C.H. (1973). IBM Journal of Research and Development, **17**, 525-532.
- [2] Bennett, C. H. (1982). International Journal of Theoretical Physics, **21**, 905.
- [3] Bennett, C. H. (1987). Scientific American. **257** (11), 88.
- [4] Bennett, C. H. (1988). IBM Journal of Research and Development, **32** (no. 1), 16-24.
- [5] Berger, J. (1990), International Journal of Theoretical Physics, **29**, no. 9, 985-995.
- [6] da Costa, N., and Doria, F. (1991). International Journal of Theoretical Physics, **30**, 1041.
- [7] Feynman, R. (1986). Foundations of Physics, **16**, 507.
- [8] Fredkin, E., and Toffoli, T. (1982). International Journal of Theoretical Physics, **21**, 219.
- [9] Geroch, R., Hartle, J. (1986). Foundations of Physics, **16**, 533.
- [10] Landauer, R. (1961). IBM Journal of Research and Development, **5**, 183.
- [11] Landauer, R. (1967). IEEE Spectrum. September, 105.
- [12] Landauer, R., (1971). Ferroelectrics, **2**, 47.
- [13] Landauer, R. (1986). Foundations of Physics, **16**, 551.
- [14] Landauer, R. (1988). Nature, **335**, 779-784.
- [15] Landauer, R. (1991). PHYSICS TODAY, May, 23.
- [16] Lloyd, S. (1990). Any nonlinearity suffices for computation. Cal Tech. report CALT-68-1689.
- [17] Wheeler, J.A., and Zurek, W.H. (1983). Quantum theory and measurement. Princeton University Press, Princeton, NJ. 785-786.
- [18] Wolpert, D. (1992). PHYSICS TODAY, 98, March.
- [19] Wolpert, D. (1992). International Journal of Theoretical Physics, **31**, 743-785.

in (1) and (2) it is forced to a chaotic trajectory? Or perhaps instead, the only kinds of computers one might think could predict any external system are everywhere chaotic, so in both the proof of (1) and (2), one is trying to predict a chaotic system, and that's why you fail? (Note that if this were true, it would rule out 100% efficient error-correcting computers as universal predictors, since they are designed to not be chaotic.)

There are reasons to believe this is not the case. For example, the results in this paper don't simply say that the computer is unreliable, sometimes getting the correct answer, sometimes the wrong answer. Rather they say that there is a scenario where the computer is *always* wrong. This makes it hard to see how the "explanation" for these results could lie in chaotic or quantum mechanical properties of the computers - one would expect such properties to give unreliability, not reliable incorrectness.

FOOTNOTES

[1] This issue is intimately related to memory systems theory, in that "remembering" an event from the past, formally speaking, means predicting that event accurately, using only information from the present. See [19].

[2] Actually, one doesn't need to invoke the psychological arrow to conclude that the reading of the result of the computation must come after the computation's start, not before it. This is because in doing computation we want to be able to initialize the computer to any starting configuration we desire, regardless of its state at times not during the computation. (Such times are before the computation starts for computation forward in time, and "after it starts" for computation going backward in time.) Such initialization constitutes a many-to-one mapping of the state of the computer, and a many-to-one mapping going backward in time violates the second law. See [20]. Consequently, despite the reversible nature of computation ([1-5, 8, 10-12, 15, 17, 20]), in

which do not rely on both having A “observe” C and C “observe” A? Or if we do allow such two-way observation, what happens if the times the observations end are not both $t = 0$? In this latter case, does the proof of (2) still go through if one simply allows one’s computer to observe itself (so it can know what the other computer will observe)?

iv) What restrictions hold if τ can depend on J_0 and/or ϑ_T as well as on T ? I.e., what happens if the time of the computer’s output depends on the task it is assigned? (In the proofs of (1) and (2), the construction of A - and therefore J_0 - depends on τ , but not vice-versa.)

v) To what degree is it true that a computer can always correctly predict the future of a system that is “simpler” than that computer? In particular, in the context of (1), are there scenarios where A does not include the computer doing the predicting, and despite the fact that there is a sufficient J_0 , the computer must make an error? (In the context of (2), the question is instead whether the computer must make an error despite being coupled to A before $t = 0$.)

vi) How are these results transformed for analog computers (so that one is concerned with amounts of error rather than whether there is an error)? What about if one is calculating the future state of a stochastic system, i.e., if one is predicting a probability distribution? How are the results transformed for quantum systems?

vii) Find the exact point of failure of the intuitive argument “If the computer is simply a sufficiently large and fast Hamiltonian evolution approximator, then it can emulate any finite classical non-chaotic system”.

viii) A related issue: Might it be that even though evolution of the combined C-A system in the overall phase space isn’t chaotic (i.e., the region of that phase space with positive Lyapunov exponent has measure 0), that any time a computer actually tries to perform a computation of the sort

the psychological arrow of time (which follows from the second law), we are only interested in computations which start with the input data and finish *at a later time* with the prediction.² Now in our time-reversed scenario, the “initial information” (J_0 in (1)) is at time $t = 0$, and the time of the information being predicted (said information being J_T in (1)) is at time $t = -T$. So the psychological arrow forces $\tau > 0$, which in turn means that $\tau \notin$ the time interval stretching from the time of the initial information to the time concerning which we wish to make a prediction. Accordingly, predicting the past can not be formulated as the situation referred to in (1) simply with opposite signs on all of the temporal dependencies.

There are many other issues that the results of this paper raise:

i) One might try to characterize (1) as the physical analogue of the following issue in Turing machine theory: Can one construct a Turing machine M which can take as input A , an encoding of a Turing machine and its tape, and for any such A computes what state A will be in after n steps, and perform this computation in fewer than n steps? This suggests investigating the formal parallels (if any) between the results of this paper and the “speed-up” theorems of computer science.

ii) More speculatively, the close formal connection between the results of this paper and those of computer science theory suggest that it may be possible to find physical analogues of most of the results of computer science theory, and thereby construct a “physical computer science theory”. In particular, it may be possible to build a hierarchy of physical computing power, in analogy to the Chomsky hierarchy.

As an example, we could define a (physical) “oracle” as a device that *can* correctly predict the future state of the universe from any current state of the universe, before that future state occurs. The behavior of such a device is perfectly well-defined. Nonetheless, by the results of this paper we know that such a device lies too high in the hierarchy to exist in the physical universe.

iii) Are there results similar to (2) concerning scenarios in which A and C are isolated after $t = 0$

for $t < 0$ results in the computer correctly predicting J_T before time T . In this, whereas (1) does not even mention observation, (2) can be viewed as a restriction on observation (a restriction in some ways reminiscent of the Heisenberg Uncertainty Principle) as readily as one on computation.

DISCUSSION

As a special case of all this, we can let A be the entire universe: even if the universe were finite, purely classical, and did not follow chaotic dynamics, and even if there were no physical limits to the size or speed of a computer, or to the spatial density of a computer's operating units, no computer which we could use could run a program which would correctly predict the future of the universe (ahead of time), for arbitrary initial conditions. In other words, even if Laplace were correct and the universe were essentially a giant clock, he still would not be able to predict the future state of that clock, for arbitrary current states. This casts an interesting light on the ideas of Fredkin, Landauer and others concerning whether or not the universe "is" a computer, whether or not there are "information-processing restrictions" on the laws of physics, etc. [8, 15].

As another example, we can have the "computer" referred to in (1) and (2) be a human mind working in concert with an electronic computer. In this context, (1) and (2) say that there are finite, classical systems whose state we humans, even with the aid of electronic computers, can not compute ahead of time. In particular, (2) says that there is no time-series prediction computer [26] that can reliably predict the future of systems powerful enough to contain a copy of that computer, regardless of noise levels and the dimension of the underlying attractor.

On a somewhat less formal note, one can observe that the proof of (1) does not overtly rely on the second law of thermodynamics, which is the only time-asymmetric law in the physical universe. Accordingly, one might wonder if by simply time-reversing all of its steps the proof of (1) directly establishes that one can not compute the past, as well as the future.¹ Although this issue isn't fully resolved, it would appear that the answer to the question is no. The reason is that, due to

opposite. In other words, the ball and sphere are the logical NOT operator applied to $\text{output}_B(\tau'')$.

Assume that the ball, sphere, B and C never interact with anything else in the universe. For the computer C, A consists of {B, sphere, ball}. For the computer B though, A consists of {C}.

Let the time-to-be-predicted (i.e., the "T") for the computer C be $t = 1$. Have $\vartheta_1(A = \{B, \text{sphere, ball}\})$ be {the same state-of-the-ball at $t = 1$ } as in the proof of (1); C takes this T and ϑ_T as its input at $t = 0$. By hypothesis, if for times $t < 0$ C is coupled to {B, sphere, ball}, then C's output at time τ'' is $J_1(B, \text{sphere, ball})$. This is true regardless of what the actual state of {B, sphere, ball} at $t = 0$ is.

Have $\vartheta_{\tau''}(A = \{C\})$ be {the state of C's output at time τ'' }. By hypothesis, if at time $t = 0$ B is fed $J_0(C)$, $\vartheta_{\tau''}(A = \{C\})$, and a specification of B's "T" as τ'' as input, then B's output at any time after $\tau' \equiv \tau(\tau'') \in [0, \tau'')$ is $J_{\tau''}(C)$. (Note that B's values of "T" and " τ " are τ'' and τ' , respectively.)

If B's prediction is incorrect the theorem is proved. If on the other hand B's prediction of C's prediction is correct, then the state of the ball at $t = 1$, which is the opposite of B's prediction, will be the opposite of C's prediction. So C's prediction of the state of the ball at $t = 1$ is incorrect. Therefore it is not possible for both B and C to be correct in their predictions.

As an aside, note that it is also not possible for both B and C to be *incorrect* in their predictions. So at least one of B and C is correctly predicting its own future ahead of time. **QED.**

As is the case with (1), in the derivation of (2) no explicit recourse is made to quantum mechanics, non-linear dynamics, or the like. Accordingly, the result could be restated to exclude systems involving non-linear dynamics, etc.

On the other hand, there are some important senses in which (1) and (2) differ. For example, (1) says that there is no computer such that for any desired ϑ_T concerning a system A, there is an input to the computer J_0 for which the computer correctly predicts J_T before time T. However (2) replaces J_0 with an observation apparatus; it says that there is no computer *and associated observation apparatus* such that for any desired ϑ_T concerning a system A, coupling the computer to A

The basic idea behind the proof is to have A contain a copy of our hypothetical predictor computer C, and then have the two computers each predicting something concerning the other. Things are set up so that both cannot make a correct prediction.

More precisely, the idea is replace the ball-and-sphere used in the proof of (1) with a copy of our hypothesized predictor computer. That second computer serves both roles the ball and sphere did; it is both the object whose future state is to be predicted by C, and it is also the object ensuring that C's prediction is wrong. It accomplishes the latter by predicting ahead of time what C's prediction will be, and then ensuring that that prediction is incorrect. If it can't do this, if our second computer cannot correctly predict C's prediction, then since it is an exact physical copy of our predictor computer, we have explicitly constructed a situation where the predictor computer cannot correctly predict the future. In which case the theorem is established. If however the second computer can correctly predict C's prediction, then C's prediction will necessarily be wrong, and again the theorem is proved.

Proof (by constructive contradiction): Let C be our proposed computer. Let B be another computer identical to C, except that the contents of its input and output might differ from those of C. (I.e., C and B share the same Hamiltonian, but at any given time they might occupy different points in their respective phase spaces.) If C can accurately predict the future for all systems isolated from C after $t = 0$ so long as C was coupled to the system before $t = 0$, then the same is true of B.

Have B and C physically isolated from each other after $t = 0$. Have C also isolated from anything else in the universe after $t = 0$.

On the other hand have B connected to an external ball and sphere (and implicit device connecting the ball and sphere to B), exactly as in the proof of (1). So the sphere and B's output are physically coupled at some time $\tau'' \equiv \tau(1) \in [0, 1)$. Note that B is not predicting the state of the ball in any sense as was the case in the proof of (1); the ball and sphere are simply a device designed so that whatever the output of B at time τ'' , the state of the ball at time $t = 1$ will be the

3. THE UNCOMPUTABILITY OF THE FUTURE FOR NON-COUPLED SYSTEMS

In the real world there is always some coupling between any two systems, as in (1). Nonetheless, a natural question is whether a result similar to (1) holds even if the computer and the system being predicted are physically isolated. To answer this, first note that if the computer is *never* coupled in any way to the system it is meant to predict, then what input it is fed is independent of the system being predicted, and therefore its “prediction” is independent of the system being predicted. So such a computer cannot predict the future state of any and all external systems.

As a result, we must rephrase our question. One way to do this is to require not that C and A are always isolated from each other, but that they are isolated for all times $t \geq 0$. In other words, the computer gets to “observe” A for some (perhaps semi-infinite) period ending before $t = 0$. The result of this observation determines J_0 , in the sense that that portion of the $t = 0$ state of C’s “input” degrees of freedom that is not a specification of T or ϑ_T is determined by this observation process. After $t = 0$, C and A are physically isolated.

For the purposes of this section, the term “computer” is implicitly redefined to include the associated observation apparatus. No restrictions are set on how the observation apparatus operates or how long before $t = 0$ it was coupled to A.

The proof of (1) no longer applies if we require that A is isolated from C for $t > 0$, since in that proof A includes C for such t . Nonetheless it is still impossible to faultlessly predict the future:

- (2) There cannot be a computer C such that, for any system A coupled to C’s observation apparatus for times $t < 0$ but isolated from C after $t = 0$, any $T > 0$, and any ϑ_T specifying desired information concerning the state of A at $t = T$, C takes T and ϑ_T as input at $t = 0$ and, at all times $t \in [\tau(T), T]$, has as output J_T .

linear function of the contents of the output section of the computer at time τ . I.e., if the output degree of freedom in question had a value .5 (rather than 0 or 1), the sphere would be positioned appropriately.

This specifies the physical scenario and J_1 . To complete the proof, we must now associate aspects of that scenario with A and J_0 . Let the total system A consist of the ball, the sphere, and the computer, coupled in the manner described. Then let J_0 be a set of information concerning the time $t = 0$ which suffices to fix J_1 . As an example, J_0 could be a physical description of the ball, C , the sphere, and the mechanism connecting C to the sphere. Since we're dealing with classical, finite systems, this set of information does indeed fix J_1 . (More generally, if there were no such "sufficient J_0 ", then (1) would be immediate.) However the computer can not correctly predict J_1 ; if it outputs a 1, then the correct output will be a 0, and vice-versa. Therefore the computer can not have the correct answer in its output at $t = \tau$.

As an aside, note that since J_0 is both encoded in C and describes C , J_0 describes itself. If however C is so computationally restricted that it cannot support such a J_0 , then trivially C cannot "always take a sufficient J_0 as input ...", so the theorem still applies. Similarly, if the size of the input region of C is restricted so that such a J_0 would be "too big to fit", then again the theorem applies. **QED.**

Note that in the proof of (1) no restrictions are set on how the computer operates, and there are no explicit assumptions about the computational power of either the computer or the universe. Indeed, even if the computer is powerful enough to solve the Halting Problem, (1) still holds. Note also that one can start the computer at any time before $t = 0$ (i.e., give the computer a potentially semi-infinite "running start") and again (1) holds. In addition, nothing in the proof of (1) assumes that J_0 actually concerns the state of A at $t = 0$ (although that's the most natural way of applying (1)); the result holds for any initial input to C , or even for none at all. Similarly, although we are primarily interested in computers that run programs which were specifically designed to try to compute the future, nothing in the proof of (1) requires this. Whatever the program, (1) shows that it must have output which never equals J_T .

that C can take J_0 , T and ϑ_T as input at $t = 0$ and, at all times $t \in [\tau(T), T]$, have as output J_T .

As a particular instance, this result holds when ϑ is a binary number concerning the state of the system.

Intuitively, the proof works by having A include C . We then arrange things so that at time τ the degrees of freedom that will determine J_T are coupled to C , in such a way to ensure that whatever C 's prediction at $t = \tau$, the actual value of J_T at $t = T$ is the opposite. This is formalized as follows.

Proof (by constructive contradiction): Take $T = 1$. Have A live in intergalactic vacuum, and consist in part of a perfectly elastic ball of small mass. We assume that in the immediate vicinity of the ball there are effectively no external forces like gravitational fields. The state of the ball at $t = 0$ is the ball travelling with velocity $(1, 0, 0)$ and centered on the x axis with its leading edge at position $(0, 0, 0)$. What we want to predict is whether or not at $t = T$ the leading edge of the ball has an x coordinate less than $\alpha \equiv \tau + [1 - \tau] / 2$, i.e., J_1 consists of a single bit saying whether or not the leading edge of the ball has an x coordinate less than α at $t = 1$. (More precisely, J_1 is the thresholding of one of C 's degrees of freedom which is interpreted as such a bit.) Note that $\alpha > \tau$.

External to the ball, we have a large perfectly elastic, massive sphere. The only characteristic of the sphere which we might modify is its position. The sphere is connected to the output of our proposed computer C : if C has the answer “ x coordinate $\geq \alpha$ at $t = 1$ ” at $t = \tau$ (i.e., if it has a 1 in a pre-determined output bit at that time), then at that time the sphere is positioned with its left-most edge at $(\tau + [\alpha - \tau] / 2, 0, 0)$, ahead of where the ball is at $t = \tau$, and before position α . If C has the answer “ x coordinate $< \alpha$ ” at $t = \tau$, then the sphere is positioned out of the ball's path. Once positioned, the sphere is not moved.

For completeness, we can assume that before time τ the sphere is in the “ x coordinate $\geq \alpha$ ” position. Also for completeness, assume that the position of the sphere at time $\tau + [\alpha - \tau] / 2$ is a

of freedom to be “falsifiable”. By this is meant that we can create an observable (by us) physical object such that that object reacts differently (and essentially instantaneously) to two different values of the output degrees of freedom iff they correspond to two different guesses for J_T . This restriction prevents us from hiding computing effort in the scheme for coding for J_T and thereby “lessening the load” on the computer; the encoding scheme must reduce, physically, to an almost instantaneously calculable one-to-one mapping.

We want to allow for both analog as well as digital computers. Accordingly, aside from the requirement that J_0 , T and ϑ_T can be encoded in the “input” section of the computer, that J_T can be encoded in the output section, and that that output be falsifiable, no other requirements are made of how J_0 , T , ϑ_T , and J_T are encoded, the physical extent of the computer (e.g., whether it’s finite), or anything of this sort. As a particular instance of what-is-not-assumed, no stipulation is made of whether the computer’s program is loaded in its input section; no stipulations are made concerning the computational universality of the computer.

2. THE UNCOMPUTABILITY OF THE FUTURE FOR COUPLED SYSTEMS

This section proves that one can not build a computer that can reliably and ahead of time predict the future state of a system for any system. It does this by considering situations in which the dynamics of the system is coupled to that of the computer. In proving the result no explicit recourse is made to quantum mechanics, non-linear dynamics, potentially infinite J_T , or the like. In other words, the result could be restated to explicitly exclude systems involving non-linear dynamics, potentially infinite J_T , etc. (Indeed, the proof of the result explicitly has J_T be a single bit.) The result also holds even if T is pre-fixed rather than arbitrary.

- (1) Fix T . There cannot be a computer C such that, for any system A , and any ϑ_T specifying desired information concerning the state of A at $t = T$, there is a J_0 sufficient for J_T such

which operates faster than that dynamics. It means that the computer takes as input some information concerning any physical system and its state at time $t = 0$, together with the specification of a time $T > 0$, and gives as output, *before time T*, desired information concerning the state of the system at time $t = T$. It is implicitly assumed that the input contains sufficient information to uniquely fix the desired information concerning the system's state at $t = T$ (whether or not the computer can calculate that information before $t = T$). Note that T need not be specified to infinite precision.

To formalize this, consider a finite, classical, non-chaotic, closed physical system A . Let ϑ_T be a suitably encoded specification of a desired finite set of information (i.e., a desired finite number of bits) concerning the physical state of A at time T . Let J_T be the actual desired information; ϑ_T specifies that we wish to know J_T . Let J_0 be another set of information. It provides some information about dynamical properties of A . It also provides information concerning the state of a set of physical variables at $t = 0$. We require of J_0 only that, together with the laws of physics, it is sufficient to uniquely fix J_T . (Such a J_0 will be referred to as “sufficient” for J_T .)

As an example, let A be closed in the time interval $[0, T]$, and let J'_T be the values, at time $t = T$, of the elements of a set of canonical variables which fully specify the state of A at $t = T$. J_T is the set of values J'_T , measured on a grid G of finite precision. ϑ_T is this definition of J_T . J_0 is the specification of the values of that canonical set at $t = 0$, together with a specification of the Hamiltonian describing A , all given on a (perhaps irregular) grid which is precise enough to uniquely fix J_T (i.e., to uniquely fix the state of A at $t = T$, as measured on G).

Our goal is to have the computer take as input J_0 , T , and ϑ_T , and then predict J_T correctly, at some time $\tau < T$ given by a pre-determined function $\tau(T)$. We want the same computer to do this for any system A , and for any associated T , J_0 and J_T such that J_0 fixes J_T uniquely. For simplicity, we assume that the computer halts at time τ , i.e., its output does not change between τ and T .

By “computer” is meant a physical dynamical system some of whose observable degrees of freedom are interpreted as (binary) “inputs” and some of whose observable degrees of freedom are interpreted as (binary) “outputs”. The input and output degrees of freedom can overlap and may even be identical. We want the encoding of the computer's guess for J_T in its output degrees

whose future state is not computable). In addition, they all assume one's computing device is no more powerful than a Turing machine. They also do not focus on scenarios where the computation is supposed to be a prediction of the future; in fact, they are not concerned with the temporal relation between one's information and what is to be predicted at all. There are then other additional caveats that apply to many of these previous results individually. For example, in [24] we are concerned with computing a real number rather than a "finite" quantity like an integer. Similarly, many of these previous results explicitly rely on chaotic dynamics.

None of these caveats apply to the results of this paper. In particular, no physically unrealizable systems, chaotic dynamics, or infinities are exploited. The results of this paper also hold even if one's computer were (somehow) more powerful than a Turing machine. Moreover, these results specifically concern the issue of having a computer correctly predict a finite amount of information concerning the future phase space position of a specific system.

Section 1 of this paper presents a formal definition of what it means for a computer to "reliably and ahead of time predict the future state of any system". Section 2 proves that one cannot build a computer that can, reliably and ahead of time, predict the future state of any system, if one allows that system to be coupled to the computer. Section 3 uses the result of section 2 to extend that result; it proves that even if the system and the computer are physically isolated from one another, still one can not build a computer that can, reliably and ahead of time, predict the future state of that system. The proof is essentially a physical version of a diagonalization proof; the result can be viewed as a physical analogue of Godel's Incompleteness Theorem.

A preliminary version of the work in this paper can be found in [21].

I. A DEFINITION OF WHAT IT MEANS TO "PREDICT THE FUTURE"

For the purposes of this paper, a physical computer will "predict the state of any system ahead of time" if the computer is an emulator of the physical dynamics of that system, an emulator

INTRODUCTION

Recently there has been heightened interest in the relationship between physics and computation ([1-22]). On the one hand, physics has been used to investigate the limits on computation imposed by operating computers in the real physical universe. Conversely, there has been speculation concerning the limits imposed on the physical universe (or at least imposed on our models of the physical universe) by the need for the universe to process information, as computers do.

To investigate this second issue one would like to know what fundamental distinctions, if any, there are between the physical universe and a physical computer. This paper addresses this issue by proving that one can not build a computer that can “process information faster than the universe”. One can not build a computer that can, for any physical system, take the specification of that system’s state as input, and then correctly predict its future state before that future state actually occurs. This result holds even if one restricts one’s attention to systems which are finite (i.e., contain a finite number of degrees of freedom), purely classical, and do not necessitate dynamics which is chaotic. (A similar result for chaotic systems is derived in [6].) In fact, for simplicity this paper will concentrate on precisely those kinds of systems. (Indeed, no explicit recourse is made to the known laws of physics - the results hold for a broad range of universes.)

There are a number of previous results in the literature related to the results of this paper. Many authors have shown how to construct Turing Machines out of physical systems (see for example [8, 23] and references therein). By the usual uncomputability results, there are properties of such systems that can not be predicted ahead of time. In addition, there have been a number of results explicitly showing how to construct physical systems whose future state is non-computable, without going through the intermediate step of establishing computational universality [24, 25].

There are several important respects in which the results of this paper extend these previous results. All of these previous results rely on infinities of some sort and rely on physically unrealizable systems (e.g., in [24] an infinite number of steps are needed to construct the physical system

AN INCOMPLETENESS THEOREM FOR CALCULATING THE FUTURE

SFI TR 96-03-008

by David H. Wolpert

The Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM, 87501 (dhw@santafe.edu).

PACS numbers: 02.10.By, 02.60.Cb, 03.65.Bz

Abstract: This paper proves that one can not build a computer which can, for any physical system, take the specification of that system's state as input and then correctly predict its future state before that future state actually occurs. Loosely speaking, this means that one can not build a physical computer which can be assured of "processing information faster than the universe". This result holds even if one restricts one's attention to predicting the states of systems which are finite, purely classical, and obey dynamics which is not chaotic, and even if one uses an infinitely fast, infinitely dense computer.