Itamar Pitowsky

The Physical Church Thesis and Physical Computational Complexity

1. Introduction: Three Models of Computation

My purpose in this paper is to examine, in a general way, the relations between abstract ideas about computation and the performance of actual physical computers. I shall deal with three abstract models of computers: the Platonist, the constructivist, and the finitist computers. These titles should be taken with a grain of salt, for they are only indirectly related to the philosophical schools which bear the same names.

The "Platonist computer" is really only a vague idea, perhaps not even a coherent one. It is a straightforward generalization of Zeno's paradox, which was proposed, e.g., by H. Weyl.[1] Suppose that we want to decide whether there exists a natural number n with the property P, that is to settle an existential question of the form $\exists nP(n)$. The Platonist computer is a machine which decides whether $P(1)$ is true in half a second, whether $P(2)$ is true in an additional one quarter of a second... whether $P(n)$ is true in 2^{-n} seconds and so on. If the machine encounters an example, it stops on the state YES before one full second and otherwise, it stops on NO after a full second.

It is questionable whether the very idea of "exhausting the set of natural numbers" is at all coherent, but I shall not address this issue here. My concern is with physical possibilities: Are Platonist computers physically possible? And if not, what is it about the physical universe

This research is partially supported by the Edelstein Center for the History and Philosophy of Science at the Hebrew University of Jerusalem.

[1] H. Weyl (1963), p. 42. For a detailed discussion of various versions of Zeno's paradox see A. Grunbaum (1967).

Iyyun, A Jerusalem Philosophical Quarterly 39 (January 1990)

which prevents their existence? The problem will be taken up in the next section.

The "constructivist computer" is just a (deterministic) Turing machine, and associated with this notion is the class of all Turing machine computable, i.e., recursive functions. Can a physical machine compute a non-recursive function? This question is addressed in section 3, where the notion of "physical computation" is made precise. Section 4 is devoted to the related "analogue–digital" distinction.

The "finitist computer" is a deterministic, polynomial time Turing machine (with a fixed finite number of tapes). Associated with this notion is the class of all functions which are computable in a number of steps bounded by a polynomial in the size of the input. As is well known many important computational problems seem to lie outside the class P, typically the so-called *NP*-hard problems.[2] The question to be addressed in sections 5–6 is whether we can invoke physical laws to reduce a computational problem that is manifestly or presumably of exponential complexity, and actually complete it in polynomial time. As stated, this question is not quite well formed, and one of my tasks here is to make it more precise.

Finally, in section 7, I shall examine the role which quantum mechanics may play in this context. In particular I will demonstrate that quantum theory allows, in principle, for unbounded parallel computations, in the sense that information of exponential size can be squeezed into polynomially many distinct quantum states. We shall see that the trouble lies with the retrieval of this information in polynomial time.

2. Zeno's Paradox in a Black Hole?

Conventional wisdom has it that Platonist computers are physically impossible, "owing mainly to the existence of a limit to the velocity with which physical operations can be performed."[3] Yet the same theory which maintains that the upper limit on the speed of information transmission is the velocity of light, also maintains that time is relative to the observer. In fact, time dilation is a logical consequence of the constancy of the velocity of light.

[2] On the theory of *NP*-completeness see M. Garey and D.S. Johnson (1979).

[3] P. Benacerraf and H. Putnam (1983), introduction, p. 20.

Let us ignore for the moment any physical restrictions on the size of computation *space*, and consider the temporal aspect only. Suppose that M, a mathematician, is literally dying to know whether Fermat's conjecture is true or false. He takes a trip in a satellite which revolves around the earth. The satellite has an immense engine, which boosts it so hard, that its instantaneous tangential velocity is $V(t) = c[1 - e^{-2t}]^{1/2}$ where t is the earth's time scale and c the velocity of light. The engines are also oriented in such a way as to keep the satellite in a fixed orbit. If T is the satellite's local time scale, then the time interval dT is given by $dT = e^{-t}dt$. Hence one second in the satellite's time scale, corresponds to eternity on earth, since $\int_0^\infty e^{-t}dt = 1$.

While M peacefully cruises in orbit, his graduate students examine Fermat's conjecture one case after another, that is, they take quadruples of natural numbers (x,y,z,n), with $n \geqslant 3$, and check on a conventional computer whether $x^n + y^n = z^n$. (Remember, we assume no limit on computation space.) When they grow old, or become professors, they transmit the holy task to their own disciples, and so on. If a counterexample to Fermat's conjecture is ever encountered, a message is sent to the satellite. In this case M has a fraction of a second to hit the brakes and return home. If no message arrives, M disintegrates with a smile, knowing that Fermat was right after all.

But this story is too simple to be true.[4] Suppose that Fermat's conjecture is false and a message to that effect is sent to M by the students. The crucial question is whether the message is going to arrive at its destiny, and the answer depends on the precise formulation of the problem and on the details of space-time geometry.[5] Suffice it to say that

[4] This and the following remarks on space-time structure, and its relations to our problems, were pointed out to me by David Malament. I am grateful to him for making the present version of "Zeno's paradox" much more interesting (and accurate) than I have originally conceived.

[5] There is a straightforward black hole analogue of the satellite story. In this version of our tale, the mathematician jumps into a black hole while his students perform the calculations beyond the horizon. The proportion of proper time scales is just right. A finite time for M is infinite for his students. Yet beyond a certain point no message can be sent to M from outside. (Black holes bear that name since no light can escape outside. Notice however that we are discussing the reverse process, sending messages from outside into the hole, which also turns out to be impossible.)

for some solutions of the Einstein equations, such a feat is possible. (The character of the solution depends, as usual, on initial and boundary conditions.) In other words, *as far as computation time is concerned, the existence of Platonist computers is compatible with general relativity* (though it is probably incompatible with the conditions in the actual universe).

The real reason why Platonist computers are physically impossible *even in theory* has to do with computation space. According to general relativity the material universe is finite. Even if we use the state of every single elementary particle in the universe, to code a digit of a natural number, we shall very soon run out of hardware.

3. Physical Computers and Recursion

The concept of "physical computer" which will be employed here is the most general, and thus the most trivial one. A (deterministic) physical computer is a triple (S,W,T), where S is a concrete physical system, W an observable or a set of observables associated with S (for example, the energy of S at time t), and T is a dynamical theory which pertains to the system S, and to the observable(s) W in particular.

In order to use the computer, we set the system S into an initial state, where the value of W is W_0. Then we let the system run its course, and at future time, we measure the value $W(W_0,t)$. In that case we say that the physical computer has computed the output $W(W_0,t)$, from the input W_0.

How do we know that what has been computed is really $W(W_0,t)$ and not some other function, say $W^*(W_0,t)$? Well, this is where the theory T enters the picture, for it tells us that the dynamic change of the observable is according to the functional relation $W(W_0,t)$ and not $W^*(W_0,t)$. If the theory T is well established, and highly verified by many observations on many systems, then there is a good reason to believe the claim. In any case we can sometimes check our "physical computation" by performing the calculation of $W(W_0,t)$ by other means, such as simulation. If no error is found, then we have yet another verification of T.

According to this rather simple picture, the planets in their orbits may be conceived as "performing computations," provided that we make suitable measurements and rely on a good theory. The concept is sufficiently general to include all actual computers. When you punch '2+2=' on the keyboard of your personal computer, the mechanical

signals are turned into electric signals, which set the chips into some initial state, whereupon a deterministic process begins, and very soon ends. The state of the chips is then automatically measured by an internal device, and the physical state is translated into the digit 4 on the screen. Even when very complicated calculations are performed on a digital computer, we still tend to believe the result. Why? The reason is twofold. Firstly, we believe that the program establishes what it is supposed to establish; this can be verified logically and also by examining simple instances. Secondly, and more importantly for our purpose, we believe the result because we assume that the hardware functions as it should. But the hardware is nothing but a physical system S and some observables associated with it, which are capable of being efficiently measured. We believe that its dynamics is given by the theory of solid state and electric wiring. Every successful run of a computer is another verification of this theory.

I have confined myself, so far, to deterministic systems only. This is not a limitation of principle; we can equally well consider systems such that, given an initial state W_0, end up at time t in one of various possible states, while the theory predicts a certain probability distribution over the outcomes. For reasons of simplicity I shall by and large ignore such systems.

Let S be a system, $W(t)$ its energy at times $t>0$. Let $f(n)$ be the integral value of the energy at integral times (in seconds): $f(1)=[W(1)]$, $f(2)=[W(2)],\ldots,f(n)=[W(n)],\ldots$. Does there exist a system S, and a highly verified theory T, which predicts that $f(n)$ is not a recursive function? If the answer is positive, then we have a method for calculating the values of a non-recursive function. In order to establish the value of $f(n)$, we simply measure the energy of the system after n seconds from its initial state and take the integral value of the result. Again one can ask the question: How do you know that the function which is being computed is really f? After all, one can only establish finitely many values of the energy by real measurements, and any finite sequence of natural numbers can be interpolated by a recursive function. The answer is that the theory T says that the values computed are those of f. Since T is well corroborated by observations on many systems, there is no reason to doubt the conclusion, at least no more than to doubt any complex computation by a digital computer.

Wolfram has recently proposed a thesis — "a physical form of the Church–Turing thesis" — which maintains, among other things, that no non-recursive function is physically computable.[6] (Wolfram's thesis has a much wider scope and I shall discuss it in detail in subsequent sections.)

It should be noted that Wolfram's contention has nothing to do with the original Church thesis. By "every computable function is recursive," Church meant that the best analysis of our pre-analytic notion of "computation" is provided by the precise notion of recursiveness. Indeed one sometimes refers to Church's thesis as "empirical,"[7] but the meaning of that statement, too, has nothing to do with physics. By "empirical" one means here that all the definitions of "computability," which were proposed for different purposes and in different contexts (e.g., by Gödel, Turing, Church, Post), turned out to be equivalent. This fact is taken as an "empirical evidence" for Church's thesis.

The question of whether Wolfram's thesis is valid is a problem in the physical sciences, and the answer is still unknown. Yet there are very strong indications that Wolfram's thesis may be invalid. Pour-El and Richards (1981) demonstrated that a physical system S, and an observable W may exist, such that $W(0)$ is a recursive real number[8] while $W(1)$ is not a recursive real. Recursive initial conditions (at recursive space-time points) are not necessarily transformed to recursive values at later (recursive) times. The system at hand is a three dimensional wave, $\psi(x,t)$, $x \varepsilon \mathcal{R}^{(3)}$, which satisfies:

$$\nabla^2\psi - \frac{\partial^2\psi}{\partial t^2} = 0, \qquad \psi(x,0)=f(x), \qquad \left(\frac{\partial\psi}{\partial t}\right)_{t=0} = 0$$

They proved that for some choices of recursive real initial condition f, the solution ψ at position $x=0$ and time $t=1$, is not a recursive real number.

This example does not quite refute Wolfram's thesis, for two reasons. Firstly, the function f in the initial condition, though a recursive real

[6] S. Wolfram (1985). A somewhat different approach to the physical Church thesis is in D. Deutsch (1985).

[7] For example in Y. Manin (1977), p. 181.

[8] The notions "recursive real number" and "recursive real function" are straightforward generalizations of the natural numbers case. See M.J. Beeson (1985).

function, is an extremely complex function. One can hardly expect such an initial condition to arise "naturally" in any real physical situation. Secondly, we deal with recursive *real* functions, and in physics we never get beyond a few decimal digits of accuracy anyway. But I believe that such theorems could be modified to yield genuine counterexamples. What has been demonstrated by Pour-El and Richards is that "recursive" is not a natural physical property. Physical processes do not necessarily preserve it.

4. On the Analogue–Digital Distinction

This seems to be the appropriate point to discuss the issue of the analogue–digital distinction, which has been a source of controversy, in particular among cognitive psychologists.[9] The distinction between "digital" and "analogue" computation has never been fully clarified. By "digital" people usually mean a computation performed in several distinct discrete steps by the manipulation of symbols. Analogue computaton is taken to be a smooth physical process, which transforms the initial value of some physical observable to a final output value, in a manner similar to the one I have described above.

The distinction seems to me to hang on the language which is being used in answering the question: "How did the machine perform the computation?" In the digital context, the answer is given in terms of a programming language, by reference to the logical steps taken during the process of computation. In the analogue case, reference is made to the physical laws that govern the dynamics of the physical machine. Since the answers are given in terms of two distinct categories of description, it very often seems that disputants in cognitive science talk past each other.

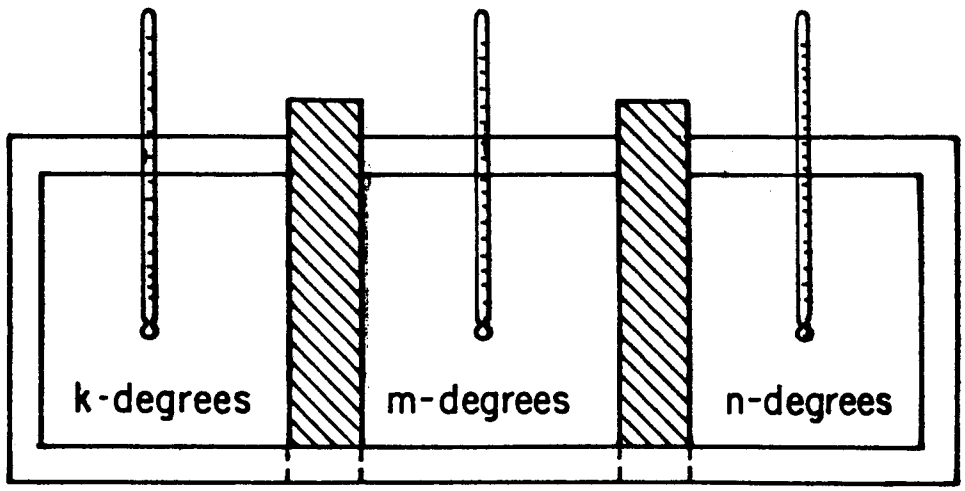
The key notion which underlies the distinction is that of recursion. Being "recursive" is an abstract property of a functional relation. The crucial aspect, which underlies the analogue–digital distinction, seems to me to be whether this abstract property is in any way reflected in the *physical* process of computation, or rather is completely extraneous to it. Take for example the motion of the planet Earth. Its position relative to

[9] For example Z.W. Pylyshyn (1986) and the opposing view in D.E. Rumelhart and J.L. McClelland (1987).

the sun at time t is a recursive real function of its position at time $t=0$. Yet there is nothing in our description of this motion that in anyway refers to this fact. Recursiveness in this case is a totally foreign and completely irrelevant feature of the observable.

Or consider another example. In order to calculate the average of three numbers k,m,n, we take an insulated container divided into three equal parts by insulated removable barriers. In each one of the three sections we put a thermometer (fig. 1).

Figure 1



To introduce the "input" we heat (or cool) the left section to k degrees, the middle section to m degrees and the right section to n degrees. Then we remove the two barriers simultaneously and wait until temperatures equalize (or approximately equalize, as best as we can tell). The temperature value present on each one of the three thermometers is the "output" $\frac{1}{3}(k+m+n)$. Although this is a recursive function, this fact has no bearing on the description of the 'computation' process, whether we give it in terms of classical thermodynamics or in terms of statistical mechanics.

Now repeat the calculation, this time, however, remove the barrier between left and middle section first and let temperatures equalize (to $\frac{1}{2}(k+m)$). Only then remove the other barrier and wait again until temperatures equalize. This two-stage process is physically distinct from the first one-stage process. If we wish to explain the result of the second

process we should, in addition to the second law of thermodynamics, invoke the fact that $\frac{1}{3}(k+m+n)=\frac{2}{3}[\frac{1}{2}(k+m)]+\frac{1}{3}n$. This (recursive) relation has nothing physical about it. Yet we must refer to it if we are to explain the *physical* process in the second computation.

Thus, instead of the analogue–digital distinction, I propose to introduce the more accurate analogue–recursive distinction. In the physical description of a purely analogue process, we require no reference to the fact that its input–output relation is recursive (even if it happens to be recursive).

A physical computation process, whose physical description does require reference to some such features of the input–output relations, we call recursive. To make things simple, a recursive computation is a recursive chain of analogue computations (a chain which may have feedback loops).

The most extreme case is that of an actual digital computer, all of whose analogue components (microchips) are essentially identical and very simple. In this particular case we can suppress reference to the mode of operation of the chips altogether, and concentrate on the recursive chain of information processing (as described in the machine language). This, however, is nothing but an extreme case of a peculiarly artificial design. The brain presumably works differently, and its basic analogue unit may not be a single neuron, but rather a group (of about 10^4) neurons.[10] Each such group may have a distinct mode of analogue operation, which depends on the specific inter-connections among its neurons. If this is the case, it is quite clear that we shall not be able to suppress reference to the physics of the hardware, if we attempt to understand cognition. I shall return to this point in section 6.

5. Finitist Computers: Reducing Complexity by Analogues

Returning to our main concern we can ask the following general question: Is there a mass decision problem, which belongs to a given time complexity class (say, exponential time complexity) but which can nevertheless be decided in a relatively short time (polynomial time), by

[10] See J. Hopfield (1982), and also sections of D.E. Rumelhart and J.L. McClelland (1987) for some specific models.

appropriate analogues? This question can be asked separately for each complexity class, but it has the most significant implications when asked relative to the classes *NP* and *P*. Even if $NP \neq P$ we may still be able to use analogues and cut the computation time for, say, the TRAVELLING SALESPERSON, into polynomial time. This can be achieved, perhaps, with the use of some physical process, which actually terminates rapidly, but whose simulation on a digital computer nevertheless requires exponential time. We have seen that the property "recursive," of physical functional relations, may not be reflected in the actual physical process. By analogy, there is no a priori reason to believe that abstract complexity constraints could not be bypassed by physical principles.

Wolfram (1985) maintains that shortcutting computational complexity in an essential way, with the aid of physical law, is impossible: "One can expect in fact that universal computers are as powerful in their computational capabilities as any physically realizable system can be, so that they can simulate any physical system.... No physically implementable procedure could then short cut a computationally irreducible process."

As stated, Wolfram's formulation is not quite well posed. The theory of computational complexity defines "computation time" as the number of steps taken by a Turing machine. The duration of physical processes is measured by real time units. Also, abstract complexity classes are defined in terms of the asymptotic halting time of unbounded Turing machines, and are thus defined only up to constant factors. The material universe, we have noted, is finite. Given this perspective we can say that any real physical process has linear complexity. All we have to do is to choose a sufficiently large constant.

There are two ways to reformulate Wolfram's thesis and make it meaningful and, I think, important. There is the theoretical way of the "thermodynamic limit formulation" and the more practical way of the "initial segment formulation." I shall take them in turn.

(a) Thermodynamic Limit. Theoretically we can conceive of families of physical systems $\{S_\alpha\}$, where α ranges over some infinite index set, such that the dynamics of all these systems is governed by the same theoretical law. (Example of such systems are "spin glasses," "cellular automata," and so on.) Let W_α be an observable associated with S_α, to make things simple assume W_α is the energy of S_α. Suppose $|S_\alpha|$ is the complexity of the

system S_α, that is, the number of code bits it takes to describe the system. Thus $|S_\alpha|$ depends on the number of units (particles) from which S_α is composed and the type of interactions among the units. Now suppose we can find such a family $\{S_\alpha\}$, for which the following two conditions hold:

(i) Calculating the minimum energy of S_α: $minW_\alpha$, is an *NP*-hard problem, that is, any known method for the calculation of $minW_\alpha$ will require a number of steps which is exponential in $|S_\alpha|$ as $|S_\alpha| \to \infty$.

(ii) The dynamic theory, which governs the systems $\{S_\alpha\}$, predicts that the system S_α reaches its minimal energy state in a number of seconds which is bounded by a polynomial in $|S_\alpha|$ as $|S_\alpha| \to \infty$.

Wolfram's thesis, according to the thermodynamic limit formulation, states that conditions (i), (ii) are contradictory (if $NP \neq P$). There are some systems for which condition (i) is known to be true,[11] but their expected real relaxation time is not known (except for estimations which are based on simulations and thus beg the question). I believe the best hope to refute Wolfram's thesis, in this formulation, is to consider quantum systems. More on this in section 7.

(b) Initial Segment. In this formulation we do not ask about asymptotic behaviour of hypothetical systems in the thermodynamic limit, rather we concentrate on a practical aspect: *Does there exist a large initial segment of an NP-complete language, many of whose instances require years to decide by an implementation on a conventional computer, but all of whose instances can nevertheless be decided in a few seconds by an appropriate analogue?*

If we want to make this question even more practical, we can relax the condition that *all* the cases in the initial segment should be decided rapidly by the analogue, and impose the condition only on, say, 99% of the cases. The success or failure of such an analogue is really a matter of practical judgment of relative time scales, but I think that my intention is clear enough. Note that such an analogue may very well exist, even if Wolfram is right with respect to the thermodynamic limit formulation.

It is not known whether constructing such an analogue is physically possible. General evidence with respect to relaxation time vs. simulation

[11] For spin glasses this is proved in F. Barahona (1982); see also I. Pitowsky (1989) for connections with other combinatorial problems in physics. For polymers, property (i) is proved in D. Sankoff and J. Kruskal (1983).

time of various natural systems suggests however that it may exist. I shall discuss two examples.

(i) Biochemists distinguish between the primary and secondary structure of a protein. The primary structure is the linearly ordered sequence (or sequences) of amino-acids composing the protein. The secondary structure is the three-dimensional configuration that the amino-acids of the protein assume in space. Determination of the primary structure is a relatively easy task, which requires simple techniques of chemical analysis. Determining the secondary structure is a much more complex task; it requires X-ray crystallography and many computation hours to decipher the diffraction patterns. Therefore, it would be nice if we could write a computer program that took the primary structure as input and produced the secondary structure as output.[12] The physics involved in such a computation is well known, all we have to do is "just" to find those configurations in three space which minimize the energy. Yet the computation is immensely complex: amino-acids, which are far removed on the one dimensional chain, may interact strongly when the protein molecule folds in three dimensions; also the interaction among amino-acids is quite complex. In short, even if such a program is possible, it will probably run for a very long time.

Consider now the natural process of protein synthesis. The protein is synthesized on the RNA molecule, which serves as a template, then an enzymatic process begins which detaches the new molecule from the RNA. At this stage, the protein molecule appears as a long, more or less linear chain of amino-acids. Then, in no time, it folds and assumes (one of) its stable three-dimensional state(s). The difference between this real relaxation time and the expected simulation time is striking, maybe seven or eight orders of magnitude. One may argue that simulation time will become much shorter when massive parallelism is introduced. To establish this claim, however, one has to demonstrate how parallelism can shortcut the simulation time of an n-body interaction, in which every body is interacting, in a complicated manner, with many of the others.

(ii) Consider the following computation problem. We are given a list of

[12] There are several groups who actually work on such an algorithm, but I do not know any details. With respect to general polymers, finding the minimal energy conformations is *NP*-hard; see note 11.

n, not necessarily distinct, natural numbers between 1 and 1000, which appear in a random order. Our task is to group the numbers into ten groups. The first group consists of all numbers in the list between 1 and 99, the second between 100 and 199, and so forth.

Every program which performs the task should screen the input. If we have a single tape Turing machine the screening process alone takes more than n steps. If we have ten tapes, running in parallel, the task can be performed ten times faster. In short, any program for this task requires at least $O(n)$ steps even when (bounded) parallelism is allowed.

Consider now the following practical problem. We are given a container filled with thousands of particles, all made of the same material, but which have different sizes, ranging between $1mm^3$ to $1cm^3$. Our task is to sort the particles into ten groups. The first group consists of all particles with volume between $1mm^3$ to $99mm^3$, etc. A fool will pour the contents of the container on a table, take each particle, measure its volume, and put it in the right group. Seventeen fools can accomplish the task seventeen times faster, using the same tedious method. Obviously there is a better method: shake the container. After a few good shakes the particles arrange themselves according to size, bigger on top smaller on bottom. The method is not full proof and some particles may end up in the wrong group, but the error is quite marginal.

What should we call "a step" in the shaking method? If one shake is one step, then it seems that the number of steps is of the order of magnitude of $log(n)$, perhaps $5log(n)$, and not the order of magnitude of n. This seems to be the case at least when the number of particles, n, ranges over reasonable limits. A simulation of the shaking process has been performed lately.[13] To get a reasonable handle on this very complex process, the program performed 3000 Monte Carlo steps per particle per shake.

In short, the relation between simulation time and the actual duration of a physical process is not clear. Even if Wolfram is right with respect to the "thermodynamic limit" formulation of his thesis, the constant factors involved can be so large that the more practical "initial segment" formulation of the question may have a positive answer.

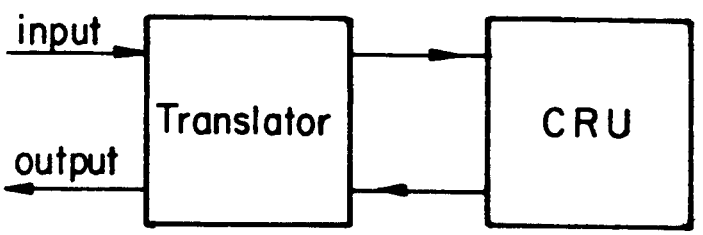
[13] See A. Rosato, K.J. Strandburg, F. Prinz and K.H. Swendsen (1987).

6. Complexity Reducing Units

Suppose that indeed there exists an analogue, which calculates very rapidly a large initial segment of an *NP*-hard problem. Call such a device a complexity reducing unit, CRU. To make the discussion more concrete suppose the CRU is composed of 10^5 units all connected to each other. Each instance of the initial segment of the *NP*-hard problem corresponds to the ground state energy of a particular pattern of interaction among the units. Each such pattern can be programmed in the CRU by automatic switching of the preexisting connections. We assume that in each instance the CRU reaches its ground state very fast, and then the energy can be measured directly by an internal device. We also assume that simulating the dynamics of the CRU on a digital computer is considerably slower; the calculation of the minimal energy may require years of computation time for some instances.

NP-complete problems are universal in their class. This means that a polynomial time translation exists from every *NP*-problem to a given *NP*-complete problem. Often the polynomial in question has a low degree, or is even linear, and the constants involved are not too big either. This means that instances in large initial segments of many *NP*-problems could be efficiently translated into their corresponding CRU minimal energy problem. So consider a combination of a regular digital computer, which we call a TRANSLATOR, and a CRU (fig. 2).

Figure 2



Suppose that we want to solve an instance of the TRAVELLING SALESPERSON (TSP). We feed the input into the TRANSLATOR, whereupon it is translated into the corresponding instance of the CRU minimal energy problem. This, in turn, is fed into the CRU and causes it

to enter into its proper state by automatic switching. The CRU reaches its ground state fast and then its energy is measured. The result is fed back into the TRANSLATOR which retranslates the answer into the language of TSP, and prints the result as output. The entire process is fast, by hypothesis, much faster than it would take to solve the instance of the TSP directly on a digital computer.

Which language should we use when we describe the combined action of the TRANSLATOR and the CRU? Equivalently, what makes the combined system "smart"? Surely the TRANSLATOR is essential, the CRU itself is just a stupid physical device. The translation process can be described only in logical, not physical terms; it has to take into account the formal isomorphism between TSP and the CRU minimal energy problem. By contrast, the action of the CRU can only be described in physical terms. Its ground state is attained fast because of some law of nature. In short, one cannot explain the combined action of the system in programming language alone, or in physical language alone. The recursive character of the problem is as essential as is the architecture of the hardware.

But aren't we constructed in a similar way? I do not mean by this that we have something corresponding to the above *NP*-complexity reducing unit, but rather that the physical action of the neural network serves, in general, to reduce the complexity of some typical human problems, mainly those which formally involve massive searches.

There are two reasons to believe this. Firstly, from general evolutionary considerations, it seems quite plausible that the cognitive apparatus of animals would take advantage of physical laws. A wiring system that relaxes fast to a stable state has an advantage over slower patterns. Secondly, and apart from such considerations, we can observe human behaviour in some cases where massive searches are involved. Present computer programs for chess, even if they include considerable heuristics, still undergo huge searches. As is well known, the achievements of computer chess is due to the sheer speed of the microchips no less than to the ingenuity of the programs. Humans, by contrast, do not use massive searches, in fact most of the logical possibilities never occur to them. At most they examine a handful of possible actions. What is the source of this difference?

As usual there are two possible answers. Either computer scientists

simply failed, as yet, to write a really good program, or else humans take advantage of an entirely different architecture (or perhaps both). I believe that physical aspects cannot be ignored, at least not a priori. It may be the case that the input (the setup on the chess board) is somehow translated in our brains into an optimization problem of a certain observable associated with a network of neurons. The network relaxes relatively fast to an optimal value, and the result is just the (code of the) next move. Maybe this is the real reason why most possibilities do not even occur to us. This story may be completely wrong, but I cannot conceive of any reasonable argument for its a priori dismissal.

7. Quantum Mechanics and Unbounded Parallelism

The principles of quantum mechanics may add to the capacity of computers. This has been theoretically proved by Albert (1983) and Deutsch (1985). As far as computational complexity is concerned the issue remains open. The remark below provides some evidence concerning the theoretical powers of "quantum computation" in the realm of reducing complexity. As usual, it is the principle of superposition which adds strength, over and above classical principles.

I shall introduce first a particular *NP*-complete problem, which will serve as our example. Let k be a natural number and consider a set of m triples of natural numbers:

(1) $A = \{\{a_1^1, a_2^1, a_3^1\},\{a_1^2,a_2^2,a_3^2\},\ldots,\{a_1^m,a_2^m,a_3^m\}\}$

Where $1 \leqslant a_1^i < a_2^i < a_3^i \leqslant k$, for $1 \leqslant i \leqslant m$. Call such a set *a proposition*. A truth function for A is any function $t:\{1,2,\ldots,k\} \rightarrow \{0,1\}$; a truth function is called a solution for A if $t(a_1^i)+t(a_2^i)+t(a_3^i) = 1$ for all $1 \leqslant i \leqslant m$. The problem to be discussed below is called ONE IN THREE 3-SATISFIABILITY, its instance is a proposition A, as in formula (1), and the question to be decided is whether there exists a solution for A. This problem is *NP*-complete.[14] In order to translate this problem into the language of quantum theory let a proposition A be fixed and consider a spin-1 massive particle. The spin space of such a particle H has three dimensions corresponding to the orthonormal states $|+\rangle$-spin up, $|0\rangle$-spin zero,

[14] See T.J. Shaeffer (1978), M. Garey and D.S. Johnson (1979).

$|-\!\!>$-spin down, in some fixed direction in physical space. If we take m identical yet distinguishable massive spin-one particles, then a basis in spin space H_m of the m-particle system is the set of all 3^m vectors of the form

$$(2)\quad |\psi\rangle = |\psi_1\rangle|\psi_2\rangle\ldots|\psi_m\rangle$$

where each $|\psi_i\rangle$ is either $|+\rangle$ or $|0\rangle$ or $|-\!\!>$. In quantum theory a measurement performed on a quantum system is represented by a self-adjoint operator. Let $|\psi\rangle\langle\psi|$ denote, as usual, the projection on the state $|\psi\rangle$ and for each $1 \leqslant a \leqslant k$ let $V_a = D(a,1) \otimes D(a,2) \ldots \otimes D(a,m)$ denote the operator on H_m given by

$$(3)\quad D(a,i) = \begin{cases} |+\rangle\langle+| & \text{if} \quad a = a_1^i \\ |0\rangle\langle 0| & \text{if} \quad a = a_2^i \\ |-\!\!>\langle-\!\!| & \text{if} \quad a = a_3^i \\ I(\text{unit}) & \text{if} \quad a \notin \{a_1^i, a_2^i, a_3^i\} \end{cases}$$

Now put

$$(4)\quad V = \prod_{i=1}^{m} (V_{a_1^i} + V_{a_2^i} + V_{a_3^i})$$

Then one can quite easily prove that all the states $|\psi\rangle$, of the form (2), are eigenstates of V with eigenvalues zero or one, and that A has a solution if and only if $V|\psi\rangle = |\psi\rangle$ for some such state.

So far we have gained nothing; we have translated a computation problem into the language of measurements on particles. Rather than checking all 2^k truth functions $t: \{1,2,\ldots,k\} \to \{0,1\}$, to see whether one is a solution for A, we can measure V on all 3^m states $|\psi\rangle = |\psi_1\rangle|\psi_2\rangle\ldots|\psi_m\rangle$, to see whether one of them is an eigenstate of V with eigenvalue 1. Still there are 3^m such measurement steps.

But quantum mechanics allows for superposition of states. In particular if all m particles are identical, *and indistinguishable*, a case which presumably occurs when they are packed densely together, then, by Pauli exclusion principle, the *only* permissible states are the symmetric ones. If k_1, k_2, k_3 are three natural numbers $k_1, k_2, k_3 \geqslant 0$, $k_1 + k_2 + k_3 = m$, then the symmetric state $|k_1, k_2, k_3\rangle$ is given by

$$(5)\quad |k_1,k_2,k_3\rangle = \sqrt{\frac{k_1!k_2!k_3!}{m!}} \sum |\psi\rangle$$

Where the sum is taken over all states $|\psi>$ of the form (2), such that $|+>$ appears k_1 times in the tensor product, $|0>$ appears k_2 times, and $|-->$ appears k_3 times. There are $\frac{1}{2}(m+1)(m+2)$ symmetric states. In other words, all 3^m potential "solutions" $|\psi>$ of the form (2) were coded into polynomially many states. All that is left is to retrieve the relevant information by appropriate measurements, this time only $O(m^2)$ many measurements, but which?

The natural candidate is the operator $W = SVS$, where S is the symmetrizer, the (normalized) sum of all $m!$ particle permutation operators. Indeed, it is easy to see that all symmetric states are eigenstates of W, and that A has a solution if and only if there exists a symmetric state such that $W|k_1,k_2,k_3> \neq 0$. Does this mean that quantum theory allows for the solution of an *NP*-complete problem in polynomially many measurement steps? Well, the answer depends on the interpretation of complex measurements. Theoretically there should exist an "apparatus" associated with the self-adjoint operator W, and which somehow, miraculously, measures the value of the observable to which W corresponds. Practically the construction of such an apparatus may require exponential time. It may be the case that in order to make it, one has to solve first the same instance of the *NP*-complete problem which we were set to decide originally.

In any case it is clear that the principle of superposition allows to squeeze information of exponential complexity into polynomially many quantum states. This can even be done in practice. The efficient retrieval of this information poses the real problem.

The Hebrew University of Jerusalem

References

Albert, D.Z. 1983. On Quantum Mechanical Automata. *Phy. Lett. A.* 98:249.

Barahona, F. 1982. On the Computational Complexity of Ising Spin-Glass Models. *J. Phys. A* 15:3241.

Beeson, M.J. 1985. *Foundations of Constructive Mathematics.* Berlin: Springer.

Benacerraf, P. and H. Putnam. 1983. *Philosophy of Mathematics — Selected Readings.* 2nd ed. Cambridge University Press.

Deutsch, D. 1985. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proc. R. Soc. Lond. A* 400:97.

Garey, M. and D.S. Johnson. 1979. *Computers and Intractability — A Guide to the Theory of NP-completeness.* New York: W.H. Freeman.

Grunbaum, A. 1967. *Modern Science and Zeno's Paradox.* Middletown, Conn.

Hopfield, J.J. 1982. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci. USA* 79: 2254.

Manin, Y. 1977. *A Course in Mathematical Logic.* New York: Springer.

Pitowsky, I. 1989. Correlation Polytopes: Their Geometry and Complexity. *Math. Programming* (Forthcoming).

Pour-El, M. and I. Richards. 1981. The Wave Equation with Computable Initial Data such that its Unique Solution is not Computable. *Adv. in Math.* 39:215.

Pylyshyn, Z.W. 1986. *Computation and Cognition.* Cambridge, Mass.: MIT Press.

Rosato, A., K.J. Strandburg, F. Prinz, and K.H. Swendsen. 1987. Why the Brazil Nuts are on Top: Size Segregation of Particulate Matter by Shaking. *Phys. Rev. Lett.* 58:1038.

Rumelhart, D.E. and J.L. McClelland. 1987. *Parallel Distributed Processing.* Cambridge, Mass.: MIT Press.

Shaeffer, T.J. 1978. The Complexity of Satisfiability Problems. *Proc. 10th Ann. ACM Symp. on Theory of Computing*, 216. New York: Association for Computing Machinery.

Sankoff, D. and J. Kruskal. 1983. *Time Wraps, String Edits and Macromolecules.* Reading, Mass.: Addison-Wesley.

Weyl, H. 1963. *Philosophy of Mathematics and Natural Science.* New York: Atheneum. (Original German edition 1927).

Wolfram, S. 1985. Undecidability and Intractability in Theoretical Physics. *Phys. Rev. Lett.* 54:735.