A PHYSICIST'S MODEL OF COMPUTATION

EDWARD FREDKIN
DEPARTMENT OF PHYSICS
BOSTON UNIVERSITY
BOSTON, MA, 02215, USA

This paper is an attempt to make a statement about what a computer is and how it works from the perspective of physics. The single observation that computation can be a reversible process allows for the same kind of insight into computing as was obtained by Carnot's discovery that heat engines could be modeled as reversible processes. It allows us to bring computation into the realm of physics, where the power of physics allows us to ask and answer questions that seemed intractable from the viewpoint of computer science. Strangely enough, this effort makes it clear why computers get cheaper every year.

Physics Applied to Computers

Let us imagine ourselves physicists thrust back into the days before Galileo. We observe a heavy object that we hold in our hand. We note that it takes a steady force to hold the object and keep it from falling. We let it go, it falls to the ground and then comes to a stop with a thud. Back then, such observations could lead one to a set of laws of physics that are quite different from what we know today. For example, we might have concluded that a steady force is needed to keep an object still. We might have concluded that the energy or momentum (vis viva) associated with a massive object in motion simply disappears through friction or inelastic collisions.

There are certain principles that help to separate rough, common sense conclusions from the underlying laws of physics:

1. There are laws of essentially unlimited and clear applicability. A good example is conservation of momentum. The object is to discover and make use of such laws.

2. It is wise to formulate our laws with regard to idealized models that don't suffer from the confusions of the real world.

3. We need to understand and make use of the concept of a closed system, where we can account completely for what is going on.

4. It is often easier to ignore boundary conditions; ie to analyze a dynamic interval as opposed to including the special circumstances about how it starts and finishes.

5.

Today, everything mentioned above seems too obvious for words. Nevertheless, the kind of thinking that is second nature to physicists seems not to have been applied to computation. After a brief foray into how present day computer scientists think about computation, we will explain computation from a physicist's perspective.

There are many different ways to model computation. The favorite of the computer scientists is the Turing machine, which is a model that simplifies and makes clearer what a non-thinking clerk can accomplish (or cannot accomplish) with an unlimited supply of time, pencil and paper. What Turing did was to abstract the concept of the memory and notes of a mathematician from such messiness as notebooks and a brain, into a one dimensional tape of paper and a person acting as a computer whose mind is in one of only a small number of states. The only allowed operations involve writing one of a small number (e.g. 10)of symbols onto a square space on the tape (erasing what was previously there), then moving the tape one square to the left or to the right, and finally changing the state from one of a small number of states to another state. The rules governing the operation of a Turing machine consists of a table of quintuples. The five items in each quintuple are:

1. The current state
2. The current symbol in the square in view
3. A new symbol to write in the square; replacing the old symbol
4. A new state.
5. An instruction to move the tape one square either to the left or to the right.

****

The number of different quintuples is at most the product of the number of symbols times the number of states. It is possible to have a Universal Turing Machine with a state-symbol product of only 3x. A Universal Turing machine can exactly mimic the behaviour of any computer given two things: a long enough tape, and an initial segment of the tape that has a suitable program on it. The Turing machine is clearly a model thought up by a mathematician as opposed to a physicist. Ulam and Von Neuman thought up another computer model that is more like physics in many ways; the Cellular Automata. Computer engineers prefer the Boolean algebra model of digital logic, mixed with a bit of automata theory. They also resort to arcane models such as Petrie nets, as a way to explore and solve complex synchronization problems. System programmers have come up with their own models; a list of all models of computation would be quite a sight.

A physicist, on the other hand, might want a model of computation that is made up of elementary parts and simple, mathematical rules; clearly in concert with the laws of

physics. For a long time, the concept of a physics based model was commonly thought to necessarily rest on the laws of thermodynamics; it was thought that all microscopic acts of computation were necessarily dissipative. What follows is a physically correct exposition of the concepts of computation that is based on simple Newtonian mechanics. It is based on the same concepts as the kinetic theory, with the exception that we look at the model in microscopic detail; not statistically. What we will show is that it is possible to model all microscopic aspects of computation better and clearer from such a physical model, while all macroscopic aspects remain as what we know from other models of computation. Once we have done this, this new computational model returns the favor, and allows what seems the nicest example of Maxwell's Demon at work.

Real computers have the distinction, amongst machines or systems, of corresponding most exactly to their abstract models. Most systems, say internal combustion engines that convert heat energy into mechanical energy, only crudely resemble their abstract models. Other systems match their models very well. A gearbox, whose output shaft turns at a rational multiple of the input shaft's RPM is quite good. On the other hand, if the model of the gearbox is that it the output torque is a rational multiple of the input torque, then the gearbox is only a crude approximation. Money is a good example, in terms of allowing the payment of bills and the making of change with great precision, but a poor example in terms of its value or purchasing power.

Computers are made up of millions or billions of parts, each of which might do millions or billions of operations per second, and each part does the exact thing expected of it every time. There is no approximation in the result. In contemporary computers, this is all due to the extreme quantization of signal and the existence of the appropriate transfer function. In other words, although signal noise is present at every microscopic action within a computer, the noise can nowheres ever exceed a very small threshold because it is converted to heat by every logic gate, while memories have sufficient hysteresis or signal thresholds to eliminate the possibility of being affected by noise. Of course this is all relative; we can construct a computer with any probability p of not making an error, $0 \leq p < 1$, given that certain environmental conditions are maintained. Economics causes us to accept computer circuits where 10^7 gates operating 10^8 times per second will collectively drop a bit no more than every 10^8 seconds provided they are protected from heat, cold, lightning, cosmic rays and other such trauma. That is about one error every every 10^23 potential operations. If you wanted one error for every 10^40 operations, the computer might triple in cost.

THINKING ABOUT COMPUTATION IN A CLOSED SYSTEM

"The game isn't over until it's over" -- Yogi Berra

It is very easy to get confused about what computation is unless computation is first modelled as a closed system. This means that we will ignore all input and output; essentially leaving it out of the model for the time being. It may be hard to understand how computation can be thought of without input or output, but it is really very straightforward. Imagine a computer that consists of nothing but a memory (RAM) and a processor. Somehow, the program and the data are already in memory. The computer runs until the computation is finished, at which point the results of the computation are in memory. We will first make a model that describes that part of computation, and then we

will show how to add input and output to the model. This is like considering the falling object between the time it is let go until some time before it hits the ground. We want to be able to answer question about how much of the resources of nature (such as matter and energy or space and time) are needed to do such a computation. With regard to energy, there are two parts to the answer, how much energy is needed to set the computer in motion, and how much energy is dissipated in doing the computation.

ENERGY REQUIREMENTS FOR COMPUTATION

> *"Do not become attatched to the things you like, do not maintain aversion to the things you dislike. Sorrow, fear and bondage come from one's likes and dislikes." -- Buddha*

If a most efficient supercomputer works all day to complete a weather simulation problem, what is the minimum amount of energy must be dissipated according to the laws of physics? The answer is actually very simple to calculate, since it is unrelated to the amount of computation. The answer is always equal to zero. It is much like looking at a machine, such as a heat engine, and asking how much energy must be dissipated in converting heat into mechanical energy. The answer is, as so beautifully elucidated by Carnot, always equal to zero. This is because Carnot leapt over all other conditions and constraints by observing that a reversible process did not dissipate energy. Strangely enough, the necessary dissipation is only related to the amount of data thrown away at the end of the computation, and unrelated to the amount of intermediate data generated during the computation. Thus the amount of energy that must be dissipated by a computation that answers the question "Is 3 an odd number?" is exactly the same as must be dissipated by a computation that answers the question "Is the 1,000,000,000th digit of Pi an odd number?"; zero in both cases.

Rather than calculating how much power must be dissipated by existing computers, we will accurately calculate how much power must be dissipated, according to the laws of physics. It has been formally proved that, in theory, reversible computers can carry out any finite computation in the following manner:

1. Some of the inputs (including the program and some other initial conditions) are loaded into the computer from a reversible source, such as a reversible tape cassette. This involves the main computer swapping constants or other data for the information on the tape. Since the computer is only borrowing the data, and will give it back at the end of the computation, this process need not dissipate any energy.
2. The rest of the inputs could come from an irreversible source, such as a person typing at a terminal or from instruments and sensors.
3. The computation would proceed in a reversible manner, dissipating no energy in the process. All intermediate results would be unprocessed back into the data used to calculate them. This whole process would dissipate no energy.
4. When the computation is finished, the reversible tape cassette would again swap its contents with the computer for the program and other initial conditions it supplied at the beginning of the computation. At the same time, the computer would receive back the data that it had earlier swapped into the tape cassette.
5. The output data that is produced in an irreversible form (printout or computer displays) cost kt per bit for the irreversible conversion of constants into data. In addition, the data received in 2 above would be cleared or printed; again at a cost of kt per bit.

To be specific, we will imagine that the initial conditions for the computation consist of a 5 megabyte program, and a 95 megabyte set of initial conditions. Let us assume that the output consists of a few kilobytes of tabular information, 10 megabytes of weather maps and various people perusing data by looking at 100 screens of high resolution displays on an RS-6000 workstation along with 10 minutes of computer generated animation.

THE MODEL

In order to model computation in a physically correct way, we will follow the lead of computer engineering by identifying the atomic parts of computers, and show that compositions of these parts can implement essentially any kind of computer. The difference will be that we will use parts that are physically correct rather than just logically correct.

When we say "computer" we are referring to that part of a computer system that computes, sans input-output devices. In other words, by "computer" we mean the central processing unit and its memories. In the abstract, ordinary commercial computers are built out of logic gates and wires. In real life, transistors, conductors, capacitors, insulators, power supplies and cooling constitute the essential kinds of elements found in computers. The normal abstract engineering model of a computer is a diagram consisting entirely of gates and wires along with a few other things that could be replaced by functionally similar things made of gates and wires; eg a magnetic hard disk could be replaced by a functionally similar device made of gates and wires. The rules are simple, each gate has inputs and an output. Each output can be connected to a few inputs. Although in ordinary computers, some things are not made up out of circuits of gates, in principle everything could be made up out of circuits of gates. The abstract model ignores things such as power, heat dissipation, packaging, and even many aspects of timing. You can be sure that the computer architects still have to pay strict attention to the realities of good computer engineering.

While, in practical computer engineering, many different kinds of gates are used it is sufficient to settle on just one kind of universal gate such as the 2 input NAND gate or the 2 input NOR gate. All other gates can be synthesized out of combinational circuits of just one type of universal gate such as the NAND gate. The abstract models of Boolean algebra and Automata Theory that represent modern computers does not take into account the physics of what happens; just the informational aspects.

Here are a few examples of practical computer circuits built out of NAND gates.

We will now develop a different abstract model that is based entirely on physically correct computational atoms and processes. At first it seems that me must account for three kinds of processes: computation, memory and communication. We will first explain why, in a formal, physically correct model of computation, memory and communication are the identical process. Second, we will look at the atomic parts, which turn out to be two kinds of particles. For space, we will use a 2 + 1 dimensional space, chosen mainly because of its correspondence to a drawing, on paper, of a computer circuit.

In kinetic theory, we can deduce the properties of ensembles of particles as the average results of the consequences of individual particles following simple dynamical laws. However, assume we know nothing more than the following rules:

1. A simple gas that is compressed isothermally gives off heat and increases in pressure.
2. A simple gas that is compressed adiabatically increases in temperature and pressure.
3. Processes 1 and 2 are reversible, ie a simple gas that is expanded isothermally absorbs heat and decreases in pressure.
4. A perpetual motion machine is not possible.

Assumptions 1, 2, 3 and 4 are sufficient to show that a Carnot engine is the most efficient possible way to convert heat energy into mechanical energy. The reason is simple; a Carnot engine is reversible, therefor if a Carnot engine can convert heat to mechanical energy and then convert that mechanical energy back into heat with exactly the same efficiency, then, assuming 4 above, we have proved that nothing else can be more efficient. For example, say that you believe that an apparatus involving lots of magnets (a magnetic engine) can do better. Then let your magnetic engine do its work and let a reversed Carnot engine undo it by running backwards. Since your engine is more efficient, the reversed Carnot engine will convert the mechanical energy produced by the magnetic engine into more heat energy than was used by the magnetic engine to produce the mechanical energy. This means that we would get a steady and inexhaustible source of free heat energy, violating rule 4 above. QED

The above argument makes no use of the kinetic theory; as far as Carnot is concerned his results are correct even if the working fluid is continuous rather than a gas made up of molecules. This is because everything that is shown true about the Carnot engine derives from just reversibility and conservation of energy. What we are going to do is to assume reversibility at a microscopic level and so simplify kinetic theory that we can understand the exact evolution of each and every particle. Strangely enough, we will also be able to prove that it will not be possible, in general to calculate, by any analytic method, any good statistical measures about the evolution of a given initial condition. Aside from modelling computation, our model will also be able to shed some new light on both mechanical engines and such conceptual engines as Maxwell's demon.

THE BILLIARD BALL MODEL

Particles

We assume that there are two kinds of particles: light particles and heavy particles. All particles are perfect spheres with a radius of one. These particle are imagined to be just like perfect billiard balls with the exception that they are only engaged in translational motion. We assume that they are all of the the same size, and we ignore friction or other sources of errors in the particles. When two particles collide, they simply bounce elastically, conserving kinetic energy and momentum with no losses or other forms of energy dissipation in the process. The most unusual aspect of the model has to do with the initial conditions.

Bridging the Gap

"By convention there is color, by convention there is sweetness, by convention bitterness, but in realty there are atoms and space." -- Democritus

Of course it is conceivable that we could simply find the magic equation and in one fell swoop recreate all of physics in a new genre. To us it seems too early to take that approach, rather we feel a need to understand how more of the simple things can be done before we attempt to do everything. In summary, we have worked to knock down, one at a time, the bugaboos that made cellular automata seem inappropriate, while finding ways to model more and more of the properties of physics. Of course we hope to put it all together with a grand equation or rule that is the answer to everything, but this may take the cooperative efforts of those who know a lot about computation as well as those who know a lot about physics. Once we have *the rule* we will be able to work with it in two ways: analytically, as with any mathematical equation, and through simulation where we use computers or specially built Cellular Automata Machines. The usefulness of simulation will depend on the scale; if it's at Plank's length, then we will be unable to compute much. If however, the scale is closer to a fermi, then we may be able to compute a lot. The rule would be the only parameter of this model of physics. From the rule it should be possible to compute every constant of physics, and answer every question posed at the appropriate scale.

Scaling, Anti-Scaling and Common Sense

"The whole of science is nothing more than a refinement of everyday thinking" -- Albert Einstein

Digital Mechanics has one set of properties that scale and another that do not scale. The informational processes that look like field equations must scale. The informational process that constitutes a particle will not scale. An electron is a certain size, has a certain mass and a certain charge. These quantities do not scale because they would be a direct consequence of the rule and the lattice size. If you could erase physics from the mind of a computer scientist, he would be hard pressed to imagine how a particle could emerge from a programmed model of physics if there were no fundamental unit of length, nor could he imagine how uniform motion could occur without a fixed reference lattice. A scientist steeped in the field of computation begins to understand certain laws of nature that have not yet been formalized. For example, "What cannot be programmed, given the necessary resources, cannot be physics." What this means is that if we can prove that we cannot program any kind of computer to model telepathy in accordance with the laws of physics, then telepathy cannot be part of physics. No physics experiment can, in general, give an answer to the halting problem. We believe that we cannot program rectilinear motion without a fixed reference frame. We cannot program particles without a unit of length. We cannot program angular orientation (and consequently angular momentum) without coordinate axes. In short, its not that we see DM as a possible model of physics, rather its that we see no way to model physics without the incorporation of much of what is in DM!

Digital Information Mechanics

In 1982, we compiled the set of ideas then known as "DIM" into a paper that was privately circulated[1]. That paper was based upon a talk given at the Mosquito Island

Conference on Physics and Computation in 1982. While the ideas presented were not ready for publication, many of the concepts reported here were well worked out at that time. All of this has led to the present, where we finally feel that an interesting number of pieces of the puzzle are on the table. What we can show are hypothetical components of a theory. It does not all fit together, but then we are working on a very big puzzle. The reason to publish now is that the current set of pieces are to us, more compelling than ever; beyond an important threshold. We believe that it should now be reasonable for physicists to become interested in this work.

DM and RUCA

We must carefully distinguish the RUCA from the informational process that may be running in the RUCA. This is similar to distinguishing a chess board, the chess men and a book of the rules from a game of chess. For each RUCA, there are one or more such Informational Processes. We call the systems of such informational processes that run on a RUCA and that in one way or another model physics "Digital Mechanics". Our common, sensible, world (intuitive physics) and RUCAs share certain properties in common: they are both universal, locally Euclidian and locally connected. In common with modern theoretical physics, RUCAs are microscopically reversible, have certain symmetries and conserved quantities. In other ways RUCAs seem very different than current concepts of theoretical physics, they are: globally cartesian, simply deterministic, non-isotropic, have a fixed reference frame and are discrete in all regards.

What is very surprising is that there is no fundamental reason why Digital Mechanics must have behavior that distinguishes it from microscopic physics. Our world (physics) and DM seem to share many more things in common than physics and RUCAs, in that DM can be Relativistically correct[2], apparently non-deterministic[3], asymptotically isotropic[4], perform as though there is no fixed reference frame, model apparently continuous phenomena with great or perfect accuracy, produce a stable of particles as easily as Conway's game of life[5] produces gliders, puffer engines, and other stable life forms out of a simple CA. We will show that DM systems with a local rule nevertheless have the property that the state of a cell at a particular time can be a function of the state of cells throughout the entire space at the same time. This makes us optimistic about the possibility that DM may be capable of using mechanistic, deterministic and local rules as a substrate and yet produce behavior that obeys the laws of QM. In short, we have good reason to believe that DM is not incapable of manifesting some, most or all phenomena associated with fundamental particles and processes in physics.

Modelling Macroscopic Phenomena

It has been discovered through experimentation that cellular automata can be constructed to behave in ways similar to higher order physical systems; in other words, they are useful for simulating physical phenomena including hydrodynamics[18], percolation, nucleation[6], the movement of sand on the beach, and other phenomena[7]. We distinguish such systems from DM, which is only a direct model of the most microscopic physics. We do not anticipate that direct DM models will be useful for anything other than such microscopic models, but the principles of DM may be very widely applicable.

We will discuss many properties of DM in the context of models of physics. Rather than covering the spectrum of possible RUCAs and DM systems, we will describe a hypothetical yet plausible DM that combines properties that we have seen in various other DM systems; we will match up those properties with corresponding properties of physics.

DM-4, The Nature of a Rule

> *"So I have often made the hypothesis that ultimately physics will not require a mathematical statement, that in the end the machinery will be revealed, and the laws will turn out to be simple, like the chequer board with all its apparent complexities." -- Richard Feynman*

DM-4 is a hypothetical member of a class of typical DM systems, some of which may be suitable for modeling physics. The purpose of this example is not to put forward a candidate for modelling physics, but rather to impart to the reader the flavor of what such a candidate might be like. We will first give a general definition of the kind of RUCA which runs DM-4.

The RUCA for DM-4 is a 4 dimensional space-time lattice. In spatial extent, x, y and z, the RUCA can be any size, according to resources. If programmed on a Macintosh, a reasonable size might be 64x64x64, for a Cray, one might do 256x256x256, and a specially constructed cellular automata machine, using todays technologies might handle more than $1024x1024x1024^8$. Modeling the whole Universe would require a very large but finite array. In the time dimension, the system has a depth of 2. This is because it is a second order system, which requires both the *present* and the *immediate past* in order to calculate the *future*. There are many different kinds of rules that are possible, but we have chosen one kind for concreteness. The RUCA is a synchronous and deterministic system that operates according to a simple rule. There is a clock that governs the timing of the system, and the clock is a 6 phase clock; instead of going "...tick, tick, tick,..." the clock goes "...tick, tock, tack, toock, teck, tuck, tick, tock, tack, toock, teck, tuck, tick.....". Various phases of the clock are associated with the fixed coordinate system. There are six directions, North and South, East and West and Up and Down. Each phase of the clock is associated with a single direction, in the order East, Down, South, West, Up, North. This sequence is abbreviated EDSWUN. It is easy to design such systems so that certain properties are either present or absent. However, there is not yet any magical method for designing just what you might want; you have to try it out and see if it does what you want.

The total operation of DM-4 is determined by the geometry and connectedness of the RUCA, the rule, and the initial state. More accurately, the complete time evolution is determined, yet in general it is unknown and unknowable in advance. This is an interesting consequence of such systems; while they are simple and deterministic, there is no shortcut to determining their future; every step must be carried out (in general). This means that there is no possibility of a *superman* within the system, who could predict any exact future state before it happens. The amount of work to predict the future exactly for any non-trivial, closed RUCA is always proportional to the space-time volume. "Unknowable Determinism" is a good description of what's happening in all DIMs.

When a RUCA is set to an initial state and set into motion, it is nearly always full of surprises. Of course, a poor choice of rules or initial conditions can quickly result in pure

chaos or perfect order, neither of which seems as interesting as the right combination of chaos and order. Many rules have structures that persist, objects that travel in various directions and at various speeds, and interesting interactions when objects collide. One of the most famous of such systems is Conway's game of life[9], a simple two-dimensional system that is full of interesting objects. However, Life, in its basic form, is definitely not a RUCA; it can be universal but it is not reversible.

The Basic Dimensional Units

The RUCA and its objects can be characterized by certain dimensional units. There are 3 basic dimensions, from which others can be synthesized. They are: the Digit-transition, Length and Time, D, L and T. The L and T are different than the L and T of ordinary physics. First of all, T can be thought of as an integer, and counting! Whether T is odd or even is important, and affects everything that happens everywhere in all DM systems. In DM-4 the value of T mod 6 is important. The RUCA T is not Lorentz invariant, and while it is locally similar to the time of physics, it is not the same thing. The unit of time is one tick, tock, tack, toock, teck, tuck cycle.

Numbers in the RUCA

The Unit of Length in the RUCA is the distance from one cell to its nearest neighbor. The dimension of Length L, is similar yet different from the L of physics. Values of L are always numbers that have finite descriptions. All effective values (or numbers) within a DM system are capable of having a finite representation. They may be real or complex. This is a simple consequence of the fact that the RUCA is finite; there is not enough information in a finite RUCA to represent even one arbitrary real number (or even a rational number with too large a numerator or denominator). Many real numbers have a finite representation, such as "Pi" or "$2^{1/2}$". Such representations of real numbers along with the concept of a real number can exist in a DM system.

Asymptotic Isotropy

While DM can be asymptotically isotropic, it cannot be microscopically isotropic. The RUCA has a preferred and absolute coordinate system and is definitely non-isotropic. The same is true for the DM system on the microscopic level; on the other hand there is no doubt that a DM system can be asymptotically Lorentz invariant and asymptotically isotropic. Nearly all trace of the preferred space-time coordinate system, its anisotropy, its absolute reference frame and its absolute lengths and times can be totally washed out so as to become relativistically correct as the scale of events moves away from the most microscopic.

One way to see why asymptotic isotropy can be a consequence of a basic conservation law in the DM system is as follows. We know that conservation laws can be a consequence of certain symmetries. For example, translational invariance leads to conservation of momentum. This concept can be thought of as a two way street; conservation of momentum implies translational invariance. It is easy to create a cellular automata rule where the quantity associated with angular momentum is conserved as a consequence of the most basic operation of the rule. This microscopic conservation of

angular momentum should allow us to derive a version of angular isotropy that is true at some scale above the most microscopic.

The Digit Transition

> *"Suppose that physics, or rather nature, is considered analogous to a great chess game with millions of pieces in it, and we are trying to discover the laws by which the pieces move." -- Richard Feynman*

The most novel dimensional unit is the Digit-transition. We are assuming that the information in each cell of the RUCA is 2-state or 3-state; one bit (for 2-state) or one trit (for 3-state) per cell. We will use the word "digit" to mean a bit or a trit. Of course such systems are possible with any number of states per cell. Each cell contains just one digit, therefore a digit-transition must take place over space, time or space-time. All other kinds of dimensions of the physics of DM-4 are derived from D, L and T. In DM, the dimensions of the RUCA's digit-transition is the same as for Plank's constant. A series of digit-transitions can have angular momentum and a digit transition's numerical value is very simply related to Plank's Constant.

The principle of *least action* takes on new significance in DM. Since action has the units of the Digit Transition, it is related to the amount of information in a process.

As we have begun to understand DM, we have realized that there must be certain universal laws that govern the behavior of such systems. For example, "There can be no information without a means of its representation." This means that if a particle is moving with a particular velocity, it must be true that there is an interpretation of a conglomeration of bits of information in the system that represent the information that describes the particle's velocity. "For a process to evolve, there must be a means of interpreting and transforming the representational information." For example, if a particle is to move, then the information that represents its velocity must interact with the coordinate system (be, in effect, processed by the RUCA) to produce the change in position over a series of time steps that is in accord with the velocity. If a field is accelerating a particle, this means that the information that represents the the state of the field must interact with the information that represents the velocity of the particle so as to change the information that represents the velocity of the particle. Of course we may find that the information that represents the state of the field is carried by digital version of a boson.

Certain quantities must be conserved. There is something loosely called "information" and in DM, the thing called "information" is conserved. Depending on both the rule and the initial conditions, there will be other conserved quantities that might correspond to energy, momentum, angular momentum, charge etc. that would also be conserved. Every RUCA transitions from one state to another along a closed trajectory. We will show a physically impractical but mathematically correct method of determining a unique quantity for each trajectory. Since this quantity can be determined for each trajectory and because it is different for each different trajectory, it is a unique conserved quantity. This means that there is a one-to-one mapping between trajectories and distinct sets of conserved quantities. In other words for a given RUCA that is in a particular state, there is a value of a conserved quantity for the trajectory associated with the state that is

different than that value for any state that is on a different trajectory. This can be proven as follows.

Assume that the RUCA is on a trajectory of length T. Assume that the number of cells in the x direction is X, and similarly Y for y and Z for z. For each of the XYZT values of the set [x, y, z, t], we map the state of every cell, $C_{x, y, z, t}$ into a one dimensional array $D_{w.}$ There may be some small number like 24 different indistinguishable ways of doing this for each cell, so we will do them all. We then consider each of the Ds as an integer. Every different D is unique to the particular trajectory. Of all of the Ds produced, we choose the smallest value V (it is OK if there are many instances of that value). V is a unique quantity of the trajectory, so we may say that it is a conserved quantity. This approach generates the same value for the conserved quantity V for distinct trajectories, that differ from each other only by displacements, rotations or reflections.

Memory and Communication

Memory and communication, must be the same physical or informational process, differing only by a coordinate transformation[15].

We can define a fundamental act of communication as follows:

> $R_{x', y', z', t',}<=S_{x, y, z, t}$; we Receive, at x', y', z', t' information Sent earlier at x, y, z, t.

We can define a fundamental act of memory as follows:

> $R_{x', y', z', t',}<=S_{x, y, z, t}$; we Read, at x', y', z', t' information Stored earlier at x, y, z, t. Normally, x', y', z'=x, y, z.

It should be clear that we can always transform memory into communication by means of a coordinate transformation, and similarly for transforming communication into memory. When we write on a floppy disk it is usually memory, but when we buy a program on a floppy disk, it serves as communication. Thus it is only our intent that separates memory and communication. From the point of view of physics, they must be seen as one and the same. If two processes are related by a coordinate transformation, they must be the same process. Within a DM, there is only one process, and it serves as memory, communication and computation.

---

[1] Fredkin, Edward, "Digital Information Mechanics," Privately circulated paper, 1982.

[2] Toffoli, Tommaso, "Four topics in lattice gases: Ergodicity; Relativity; Information flow; and Rule compression for parallel lattice-gas machines," to appear in the proceedings of an International Workshop on Discrete Kinetic Theory, Lattice Gas Dynamics and Foundations of Hydrodynamics, Institute for Scientific Interchange, Turin, Italy, 20-24 September 1988.

[3] Wolfram, Stephen, "Random-Sequence Generation by Cellular Automata, " Adv. Applied Mathematics 7 (1986), 123-169

[4] Frisch, Uriel; Hasslacher, Brosl and Pomeau, Yves, "Lattice-Gas Automata for the Navier-Stokes Equation," Physics Review Letters 56 (1986), 1505-1508.

[5] Gardner, Martin, "The Fantastic Combinations of John Conway's New Solitaire Game of 'Life'," Scientific American 223:4 (April 1970), 120-123

[6] Creutz, Michael, "Deterministic Ising Dynamics," Annals of Physics 167 (1986), 62-76

[7] Toffoli, Tommaso and Margolus, Norman, "Cellular Automata Machines - A New Enviornment for Modeling, MIT Press (1987).

[8] Margolus, Norman; Toffoli, Tommaso, "Cellular Automata Machines," to appear in Large Nonlinear Systems, Gary Doolen ed. (1988), as a revised version of a paper by the same title which appeared in Complex Systems 1, 967-993 (1987).

[9] Conway, John, Invented the Game of Life in the late 1960's; see 19 above.