Universal Field Machine that Computes Beyond the Turing Limit

Julián M. Fernández[*]
*Physics Department,
National University of La Plata,
C.C. 67, 1900 La Plata, Argentina*

A model of an idealized machine is proposed that makes use of various "physical" fields to perform computations. Its computational power surpasses that of a Turing machine, but the set of functions that it computes is still countable (on the contrary, analogic computation can compute any function, i.e., a noncountable set). As a major difference with analogic computation, the field machine has the important advantage of being programmable (i.e., universal field machines exist), and because of this, it is possible for a hypothetical operator of such a machine to know which problem the machine solves.

1. Introduction

The Church–Turing thesis, the prevailing paradigm in computer science, states that no realizable computing device can be more powerful than a Turing machine [1].

In computer science, machines are classified according to the classes of tasks they can execute or the functions they can perform. The most popular model is the Turing machine [1], introduced by the English mathematician Alan Turing in 1936. The Turing machine allows for unbounded "external" storage (tapes) in addition to the finite information represented by the current "internal" state (control) of the system. At every step, the machine reads the tape symbol α (where $\alpha \in \{0, 1, blank\}$) from under an access head, checks the state of the controls (where $s \in \{1, 2, \ldots, |S|\}$), and executes the following three operations.

1. Writes a new binary symbol under the head ($\beta \in \{0, 1\}$).

2. Moves the head one letter to either right or left ($m \in \{L, R\}$).

3. Changes the state of the control ($s' \in \{1, 2, \ldots, |S|\}$).

The transition of the machine can be summarized by a function

$$f : (\alpha, s) \to (\beta, m, s'). \tag{1}$$

[*]Electronic mail address: fernande@venus.fisica.unlp.edu.ar.

When the control reaches a special state, called the *halting state*, the machine stops. The input–output map (or the function) computed by a Turing machine is defined by the pair of finite binary sequences on its tape before and after the computation, respectively. The finiteness of the input and output is a crucial requirement in computer science, assuring that the ability to compute depends purely on the inherent computational power of the model rather than on a larger amount of external information.

There are many variants of the basic model that yield the same power, such as adding tapes, access heads, and tape dimensions.

Other models of computation exist that are more powerful than Turing machines, though such models are physically nonrealizable. However, they are important from a theoretical point of view.

One such model is the model of computation over the real numbers [2]. This interesting model, however, is incompatible with the modern theory of computability, as it does not comply with the finiteness of input and output.

Siegelman and Sontag [3] introduced a model of computation based on the classical analog recurrent neural network (ARNN). This model complies with the finiteness of input and output. They demonstrated that if at least one of the weights is real, the ARNN can in principle compute all functions. Of course, no single ARNN computes all functions, there is no equivalent of the universal Turing machine. It is not a programmable machine. The ARNN only "extracts" the information "codified" on its real weight. If this real weight is a non-Turing-computable number, the ARNN can compute a non-Turing-computable function. We need a noncountable number of ARNNs to be able to compute all possible functions. We also have the disadvantage of not knowing which problem a particular machine solves, due to the fact that we do not know which problem is codified by the real weight (because we cannot know the infinite number of digits of this weight). We will discuss this question later.

Another recent interesting model is the "analog shift map" [4]. It is interesting because it is a natural model of chaotic idealized physical dynamics. For example, it is equivalent to the motion of a particle in a three-dimensional potential, such as a billiard ball or a particle bouncing among parabolic mirrors [5]. The idealization allows for model assumptions such as any convenient scale to describe the system, noise-free environment, and physics of continuous medium. However, this model suffers the same problem as the ARNNs.

All mechanisms proposed until now for analogic computation are equivalent to a physical device that, by means of some method, "measures" with infinite precision some parameter that is inside the system (e.g., the weight connecting two neurons in an ARNN, the initial position of a particle, and so on). In this way, we have in principle the capacity to "compute" any real number (from now on we use the verb *compute* in an extended way, in such a manner as to include nonalgorithmic physical processes).

However, this method has two major problems. (i) In general the machine stores an infinite quantity of information on its real parameters. This infinite information cannot be obtained and analyzed by an observer to figure out which problem the machine solves. We know that the machine is solving a problem (or making a computation) but we cannot characterize this problem in any way. We do not have access to the entire "program" of the machine. We can, for example, have a machine that solves the halting problem, but we will never know that the problem the machine is solving is the halting problem. (ii) Related with the previous point, they are only "purpose specific" machines (this purpose, as mentioned before, is notwithstanding unknown), that is, they are nonprogrammable machines.

An interesting fact would be to find a computational process that allows us to know which problem the machine is solving, or equivalently that it would be programmable. This is a very important fact.

It is clear that in this case the number of computable problems will still be countable and so the computational power of the machine will be inferior to that of analogic machines, but it could be superior to that of Turing machines.

In this work we introduce a system of differential equations whose solution is not computable through approximative algorithms and is equivalent to an ideal physical system consisting of a set of interacting fields. This system also allows for the construction of universal machines.

2. Physics, mathematics, and the meaning of the physical laws

Before we study the necessary requirements for the construction of universal field machines, we must discuss an important question regarding the relation between physics and mathematics. In mathematics, we can have a system of differential equations that depends on several variables, some of which can present discontinuities, or could be undefined on some points or in some regions of the space. If the variables of the system represent some physical quantities a question arises: What does it mean that a physical variable is not defined in some region? There are several answers for this question. One possibility is that the system of equations is only an approximation to the real physical system. For example, if a discontinuity appears in the system of equations, in the real physical system this discontinuity is only approximate, in such a way that if we observe the system in an adequate spatial scale the discontinuity disappears. One example would be the velocity of a particle when it hits a rigid wall. In the precise moment of the collision the velocity is not well defined (it changes from $+v$ to $-v$), but we know that this is only an approximation.

Physicists usually assume that all real physical variables are continuous and well defined. However, this is not necessarily so. Due to the fact that physical laws only represent the behavior of a physical system in terms of fields or other variables that do not necessarily "actually" exist as a real physical entity, it is not strictly necessary that these fields must always be well defined or continuous. In fact, even if they were real physical entities they would not *a priori* be exempted from being discontinuous or undefined (this is only an aesthetic imposition of modern scientists).

As an example we can mention the classical problem of the value of the electric field over an electron. If we assume that the electron is an elementary, nondivisible particle, of zero dimension, then, the electric field diverges as we approach the electron (if Coulomb's law is still valid). Then two possibilities arise: (i) the electric field has an unknown value at the electron's position, or (ii) it is not defined in that point of the space. In either of these two cases the electric field should be considered a discontinuous quantity. In the past this was not considered an approximation. Scientists had no problem in accepting this fact. Of course, quantum mechanics radically changed that picture.

An intermediate and less extreme example would be that of a physical system that is incompletely described by a set of laws in such a way that these laws, when giving predictions, give the correct ones, but there are also cases in which these laws make no predictions: they do not permit calculating or computing the values of some variable. This would happen, for example, when some variable is not defined on some region of space.

If the discussion above is not convincing enough, we should take into account a much stronger argument, one of the most important results in the science of the past century: Goedel's incompleteness theorem, applicable to all formal systems that include the arithmetic of natural numbers. Even if physicists often do not care about the implications of this theorem, it implies that for any complex enough physical universe described by mathematical laws, the laws are by no means complete.

This idea is at the core of the working of the field machine: we rely on a universe which has noncomputable physical laws, and this fact will help us in solving problems which are noncomputable in the classical sense, like the halting problem. The trick is to master noncomputability in a physical world (not our own, but a hypothetical one) in order to help us in making computations that cannot be done with a standard Turing machine.

Summing up, the fact that a field or variable utilized for the description of a physical system is not defined in a region or point of space could signify that: (i) the field or variable does not exist or cannot be defined in such a point or region, or (ii) it exists and has a defined value, but the equations we possess do not permit computing it because they are incomplete. In the following we will not be interested in which of the two situations applies to our system, but we will only make the assumption that when the equations make predictions, they are correct.

3. The field machine

The conception of the machine is very simple and is based on an almost complete analogy with a Turing machine. The basic idea is to build a device similar to a Turing machine but for which the undecidable problems become decidable. For a Turing machine, we can generate noncomputable problems using the complement of an undecidable problem. In the field machine, these problems are computable because the original one becomes decidable. To build a device that overcomes the undecidability of a Turing machine we need to rest on certain hypothetical physical laws. Of course, from a theoretical perspective it is not important if our particular universe supports or does not support that set of laws, but only if such a set of laws could in principle exist.

In the following, we will use a different version of a Turing machine than that described by equation (1). In general, we cannot rewrite the function f from equation (1) as:

$$f = g \circ h \tag{2}$$

where $h : (\alpha, s) \to (\beta, s')$ and $g : (\beta, s') \to m$, because in equation (1) two different input 2-tuples $(\alpha 1, s1)$ and $(\alpha 2, s2)$ can give the same (β, s') but different m, and this is not possible in equation (2). However, it is always possible to build a new Turing machine, with a function f' in equation (1), that computes the same function as f, but such that f' can be written as $f' = g \circ h$ (this can be done adding additional states which remove the ambiguity of the input 2-tuple).

So, in the following we will assume that the head turns to the left or to the right as a function of the new symbol written on the tape and of the new state taken by the machine.

The field machine consists of a square region with sides of length one. Let us call x and y the cartesian coordinates of a point inside the square. We can imagine the square as divided into an infinite number of cells, each one of them able to contain a symbol. The symbols can be either "one" or "zero," corresponding respectively to the spin up or spin down states of a field $S(x, y)$ on the cell. If the field has zero value on the cell, there is no symbol there (or there is a blank symbol if you like). It could also be the case that the field is not well defined in a region of the space. In such a case the symbol is also not well defined, this case is treated in more detail later.

The size of the imaginary cells is not uniform. The size of the sides decreases as we are closer to 1 (on the x and on the y axis). One possible election is to choose the positions of the cells as

$$(x_n, y_m) = (1 - 1/2^n, 1 - 1/2^m); \qquad n, m \in [0, \infty).$$

However, any other election for the positions (and sizes) is valid (in fact they will be determined by the physical laws governing the fields).

The cells corresponding to $n = 0, m = 1, \ldots, \infty$ (i.e., the first imaginary column) represent the input tape of a Turing machine, and the values of the field $S(x, y)$ must be clamped in this region at the beginning of the computational process. The successive columns (i.e., for increasing values of n) represent the successive tape states for the successive time steps of the associated Turing machine (ATM).

The field $S(x, y)$ must satisfy field equations of a very special kind in order to be able to simulate the behavior of a Turing machine in the manner just described. We will not explicitly write these equations, because the system will in general be very complex. There exists an infinite number of possible sets of equations that are suitable for describing the system. Our intention is only to show a proof of existence, to see which are the necessary characteristics that this set must fulfill. We will see that these requirements are indeed possible for the equations describing the evolution of a set of fields. To give an explicit particular set of field equations that fulfills those conditions is beyond the scope of this article.

At the beginning of the computation, and as an initial condition, the static field $S(x, y)$ will be put in a configuration such that: (i) its value does not depend on the x coordinate, and (ii) in the y direction, the field fluctuates according to the values corresponding to the initial configuration of the tape of the ATM. The idea is that, when we put into the system (from the left) a "beam" of a propagating field $F(x, y)$ that interacts with $S(x, y)$, the values of $S(x, y)$ will change as F propagates towards positive x values. Because F is a beam, that is, its value is nonzero only in small discrete regions of the space, then S will change accordingly. The changes must be in such a way that the value of $S(x_n, y_m)$ is in correspondence with the state of the symbol at position m at time t_n in the tape of the ATM. If so, after a finite time (the time F needs to propagate to the end of the machine, i.e., $x = 1$), the configuration of S will contain the configuration of all the states of the tape (an infinite number if the ATM does not stop). Let us remember that the tape is infinitely compressed as we approach $x = 1$ (and also as we approach $y = 1$). It is worth remembering that $S(x, y)$ is not defined in $x = 1$ and $y = 1$.

Of course, at first sight there are some problems with this idealized machine. We will analyze and try to overcome all of them below.

First of all, what happens with S for $x \geq 1$ if the ATM does not stop? As we will see, the set of equations followed by the fields of the system make S (and F) not defined for $x \geq 1$ in such a case. This means that, in practice, when we try to measure S (or F) for $x \geq 1$ we will obtain a zero, a random value, or an inconsistent value. On the other hand, for the case in which the ATM stops, S will have a well defined value, and things will be defined in such a way that the field F will be in a particular state for $x \geq 1$: the halting state. To prevent errors, we can make the probability of F having the halting state when the ATM does not stop as low as we want. This can be done by letting F possess a very high number of possible states (in this case the probability of F reaching the halting state by random fluctuations would be very low). We will return to this later.

Still an important question remains: Could we actually find a set of fields whose behavior is the one we described before? First, let us remember that in a Turing machine, the tape is modified at discrete times according to a function

$$f : (\alpha, s) \to (\beta, m, s')$$

that also depends on the current position of the machine head.

Let us remember also that in our machine the tape subdivisions (in the y direction) are imaginary, and we assume them to be situated at the positions mentioned before. These positions must be defined when we impose the initial state of the tape, and at these same positions the field F must produce the changes as required. In the x direction, the successive states of the tape are separated in a different way. They are defined by the negative values of a field $P(x, y)$ which does not depend on time. The field $P(x, y)$ fluctuates between positive and negative values in the x direction and does not depend on y (as an example, we can choose $P(x, y) = \sin[1/(x-1)]$). The reasons for introducing this new field will be seen later.

The equivalent to the position of the head of the ATM is in our machine given, as stated, by the field F, which is different from zero only in a narrow region of the space. It is then required that, as this field F propagates, its thickness adapts to that of the cell that lies at the same position. It is also necessary that the direction and velocity of propagation in the y direction be automatically adjusted in such a way that the position of F corresponds with that of the head of the ATM. All this can be accomplished if the field F propagates as a soliton [6, 7]. For this to be possible, we need to introduce a new field $n(x, y)$ (a kind of "refraction index") with which F interacts nonlinearly [6, 7].

Also, as mentioned above, we need to give F an extra (internal) degree of freedom represented by the variable q, that can take an entire number Q of different values or states. These different values represent the equivalent of the different internal states that the ATM can take.

The field P was introduced before to allow F to modify S only when P is positive, and to change direction only when P is negative. This makes the tape well defined for negative P values (i.e., S keeps a constant value over the cell so that the symbol is not ambiguously defined), and also allows the beam F to switch to another cell position without modifying S during the passage. When P is positive, F moves only in the x direction, and gradually modifies the state of S in such a way that when a negative value of P is encountered again, a new symbol (according to equation (1)) is found in that position of the tape (which corresponds to the state of the tape at $t+1$). Now, with P negative again, F acquires a velocity component in the y direction too, such that it will be in the correct cell position (corresponding to the position of the head of the ATM) the next time it has to modify a symbol. Summing up, the well defined tape states for successive times in the ATM are represented in this field machine by parallel vertical stripes (in which P takes negative values) separated by other vertical stripes corresponding to transition states (in which P takes positive values).

Now we have all the necessary ingredients to build the field machine. If adequate field equations are satisfied, our machine will implement equation (1) in the following way: The state of the tape (the field S) will change as the beam F propagates in the x direction. Due to the fact that F has a definite width (equal to that of a cell that lies in the same position) the symbols are modified one at a time: When the field P has a negative value, the field S is not modified by the presence of F, and F has a nonzero velocity component in the y direction. When P has a positive value, S is modified by the presence of F which moves in the x direction only. The velocity and width of F are adequately modified to match the position and size of the corresponding cell (these do not depend on time). This is accomplished by means of the nonlinear interaction with the field $n(x,y)$.

The sign of the vertical component of the velocity of propagation of the beam F depends on the value of S and also of q, that is, of the tape symbol and of the internal state of the ATM, as required by equation (1). This corresponds to movement to the right or to the left of the head of the ATM.

When x is such that P takes positive values, F begins to propagate in the x direction only. Also, the fields S and F interact in such a way that both can change their state, in accordance with equation (1), that is, both the internal state of the machine and the symbol on the tape are modified. Also, this new value of S must propagate to the right (towards $x=1$) in order to update the state of the tape. In the appendix we give more details about the way this could be accomplished.

The dependence of $n(x,y)$ with y must be such that the width of the beam F coincides with that of the cell that lies at the same position. Then, the transition occurs only in regions where P is positive. The transition begins when P changes from negative to positive and is totally completed when P is zero again and changes from positive to negative.

If at any moment the field F reaches the particular state q_h (corresponding to the halting state), it begins to propagate in a straight line in the x direction without interacting with the other fields (with the exception of n). The other fields are not modified either. In this way, if the ATM defined by equation (1) halts at some moment if it is given an input tape similar to the initial state $S(0, y)$ in the field machine, then the fields F and S will be well defined for $x > 1$ (the field F always taking the state q_h). If, on the contrary, the ATM does not halt, neither S nor F will be defined for $x > 1$. They will take null or arbitrary values, depending on the unknown laws that prevail in the particular hypothetical universe that the field machine inhabits. As we mentioned before, we can reduce to a vanishingly small value the probability that F takes the state q_h in these cases if we use a field F which has an arbitrarily large number of internal states not utilized by the field machine. In this way, a problem which is undecidable for a Turing machine becomes decidable for the field machine.

In order to make the field machine work in the way explained above, we only need a system of m differential equations that implements equation (1):

$$f_i(S, P, n, F) = 0 \qquad i = 1, \ldots, m \tag{3}$$

where the f_i represent differential equation operators over the fields.

The system of equation (3) can be arbitrarily chosen, whenever it satisfies the conditions and requirements mentioned before. Let us briefly summarize them.

1. The propagation of F must depend on n, in such a way that n regulates the width of the beam F and also its direction and velocity of propagation.

2. On the other hand, n must depend on S, in such a way as to produce the adequate change of direction in F in the regions where P is negative. In the regions where P is positive F must propagate in the x direction only.

3. S and F interact in the regions where P is positive, both changing its state in correspondence with equation (1).

4. This new state of S must propagate to the right faster than F.

If the Turing machine defined by equation (1) corresponds to a universal one, then the field machine will be universal too, that is, it will be programmable. This means that the "problem" computed by the field machine does not depend on the machine itself but on the program recorded on the input tape (the configuration of $S(0, y)$ on the field machine).

It is worth mentioning one last question. The field machine described above requires an infinite velocity of propagation for the field F. Let us see why this is so. As F becomes nearer $x = 1$ the width of the cells diminishes and so the number of transitions per unit time can grow without limit. In these circumstances the distance covered by F may also grow without limit. For example, it becomes infinite if the head of the ATM must go over the tape from the first cell to a fixed cell and then back to the first in a periodic motion, never reaching the halting state.

However, this problem can be avoided if we introduce another field that generates a shift and compression of the cells (and in fact of the entire system) on the y direction as the beam propagates (see the appendix for more details).

For simplicity's sake this question was not addressed when we considered the description of the field machine given.

4. Discussion and conclusions

If a field machine like the one described in this paper could exist in an ideal universe, it is necessary that the physical laws of such a universe are complex enough to generate a set of fields like the ones defined in equation (3) (or some equivalent ones). This must not imply any theoretical objection, because for the existence of a classical Turing machine the laws of the universe must satisfy certain minimum requisites too (we make no distinction between a platonic mathematical universe and a "physical" one like our own: the only difference between them is their complexity).

Our philosophical position here is that any set of equations may represent a potential universe, whose laws, by virtue of Goedel's incompleteness theorem, are by no means complete. The laws describing our universe must have such a restriction (incompleteness) also. On the other hand, we do not restrict the possible set of laws by imposing on them aesthetical or other *a priori* constraints (e.g., that the fields must be continuous). We know from the history of physics that these constraints can be of help in some cases, but also that they are often wrong. The only assumption will be that the field's interactions are local. Of course, this last restriction is not necessary, but we made this election just to keep some analogies with current physical theories.

The class of functions that the field machine computes (a countable set) is certainly superior to the class of partial recursive functions (the class of functions computable by a Turing machine) but is inferior to the class of all functions (a noncountable set). This is because an undecidable problem for a Turing machine becomes a decidable one for the field machine. Thus, one interesting question arises: How could we characterize the class of functions computable by a universal field machine?

In principle, the machine can solve all problems that can be reducible to halting problems. However, not all conjectures are reducible to halting problems; see, for example, [9]. Even if there are reasons to believe that most interesting mathematical conjectures are reducible to special cases of a halting problem (for the relationship between propositions in any formal system and halting problems in Turing machines see [8]), there are still some mathematical questions not reducible to a halting problem. However, these kinds of nonreducible questions have a tendency to be rather artificious and self-allusive [9].

In any case, the field machine cannot solve all "definable" problems (which are a countable set) but only a subset of them (those reducible to halting problems). As an example, it could be programmed to compute Chaitin's Ω number [8] (e.g., [9] to find out how to compute Ω from a machine that could solve the halting problem). The number Ω is one of the most important numbers in mathematics (which is noncomputable by a Turing machine, i.e., by any algorithm) and whose knowledge could be used to learn the solution of many nondemonstrable theorems in mathematics.

We can also ask some more questions: Does the class of functions computable by this kind of field machine contain all classes of functions computable by *any* kind of universal field machine? In other words, does a more powerful universal field machine exist? Does a more powerful universal theoretical machine which is not a field machine exist?

Even if the answer to the first question is positive and the answer to the second one is negative, we have seen that the field machine does not solve (or compute) all definable problems (which are clearly countable, as mentioned above).

As far as we do not know the answers yet, we propose a generalization of the Church–Turing thesis (let us remember that the Church–Turing thesis states that the class of partial recursive functions contains all computable functions): "The class of functions computed (in the general sense) by this class of field machines is equivalent to the class of all computable functions (i.e., the class of functions computable by programmable machines of any kind)." Of course, this is just a thesis and we do not have any proof for it.

Apart from the fact that in practice the computational power of a Turing machine will never be surpassed, the study of the theoretical limits of generalized computational processes will always be of interest. This study will allow us to classify the computational complexity that characterize the problems that are beyond the reach of the limited computational processes accessible to the human being (and/or to the particular universe that it inhabits).

Appendix

First, we want to sketch a possible way for the shape of the interaction between fields S and F in such a manner that it satisfies the requisites previously imposed.

One possibility is an interaction mathematically equivalent to a chemical reaction. In this way, the "substance" F_{q_i} is introduced by the left side of the cell, within which is the "substance" S_i. As the "reaction" proceeds, on the right side of the cell we will find the substances in the final state: F_{q_f} and S_f. Such a reaction must be completed in the width of the cell. Due to the fact that the size of a cell decreases with x, one possible way to be sure that the reaction is always completed is to impose that the coupling (or the "reaction constant") between F and S is proportional to

$$\frac{1}{\left(\frac{dP}{dx}\right)^2 + \left(\frac{d^2P}{dx^2}\right)^2}.$$

In this way as the size of the cell decreases, the intensity of the coupling rises and then the time needed for the completion of the reaction decreases.

However, there is a problem if the interaction is this way. Let us suppose that our machine has to carry out the following two transitions:

$$\begin{aligned} &S_0, F_{q_0} \to S_1, F_{q_1} \\ &S_1, F_{q_1} \to S_0, F_{q_0}. \end{aligned} \tag{A.1}$$

Clearly, we need an extra signal that indicates in which direction the transition must proceed. There are many ways to do this. A simple one is to introduce a new degree of freedom in F, let us call it γ. Then, we can label F as $F_{q_i\gamma_j}$. Now, γ regulates the direction in which the transition proceeds. The transitions described in equation (A.1) can now be written as

$$\begin{aligned} &S_0, F_{q_0\gamma_0} \to S_1, F_{q_1\gamma_0} \\ &S_1, F_{q_1\gamma_1} \to S_0, F_{q_0\gamma_1}. \end{aligned} \tag{A.2}$$

In this way, we can be sure that the transition will go in the appropriate direction. Of course, we must avoid an entry to the cell with a configuration like $S_1F_{q_1\gamma_0}$, because it will remain as $S_1F_{q_1\gamma_0}$ instead of changing to $S_0F_{q_0\gamma_j}$ as required by equation (A.1).

To avoid this problem, the γ-degree of freedom could satisfy the following rules.

1. When $P > 0$, it does not change its state.

2. When $P < 0$, it suffers a transition from the current state to the state γ_i, determined by the current state q_i. This transition is completed within the length of a cell. In this way, F is always capable of making a transition in the required direction.

Another question that deserves more discussion is the way through which the new state of the cell (x_n, y_m) updates towards the right. One possibility is to have field equations for S that are anisotropic in the spatial coordinates. Another way is to introduce a new field that produces the anisotropy. In any case, the new value of $S(x_n, y_m)$ must update to the right (i.e., to the cells (x_i, y_m) for $i > n$) at a speed faster than that of F.

Last, let us look with more detail at the question about how to avoid infinite propagation velocities for F. As mentioned in the paper, we can introduce another field that generates a shift and compression of the cells (and in fact of the entire system) on the y direction as the beam propagates. In the normal case (i.e., the field machine described in the article) the state of a cell at position (x_n, y_m) represents the state of the symbol at position m in the tape of the ATM at time step n.

In the modified picture with a shifting field, the vertical position of the cells shift continuously with time, in such a way that the position of the symbol at position m in the tape at time step n of the ATM will be represented by the cell (x_n, y_{m+n}). In this way the velocity of F remains finite, because the size of the biggest cell diminishes with time.

If the halting state q_h is reached then a change in the shifting field ensues and the shifting stops. In this case the total shift (or shrink, as we must properly say) remains finite.

References

[1] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, Redwood City, CA, 1979).

[2] L. Blum, M. Shub, and S. Smale, "On a Theory of Computation and Complexity over the Real Numbers: NP-Completeness, Recursive Functions and Universal Machines," *Bulletin of the American Mathematical Society*, **21** (1989) 1–46.

[3] H. T. Siegelmann and E. D. Sontag, "Analog Computation via Neural Networks," *Theoretical Computer Science*, **131** (1994) 331–360.

[4] H. T. Siegelmann, "Computation Beyond the Turing Limit," *Science*, **131** (1994) 545–548.

[5] C. Moore, "Unpredictability and Undecidability in Dynamic-Systems," *Physical Review Letters*, **64** (1990) 2354–2357.

[6] Y. S. Kivshar and B. Luther-Davies, "Dark Optical Solitons: Physics and Applications," *Physics Reports*, **298** (1998) 81–197.

[7] N. N. Akhmediev and A. Ankiewicz, *Solitons, Nonlinear Pulses, and Beams* (Chapman and Hall, London, 1997).

[8] G. J. Chaitin, *The Limits of Mathematics* (Springer-Verlag, New York, NY, 1998).

[9] C. H. Bennett, "On Random and Hard-to-Describe Numbers," *Scientific American*, **241** (5) (1990) 20–34.