WORKING NOTE AC93

Alan Creak
7 December 1994

CELLULAR AUTOMATA WITH A PURPOSE

Much work has been done on cellular automata as entities which are interesting in their own right. Their behaviour has been explored and charted in many studies, and theoretical work has illuminated the results obtained by experiment and observation. More particularly, this work is based on the publications of J.P. Crutchfield, M. Mitchell, and their various colleagues, hereinafter abbreviated to C&M.

The work discussed here is of a rather different sort; it is specifically concerned with cellular automata which compute functions of their initial states. Such automata have *N* stable final states, each of which corresponds to a predicate defined over the initial states, in such a way that an automaton which begins with an initial state which satisfies predicate P_I will evolve into final stable state *I*.

Any cellular automaton with stable final states satisfies the requirement set out in the previous paragraph, but in most cases it is likely that the predicates P_I will be defined only in terms of the automaton. I shall concentrate on the more interesting case of automata in which P_I has some independent significance, such as "the majority of cells in the initial configuration were in state 0". (Whether this case strikes you as more interesting depends on whether you are human, or a cellular automaton.)

I shall discuss one-dimensional automata in which cells may have two states, which I shall call *black* and *white* for convenience. I shall denote them, when convenient, by $\mathbb{B}$ and $\mathbb{W}$, using # for don't-care.

Can only preserve existing conditions.

A cellular automaton is a machine which translates patterns into patterns. The patterns are represented as *configurations* of the cells of the automaton, so we can represent a step in the operation as

$$C^s \rightarrow C^{s+1}$$

C^1 is called the initial configuration, IC; in an automaton which computes, the aim is to evolve towards a final configuration, C^∞, in a way which depends on some property of C^1. Generally, we may choose a set of final configurations, each of which we wish to correspond to some predicate over the initial configurations; and the aim of the exercise is then to find a rule for the automaton which will cause any initial condition to be converted into the final condition which encodes the predicate which the initial condition satisfies. The set of predicates must be complete, in the sense that every C^s satisfies a predicate, and they must be mutually exclusive, so that any C^s satisfies exactly one predicate.

There is then just one predicate which applies to any configuration, and in the ideal automaton this predicate will determine the final configuration. Denote the predicates by x, y, z, etc., the result of applying predicate x to configuration C by P(x, C), and the final configuration for predicate x by C^∞(x).

Then

$$P(x, C^1) \Rightarrow C^1 \rightarrow \dots \rightarrow C^\infty(x)$$

In the first step, $C^1 \rightarrow C^2$. If the behaviour required is to work, we must therefore also have

$$C^2 \rightarrow \dots \rightarrow C^\infty(x)$$

and this can only happen if P(x, C^2). Also, C^∞(x) must be stable – so

$$C^\infty(x) \rightarrow C^\infty(x)$$

implying that

$$P(x, C^\infty(x)).$$

In words : a correct rule for the automaton must always preserve the predicates which describe the configurations, and the final configuration must satisfy its own predicate. In practice, this is likely to look as though an initially confused picture becomes clearer as the overall predicate is developed. I had originally thought of this as the original property being *exaggerated* in some sense, though I'm not sure now that that was a happy choice of word. I was perhaps influenced by C&M's use of a completely white (black) final configuration to denote an initial preponderance of white (black), but I'm not sure that the extreme case is necessary. Is there any reason why a region with a majority of white (black) cells should not be represented by a pattern which is light grey (<𝕎𝕎𝔹>*) (dark grey (<𝔹𝔹𝕎>*)) ?

Exact solutions ? – only if radius is big enough.

What problems *can* the cellular automata solve ? The set of soluble problems must include problems in which it is always possible to know exactly how to set the next state of a cell given the view of set radius from the cell. There may be other soluble problems, but it's harder to see how they might work.

Consider the problem of determining whether or not there are any isolated black cells in the initial configuration. Suppose that the final configuration is all white if there is no isolated black cell, and has some isolated black cells if there are initially isolated black cells. This problem is trivially not soluble by an automaton of radius zero, but it is soluble (in one step !) by an automaton of radius 1, because radius 1 is always sufficient to determine whether or not a cell is black and isolated. The rule is :

{𝕎𝔹𝕎} → 𝔹; {anything else} → 𝕎.

This problem is unusually simple because the answer can be computed directly from the local properties of sufficiently small segments of the automaton, so no information transfer through the automaton is necessary.

The possibility of solution also depends on the representation chosen for the solution. I've shown that the solution must satisfy its own predicate, but that isn't a sufficient condition. Notice that the problem would not (obviously) be soluble in radius 1 if the final configuration for the existence of isolated black cells had been chosen as <𝕎𝔹>*, even though this pattern satisfies its own predicate. To achieve this final configuration, isolated black points must be expanded, following a pattern such as

...𝕎𝕎𝔹𝕎𝕎... → ...𝕎𝔹𝕎𝔹𝕎...

If this is to work, though, there must be a transformation

{𝕎𝕎𝔹} → 𝔹

This transformation is clearly unacceptable, because it would cause a change of the form

...𝕎𝕎𝔹𝔹𝔹... → ...#𝔹###...

which proliferates black points which are not isolated.

It might be possible to achieve the new final configuration with radius 2, when the problem illustrated above could be overcome with transformations of the form

{𝕎𝕎𝕎𝔹𝕎} → 𝔹

in which the isolation of the neighbouring 𝔹 can be explicitly specified. Now, though, there may be some difficulty in grafting together merging domains of 𝔹𝕎𝔹𝕎𝔹𝕎 with different parity; the pattern {𝕎𝔹𝕎𝕎𝔹𝕎} must not be allowed to become {𝔹𝕎𝔹𝔹𝕎𝔹}, as the {𝔹𝔹} in the middle will be eliminated at the next step. Instead, to achieve the required alternating pattern over the whole domain it may be necessary to shift parts of the pattern one place left or right, and that isn't obviously easy. (Which isn't to say that it's impossible, though it must be impossible with C&M's circular odd-parity automata.)

It is clear that, once we address problems which can't be solved within the field of view of a single cell, the requirements become much harder to satisfy. A cell requires information from distant parts in order to determine its correct output; and the information must necessarily take some time to reach the cell, because the only available mechanism is by transmission through intervening cells. If we knew how this could be managed reliably, there wouldn't be a problem, but whatever the mechanism it must be able both to gather information together to produce the final result and know what sort of information to transmit to cells which need it. In addition, cells must be able to pass on transmitted information without interfering with their own computations. I am not suggesting that there must be recognisable mechanisms to implement these functions, but contemplating the requirements does bring out the level of complication in the problem.

COMMENT ON OTHER WORK.

So C&M stuff can't work ?

All C&M's investigations have centred on what they call the $r_c = {}^1/_2$ task, or the majority problem, in which the automaton is supposed to work out whether its initial configuration had more black cells than white. *None of their automata solves this problem correctly*. I conjecture, on the basis of experiments and thinking, but with no proof, that it can't be done, because the correct solution depends on a global property which cannot be encompassed by a cell with a view significantly smaller than the whole configuration of the automaton, and I don't believe that the information transport machinery available can be made sufficiently reliable to carry out the job required.

I don't know just how much that matters, but if you want to investigate the evolution of problem solving it seems odd to work on a problem you know your machine will never be able to solve. If you want to evolve approximate solutions, I suppose it's fair enough, but I want the algorithm that's driving the aeroplane in which I'm riding to be a bit better than an approximate solution that works most of the time.

Therefore useless ? (- but very human.)

Are we really interested in genetic algorithms which do millions of experiments to discover the wrong answer ? In fact, the algorithms do much the same as we do : they solve the easy cases (mostly black, or mostly white), and then guess, or fail to guess, on the cases for which we could really do with mechanical assistance.

Unless you're a molecular biologist interested in the process for its own sake, the point of looking at genetic algorithms is presumably to discover new ways to do things. What I'd like to get out of a study like this is (just possibly) a reliable way to solve the original problem, or (more likely) some insight into how I could go about designing a device which will solve certain sorts of problem.

What C&M end up with is an unreliable way to solve the problem (or perhaps a reliable way to solve some other, unidentified, problem), and possibly some insight into how to evolve wrong answers. It's good fun, and the pictures are fascinating, but I'm far from sure that it's doing much good.

MOVING INFORMATION ABOUT.

Particles or logic ?

If we lose interest in the C&M approach on the grounds that it's never going to work, then what's left ? I've already observed that a non-trivial cellular automaton has to collect information, and the C&M automata certainly do that. If their work is producing anything interesting, then, perhaps it's the *particles* which they abstract from the patterns produced by their automata, and which they regard[1] as "one of the main mechanisms for carrying information over long space-time distances". They also say : "Logical operations on the information they contain are performed when the particles interact".

It seems to me that this view is unhelpful. If the computation is done by the particles, how does the result end up in the domains between the particles ? I suggest that the real computation is done where the particles *don't* interact. A "particle" is the boundary between two regular domains of the diagram. In my view, each domain is a collection of cells, each of which expresses a local predicate Π_i, which may or may not be a local version of one of the global predicates P_I which I defined earlier. This is clearly a plausible interpretation in these "computing" problems, where the meaning of the final domains is predetermined, and the computation proceeds by reactions between local domains which (in successful cases) conclude with the expansion of local versions of the final predicate until they cover the whole automaton..

For example, in C&M's figure 2f (page 9), there are three characteristic domains (after the first few steps), which show up as black, grey, and white. They call these domains B, #, and W respectively, and on page 10 tabulate some of the properties of the "particles" corresponding to the domain boundaries. But what are the "particles" doing ? Why is # eaten up by W ? (Notice that B, W, and # refer to domains; 𝔹, 𝕎, and #, as defined earlier, refere to individual cells.)

I suggest that it's because the various domains are carrying information about their "ancestor" domains. In particular, W (B) is a domain over which it is certain that white (black) predominates, while # is a domain in which there are equal numbers of black and white points. In effect, for each domain there is a predicate which is true for its interior. (I cannot be certain that these interpretations of the domains are correct, but they're plausible, and they conform to the notion that a domain pattern must be an example of its predicate.)

Within a domain, the predicate is simply propagated from generation to generation; once established, it remains true for as long as the domain remains undisturbed. Over a range k well within the domain, we can write

$$B_k \rightarrow B_k;\ \#_k \rightarrow \#_k;\ \text{and } W_k \rightarrow W_k,$$

whichever is appropriate. (I carefully avoid defining the range too carefully; it must contain more than one cell, because the lines aren't exactly reproduced, and the odd-looking area at the boundary between two domains can be quite wide.) But now consider the bottom boundary between # and W in figure 2f. As the pattern develops through generations, the W domain spreads to the right, and we must now define the behaviour by writing

$$\#_{k-1}{}^{s} \;\&\; \#_{k}{}^{s} \;\&\; W_{k+1}{}^{s} \rightarrow \#_{k-1}{}^{s+1} \;\&\; W_{k}{}^{s+1} \;\&\; W_{k+1}{}^{s+1}$$

This is good logic : if there are equal populations of black and white over range k and an excess of whites over range k+1, then there must certainly be an excess of whites over the combined range of k and k+1. This is where the information is propagated, and therefore where the work is done.

Another example is the vertical boundary between B and W towards the top left corner of the same figure. This is a boundary between an area with an excess of blacks and an area with an excess of whites; without knowledge of the magnitudes of the excesses, no conclusion can be drawn at the boundary, so it must simply be preserved.

There is bad logic in C&M's pictures too. An example is the upper boundary of the prominent wedge of # in figure 2f. The logical function executed at this boundary is

$$W_{k-1}{}^{s} \;\&\; W_{k}{}^{s} \;\&\; \#_{k+1}{}^{s} \rightarrow W_{k-1}{}^{s+1} \;\&\; \#_{k}{}^{s+1} \;\&\; \#_{k+1}{}^{s+1}$$

This operation can be read "If an area with an excess of whites adjoins one with equal numbers, we can take a white and a black from the W area and attach them to the # area". That may well be true most of the time, but it is not true if the W area contains only whites, and we have no quantitative information which can decide that question.

(It's interesting that the whole of the large # triangle in figure 2f may in fact be fraudulent. It appears to grow from the W area on its left, as just described, and similarly from the B area on its right, by a process open to similar criticism. The overall result of the computations involving the # area is to eliminate the B triangle, most of which is at the top left corner of the diagram, in favour of the surrounding W – a process which is certainly invalid if my identification of the predicates associated with the domains is accepted. Observe, too, that almost all the computation visible in figures 2b and 2c is similarly fraudulent; perhaps we should be reassured that in the more highly evolved automaton of figure 2f there is at least one valid argument, even though it should never have happened.)

What can be achieved with logic ?

If we start at the end of the process, the logic view is fully justified : a completely black (white) state means, by definition, that black (white) predominated over the whole range at the beginning. Similarly, it is clearly, if trivially, true for individual cells throughout the computation. We have seen that some, at least, of the observed logical processes are valid when plausible meanings are ascribed to the regions concerned. Unfortunately, though, none of these observations guarantees that the interpretation is always correct, so it is of interest to explore a little to probe the boundaries of what can and can't be done with this logic.

It is instructive to contemplate the vertical B-W boundary which we discussed briefly above. This boundary cannot move, because we are short of precise information. We can denote the boundary in terms of the predicates of its domains like this :

{ excess(𝔹) < 0 | excess(𝔹) > 0 }

where excess(𝔹) is the difference between the numbers of cells in states 𝔹 and 𝕎.

With the inequalities, we do not know how to proceed to any other configuration. But suppose now that we have much more precise information :

{ excess(𝔹) = -5 | excess(𝔹) = 3 }

(There is a little problem even in putting forward this supposition; I shall return to it later.) Now we are on much firmer ground; we can immediately see that the whole area can be combined into the single area

{ excess(𝔹) = -2 }.

Unfortunately, though it is clear to us that this transformation is possible, it isn't clear to the cellular automaton, which does not have the advantage of our breadth of view. The automaton must somehow start at the boundary, where it has access to the nature of each of the domains, and distribute the result throughout the combined area. At the end of the operation, the whole area must be filled with a pattern which means excess(𝔹) = -2, so the mechanism must begin by making a small domain with this meaning, then expanding it. The original step must therefore be to form the pattern :

{ excess(𝔹) = -5 | excess(𝔹) = -2 | excess(𝔹) = 3 }

So far, so good. But now what ? Once the width of the central domain exceeds the diameter of a cell's input field, nothing in the system knows what's supposed to be happening. At each of the new boundaries, the same sort of process will be repeated, giving :

{ excess(𝔹) = -5 | excess(𝔹) = -7 | excess(𝔹) = -2 | excess(𝔹) = -1 | excess(𝔹) = 3 }

And so on. This isn't going to work.

Even if it were, there is the other difficulty which I mentioned above. This is of quite a different nature, but is equally – or perhaps even more – conclusive. I have demonstrated that the pattern of a domain must satisfy its own predicate; but what sort of pattern can satisfy, say, excess(𝔹) = 1 ? Given the discreteness of the states, the only way is to have most of the area coded as 𝔹 = 𝕎, with a change of phase somewhere involving two adjacent black cells. And what is that but a particle ?

How could particles do the job ? Some sort if computation is certainly possible in principle. One can imagine a system in which all inequalities are represented by particles moving on a grey (#) background. We need particles of two sorts : one sort (say, black) moving to the left and meaning excess(𝔹) = 1, and another (white) moving to the right and meaning excess(𝔹) = -1. The first task of the automaton is to generate the correct numbers of these particles from the initial state. (I don't think that's trivial, either, but let that pass.) If now things are so arranged that black and white annihilate each other on meeting, the arithmetic will work. But now, of course, we have another problem : how do we know when the arithmetic has finished, and how then do we expand what's left to give the required final configuration ?

It seems likely, then, that patterns can represent inequalities, and perhaps some simple ratios such as 𝔹 = 𝕎, 𝔹 = 2𝕎, etc. I conjecture that the number of such patterns will be limited by the diameter of the cells' input fields, as a cell must know that the pattern is there in order to respond appropriately. But patterns can not sensibly represent differences; it may be that particles can, but there are difficulties in that approach too.

WHAT IS REALLY GOING ON ?

I shall now return to the performance of C&M's system, and offer some suggestions on how it can be described. So far, my only suggestion has been negative : I don't believe that it is solving the majority problem, or that it will ever be able to do so. How else can we look at it ?

We would like to know, because a more appropriate model gives us a better way to evaluate the success, or otherwise, of the method. If we believe that the C&M system is intended to be working out algorithms to determine which colour is in the majority, then we must assess it as a failure; the "algorithms" don't always work. On the other hand, however we describe the phenomenon, it is clear that *something* is developing, and that *something* is improving its performance. If we could identify the something, then we might be able to learn more from the experiments.

Is it science ?

Two features of the behaviour of the C&M system strike one immediately. (I should remark that I write after contemplating the topic for several months, on and off.) They are :

- The system learns by experience. Every change that it makes to its behaviour is based, however loosely or remotely, on the results of experiments.

- The system controls its experience. The experiments which it performs are not chosen at random : they are directed, however loosely or remotely, by the results of previous experiments, using the knowledge gained in the earlier experiments to concentrate on more profitable avenues of investigation.

What sort of activity do we know which can be described in this way ? The obvious first guess – for me, anyway – is science.

Science proceeds by just such a process. The scientist performs an experiment on the universe, and incorporates the result into a growing body of understanding of the nature of things. He can then plan the next experiment to probe some uncertainty in the augmented body of understanding, and so on. The result of each experiment is integrated with the database at each step, and guides future work.

Clearly enough, there is some correspondence between that description and the behaviour of the C&M system. Equally clearly, we cannot reasonably expect the C&M system to operate with the same level (or the same sort) of intelligence as the human scientist, so a precise analogy is not to be expected. Nevertheless, I don't think that science is the most appropriate description of the system's activity, for two reasons : there is no sense in which the system constructs a model of the world; and the criterion of success or failure is associated with a specific goal.

Science is more than the accumulation of a mass of information; it is an attempt to study not only what happens in the universe, but how it happens. The facts that the moon rotates round the earth and apples fall to the ground are interesting, but that both phenomena can be accounted for by the same hypothetical gravitational force is exciting science. The notion of gravitation is a model of the behaviour of the universe which can stimulate further experiment. Notice that whether or not it is "true" is irrelevant, and impossible to verify; it is more important that it should work. In the C&M system, there is no suggestion that decisions are taken on the basis of any notions above the level of phenomena. The notions of logic and particles which I have discussed are from me and from C&M; they have no counterparts in the operation of the system, which contains no machinery for wondering why its universe behaves as it does.

The significance which I ascribe to the system's goal – to solve the majority problem – is related to the ideas put forward in the previous paragraph, and may be more contentious. Science has a goal, but it is more abstract; we can think of it in terms such as "to find out how the universe works". A scientist living in the C&M system's world and aware of the sort of phenomena found in C&M's experiments might wish to probe further into the behaviour of particles or the logic which occurs at boundaries between domains; whether or not this would help to solve the majority problem would be of minor significance. Repeated failures to solve the problem would be of no interest in themselves, but the development of a model of the universe from which statements about the problem could be inferred would be a rewarding, even if incidental, conclusion. (Compare attempts to exceed the speed of light with theories which predict that it's impossible.)

No – it's engineering !

Those considerations give a very strong pointer to an area in which the C&M system feels much more at home. The human occupation devoted to finding better and better ways to achieve stated goals is engineering.

The engineer's goal is to use whatever resources are available to attain specified ends. The resources can be of any form; physical resources such as metals, wood, stone, water, air, and so on are used as needed, and intellectual resources are treated similarly. Modern engineers use the ideas of science as resources because they work, but engineers can function just as professionally without this scientific background. It is very handy to use our knowledge of physics to work out the forces in the members of a bridge before building it, but engineers were successfully building bridges long before we knew about the physics. Even now, engineers will exploit new discoveries using empirical guidelines before proper mathematical analyses of their behaviour are available – consider neural networks and fuzzy systems.

If we adopt this point of view, we can interpret the system's behaviour as an exploration of its universe in search of better tools which it can use to complete its specified task. We can suppose that, as the evolution proceeds, the system is developing an engineer's "understanding" of the behaviour of its universe, and the reproducible appearance of the stages of sophistication becomes quite intriguing. In effect, as the system proceeds through the stages of evolution identified by C&M, it develops better and better ways of addressing the problems which face it, culminating (in their experiments) with the invention of a device for moving information around the automaton. I find it particularly interesting that in the great majority of evolutionary sequences studied by C&M they find the same order of development of tools. I have no idea whether or not that observation can be carried over to human development – it's quite a big step ! – but it does suggest that there is a path of development which in some sense is obvious and comparatively easy.

I cannot resist one final comment in this section. I mentioned above that an engineer uses science as a tool because it works. One could therefore wonder whether C&M's "engineer" might eventually invent science, not as a philosophical pursuit, but simply as a tool to get better engineering done. The immediate answer must surely be "no", as there's no machinery in C&M's system which can do anything like that. Given a more complex system, though, could something of the sort happen ? – which is to say, how big is the step from knowing certain facts about the universe to wondering why they are as they are ? Whence cometh curiosity ?

Who's doing the engineering ?

We find ourselves with a system which is intelligent enough to learn how to use the phenomena of its universe as tools to accomplish set tasks. It is composed of two "hardware" units, the cellular automaton and a learning engine, which happens to be implemented using a genetic algorithm. Where is the intelligence ?

I know that's a silly question, which has been asked many times about artificial systems, but in this case it may not be quite so silly because an immediate consequence of the intelligence, such as it is, is observable : the major intelligent behaviour is learning, and we can try to identify the part that learns by looking for changes in the system.

First, then, is the intelligence in the learning engine ? No : the learning engine learns nothing, because it never changes.

Is the intelligence in the cellular automaton ? No : each cellular automaton only applies what it knows, never originates anything, and dies when it's finished.

The thing that changes as learning proceeds is the structure of the cellular automata which are used. You can think of this, romantically, as a change in the genetic material which determines how the automata are constructed, or, prosaically, as a change in the programme executed by a universal cellular automaton. It is interesting that the romantic view leads to the idea of genetic algorithms, while the prosaic view is invariably (I imagine) used for the implementation.

The intelligent part, then, is a longish list of numbers ? Having eliminated the learning engine and cellular automaton, that's all we have left. I find it a little hard to accept that this view captures what I mean by learning or intelligence; I want some*thing* to learn, not an abstraction. It is true that the limitations of my credulity can't affect the correctness or otherwise of the conclusion, but my understanding of *what I mean* by learning and intelligence can, particularly as I have no reason to believe that what I mean by these terms is much different from what other people mean.

That's because "all we have left" is too little. The numbers cannot learn without the universal cellular automaton, which alone gives them meaning (if you change the automaton, the numbers may give quite different behaviour) and the learning engine, which evaluates the meaning and does something about it. Without these three components, even such learning as is exhibited by the C&M system can't happen. Given the three, I am more ready to believe that the phenomenon is learning, though I'm no more enlightened as to its nature.

Should we regard learning as an "emergent property" of these three components ? Yes – if we believe that "emergent property" is a euphemism for "something we don't understand", which may well be as good a definition as any. No – unless we want to think of power generation as an emergent property of the parts of an internal combustion engine, which, like the parts of the C&M system, are expressly designed to work in just the way they do work. The intelligence of the C&M system isn't a surprise, which makes it even odder that we don't really know what it is. How can C&M design a system to exhibit a specific sort of behaviour without knowing what that behaviour is ?

I think that my answer to that question is that they didn't. They designed a system to exhibit a well defined form of behaviour which resembles in some aspects the consequences of the activity I call learning. (I don't remember now whether they even called it learning, though they certainly speak of discovery, and I think that much the same case could be made out for that word too.) *I* called it learning – and then complained because the phenomena didn't match my notions about learning, which I can't define anyway.

There is certainly a gap, though. When I learn things, in engineering or elsewhere, I don't claim that evolution is doing it. I change, somewhere; the new ideas develop from the old, by extension or replacement, and I think that this impression of growth and refinement is a part of what I mean by learning, which is absent from the C&M machines.

REFERENCES.

1 : R. Das, M. Mitchell, J.P. Crutchfield : *A genetic algorithm discovers particle-based computation in cellular automata*, preprint for the Third Parallel Problem-Solving from Nature Conference, March, 1994.