Two-dimensional Four Color Cellular Automaton: Surface Explorations

Robert H. Barbour*

School of Computing and Information Systems,
Unitec New Zealand,
Carrington Road, Mount Albert, Auckland, New Zealand

An implementation of a two bit Gray-coded temporal logic based cellular automaton created several emergent properties. A program implementing the two-dimensional four color cellular automaton using Ant Farm is demonstrated. New and previously described emergent integer sequences are also identified. Possible applications in spatial modeling, knowledge domain modeling, and mobile robot movement are suggested.

1. Introduction

This paper studies the movement of a simple cellular automaton called the lean quaternary temporal logic (LQTL) ant. In particular, I study the movement of an ant that is searching a "space" using a Gray-code logic to direct it. In this paper a minimal interaction context is modeled and some outcomes reported about the interaction between two entities. In the first section the fundamental premises on which the ant's movement is based are described. In the second section, the two entities used to model a minimal interaction are described and their behavior is detailed. In the third section, observations are reported on the results of running an algorithm that models the behavior of the two entities. In the fourth section, the behavior of the interacting entities is described and further work identified.

The task of identifying the simplest emergent algorithmic spatial search behavior that also included a return-to-base attribute led to my study of cellular automata. No other spatial search strategies are reported here from the literature. Research, over some 40 or more years into cellular automata, is reviewed in Sarkar [1] and Ganguly *et al.* [2]. Cellular automata take many forms but underlying them all is a set of simple rules and the behavioral interactions that result from playing the rules out in a modeling facility, usually a computer. A two-dimensional (2D) cellular automaton can be viewed as a "virtual ant" interacting with a grid of cells. Little communication occurs between adjacent cells.

*Electronic mail address: bbarbour@unitec.ac.nz.

Each cell has a limited rule set that controls its behavior. Repeating the steps described in an algorithm generates cell states sometimes producing complex patterns that are often unexpected (Langton in Wolfram [3]).

2. Underlying logic

The logical model (a Gray-code developed from Frank Gray's 1953 concepts [4]) that underlies the interaction is consistent with modal logic S4S (modal logic S with four successants [5]). Applying that logic, there are four (among many possible categories) statements about the status of the interaction: false, moving towards truth, true, and moving towards false. Taking these four possibilities and their logic and applying them to a 2D four color cellular automaton is described in this section. The choice of colors for the cellular automata cells is arbitrary but makes the differences between the status of the interactions taking place in the cells clearer for the observer. By construction, the status changes follow a defined succession from false (color white), to moving towards true (color green), true (color black), and those moving towards false (color yellow). We avoid anthropomorphizing the behavior of the model, objects, and processes under consideration. To avoid confusion words used in unusual ways will be "quoted" when introduced. The entities referred to in this section are conceptual entities though they have analogs in screen displays as will be illustrated and can be readily exemplified by suitable physical models. The major goal in section 2.1 is to outline a model of a minimal data exchange system for the LQTL ant. The model described is usually known as a 2D four color cellular automaton [3].

2.1 The ant and the grid

An "ant" or cellular automaton, occupies a single cell and has a simple set of behaviors. It interacts with the grid surface that also has simple behavior. In its simplest form, the grid is a 2D array of pixels. The interaction begins with all cells set to a single "status," represented by a color, say white. The simulation of the cellular automaton instantiated in the Ant Farm software [6] has a screen-displayed surface of the grid that is programmed to record an interaction with the second entity, the ant, by changing color as it moves on through a cell.

Each cell may be one of the four colors listed where the color indicates the number of times the ant has passed through the cell. To start the simulation the ant may be positioned anywhere in the grid of cells. The cells begin white and change to green when the ant passes through them the first time. The cells then change to black on the second visit and yellow when the ant moves through a cell on the third occasion. After the fourth occasion each cell begins the color change cycle again,

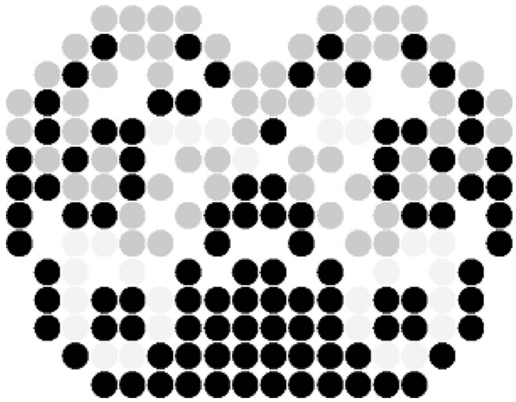

Figure 1. LQTL ant after 5000 iterations showing four color bilateral symmetry.

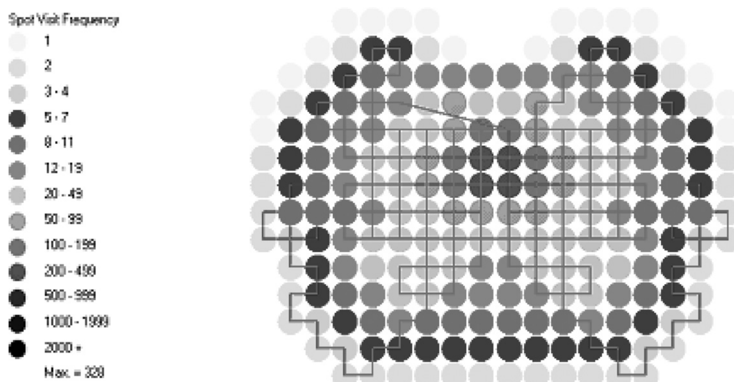

Figure 2. LQTL ant cell visit frequency and most recent path after 5000 iterations.

returning to white. The cell "reports" its grid coordinates to the ant which "collects" the cell data that is written out to a file enabling a subsequent recapitulation or examination of the path of an interaction. In a metaphoric sense the ant can be programmed to "remember" where it has been by having it step back through the path of coordinates recalling previous moves. The grid of cells is programmed to keep a counter reporting the number of times each cell is entered. The cells "provide" a second piece of information to the ant, the current color. To the observer these behavioral details are made visible on screen as illustrated in Figures 1 and 2 after 5000 iterations. The overall pattern in these illustrations is symmetrical about the origin.

The ant has simple behavior; it can turn left (say away from truth), or right 90 degrees (say towards truth), and it can move ahead one cell at a time. If the ant moves into a white or green cell it makes a right-hand 90 degree turn and moves ahead into the next cell. If the ant enters a black or yellow cell it turns 90 degrees to the left and moves ahead into the next cell. The grid surface is also programmed to "maintain" a record of the number of times each cell is visited and retains the cumulative total of cell visits as color shown in a contour map. The software that implements the 2D cellular automaton is called Ant Farm [6]. In the following, Langton's ant is briefly discussed as a forerunner to a more detailed description of emergent behavior of Ant Farm.

Langton's ant is a 2D two color cellular automaton. The demonstration of Langton's ant gave a first indication that there may be interesting emergent behavior with other variants of cellular automaton. Using Ant Farm, configured to replicate Langton's ant moves, the following integer sequences emerged.

Integer sequence A102358 [7] is the finite sequence of 28 iterations at which Langton's ant passes through the origin of the grid:

$$0, 4, 8, 16, 52, 60, 96, 140, 184, 276, 368, 376, 384, 392, \\ 428, 436, 472, 656, 660, 3412, 4144, 4152, 6168, 6764, \\ 8048, 8052, 8056, 8060, 8068.$$

It is puzzling why there should be just 28 of these passages. It is equally puzzling that the integer sequence A102369 [7] should be generated by the 29 successive arrivals of Langton's ant at the origin:

$$4, 4, 8, 36, 8, 36, 44, 44, 92, 92, 8, 8, 8, 8, 36, 8, 36, 184, 4, \\ 2752, 732, 8, 2016, 596, 1284, 4, 4, 4, 8$$

The Ant Farm parameters used for this research are configurable to facilitate investigations into a range of 2D two, three, and four color ants. The speed at which the ant moves and the number of steps it takes are also configurable, an attribute that enables, at slow speeds, identification of emergent patterns and their further close study. The size of the ant is also user selectable enabling very large iterations to be visually examined or detailed examinations of a very few iterations. At high speeds, ant paths can be rapidly searched for patterns at selected step totals. Lastly, the ant may have one of eight compass points designated as the "start direction." The interaction of Langton's ant with the grid has well-known emergent behavior. After around 10,000 steps, Langton's ant creates a so-called superhighway with a regularly repeating pattern of 104 cells. Running Ant Farm with the 2D two color cellular automaton raised the question: What would occur with a 2D four color ant implementing a Gray-code?

With a 2D four color ant, traversing the grid could be taken as a metaphor of learning about exploring a grid space. The learning process would require two visits by the ant. On the first visit the cell would respond by turning from white "not learned" to green, "being learned." On the second visit, the cell would turn black indicating it was "learned." A next visit would report "forgetting" making the cell yellow, and by the next visit the cell status would return to the "unknown" color of white, that is, it would be "forgotten" in relation to other cells.

In the foregoing, a description of Ant Farm was provided and its relationship with research into cellular automata identified. In the next section the relationships between the logic underlying the ant and its behavior is described. Then the patterns in the pixels created by the interacting ant and grid are reported.

2.2 Lean quaternary temporal logic

The constructs of false, becoming true, true, and becoming false can also be thought of as representing temporal extensions to boolean logic. Lean quaternary temporal logic (LQTL) [7] is a terse form of temporal logic created by assigning four descriptors such that false, becoming true, true, and becoming false are represented and become a linear sequence (S4S, mentioned earlier). An analogous set of constructs is used in rotary control system logic where four states are assigned two bit representations 00, 01, 11, and 10 [4]. Movement through the ordered two bit pairs indicates both that change has occurred and more importantly indicates the direction of that change. As a status is changed the direction of the change is recorded by changing a single bit that, in Ant Farm, is reflected in the grid cell color change. In a balanced binary tree two alternatives are open at any point in time, change or no change. Should there be a change then, in the ant cellular automaton described, that change causes the ant to move to the next cell. The changes to cell color as the ant moves on-screen reflect the underlying logic described here and reported as results in the next section.

3. Results

Three patterns of integer sequences created as the LQTL ant moves through the grid are described next.

a. The count of the row possibilities of the four states over successive steps [8].

Here the sequence does not predict what the status will be but rather we look ahead and on the basis of previous status probabilities for prior steps, report the probability of each of the four states on that particular step. Given that there are four possibilities it could be expected that the possible outcomes would be about the same for each. It is apparent in Table 1 that prior states influence future states. In the short run, the sequence shows that the most likely outcome over the first four steps shifts rightwards from becoming true, to true, to becoming false. Note

Status	00	01	11	10
Row 1 count	1	1	0	0
Row 2 count	1	2	1	0
Row 3 count	1	3	3	1
Row 4 Count	2	4	6	4
Row 5 Count	6	6	10	10
Row 6 count	16	12	16	20
Row 7 count	36	28	28	36
Row 8 count	72	64	56	64
Row 9 count	136	136	120	120
Row 10 count	256	272	256	240

Table 1. Moving count of occurrence of status towards equiprobable outcomes.

Status	00	01	11	10
Iteration 1	1	0	0	0
Iteration 2	2	1	0	0
Iteration 3	3	3	1	0
Iteration 4	4	6	4	1
Iteration 5	6	10	10	5
Iteration 6	12	16	20	15
Iteration 7	28	28	36	35
Iteration 8	64	56	64	71
Iteration 9	136	120	120	135
Iteration 10	272	256	240	255
Iteration 11	528	528	496	495
Iteration 12	1024	1056	1024	991
Iteration 13	2016	2080	2080	2015
Iteration 14	4032	4096	4160	4095
Iteration 15	8128	8128	8256	8255

Table 2. Predominance of status values by iteration.

that as the steps continue, the difference in magnitude between the outcomes rapidly moves to roughly similar counts, suggesting confirmation of the intuitive expectation of a 1-in-4 probability for each state.

b. The count of the column possibilities of the four states over successive steps [9]. The cumulative column frequencies are shown in Table 2 from the first to fifteenth iteration of LQTL logic. Note that with an initial false in the zeroth iteration Murphy's Law holds in all but six iterations (the 6th, 7th, 14th, and 15th). Note also that these

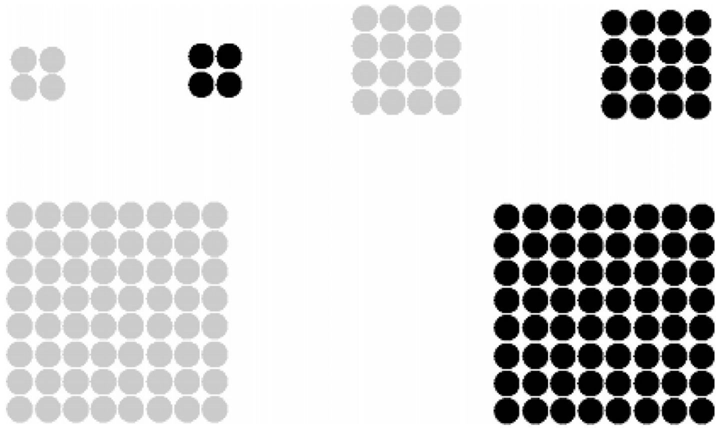

Figure 3. LQTL ant completed squares on iterations 4, 8, 32, 64, 416, and 832.

are the only iterations where the reports favor "true" or "learned" in the first 15 iterations.

c. The unique squares created by the ant completing specific successive iterations in the grid [10]. The integer sequence A094867 is the number of unique iterations required to generate six emergent regular squares using LQTL logic. Subsequent emergent behavior at greater than 832 iterations becomes symmetrically random (Figure 3).

Iterating the sequence creates a remarkable emergent outcome: regular single color homogenous status squares from the origin and created by only two of the four status reports, green (01) and black (11). Note that yellow (10) and white (00) squares are never formed even though the iterations have been observed into the tens of thousands. Furthermore, there do not appear to be any further Barbour–Chapman squares beyond these six iterations: 4 (green), 8 (black), 32 (green), 64 (black), 416 (green), and 832 (black). The squares are completed when the ant "returns-to-base."

The movement patterns of the LQTL ant are shown in Figures 3 and 4. Figure 3 shows the cell visit status while Figure 4 shows the frequency of visits overlaid by the path taken by the ant during the previous cell visits. Notice the early signs of symmetry loss in visit patterns in the top right of the last square. The following section draws the data and conjectures presented in this paper together pointing to future work that the results suggest.

4. Discussion

The behavior of cellular automaton (2D, two and four color) described in this paper has not been reported in the literature. So far as has been

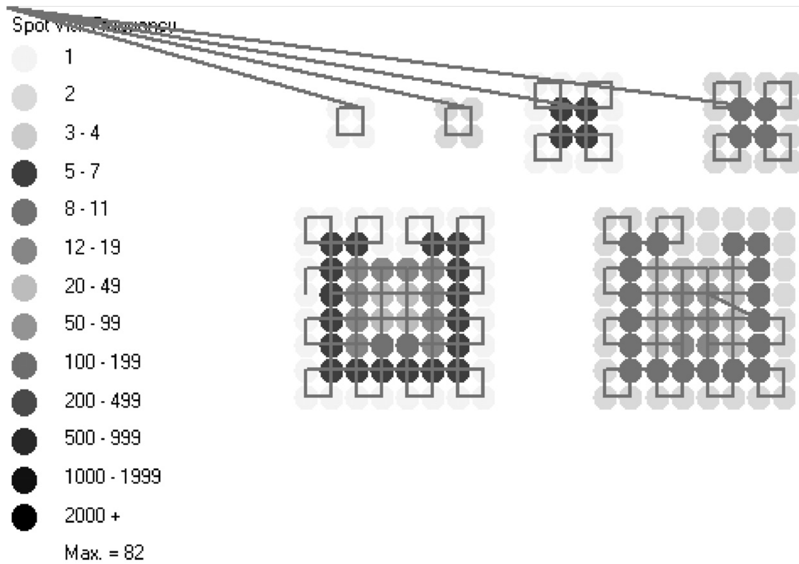

Figure 4. LQTL ant cell visit frequency and recent path during iterations 4, 8, 32, 64, 416, and 832.

determined there have been no reported attempts as yet to study the relationship between cellular automata and Gray-code though Wolfram raises the question of possible relationships [3]. The behavior of the ant described here has been observed over thousands of iterations over a 24-month period of time. At iterations beyond 832 the ant creates no further regularities in any of its subsequent patterns. Bilateral symmetry is preserved but no further single color squares appear (Figure 1). Figure 2 shows the frequency of cell visits overlaid by the path traversed through the grid by the automaton. The relationships modeled in the LQTL ant are readily observable, easily replicated, and demonstrate intriguing regularities accessible to beginning programmers.

The Ant Farm illustrates the emergence of symmetry in apparently random behavior. Langton's ant does not demonstrate emergent behavior until well into the iterating sequences whereas the LQTL ant shows emergent regularities in behavior very early in the interaction. The LQTL cellular automaton clearly demonstrates both early unpredictability and irreducibility.

Given that a cellular automaton can be viewed as a "virtual robot," the ant behavior described here is predictive of robotic behavior under conditions of a featureless plain. Taken at its most simple level the "learn about a space" and "return-to-base" tasks require the automaton to search and map the extent of a featureless grid. It does this, as required, though perhaps there are more efficient return-to-base space-searching algorithms. For those robots operating at very large or very small scales two fundamental questions are addressed by the LQTL ant simulation. How shall the ant move and how long should it continue to move? The simulation shows that the movement is well defined up to 832 iterations. The simulation also shows that the ant will return-to-base six times during those 832 iterations when all the cells have the same status. That means six checks can be made over the 832 movements on the cell's status. Failure to return means that the ant has encountered some impediment. Here then is a very simple algorithm that when implemented gives a number of control or check points for a virtual robot that could be applied to an actual space-searching robot. Using this algorithm could very well provide a key control element for autonomous nanobots as to both movement and life span. The fact of the emergent regularities has a mathematical significance as yet undetermined. Some outstanding questions include: Why only six regular squares? Why no yellow or white squares? It could be speculated that once visited it would be unlikely that all visited cells would return to white (unlearned/forgotten) at the same time. Would such speculations be making much out of mere coincidence? Probably, but that does not account for the visited sets of green and black squares (learned) having repeating single color squares or alternating but increasing size.

5. Conclusion

The significance of the cellular automaton simulations and the experimental results described here are that they provide a visual representation of spatial search behavior under a very simple set of rules. The patterns open to question the long held view that, all other things being equal, four equally possible outcomes have equal probability in the short and long run. The four possible outcomes have no other association than that they are ordered in a linear sequence. Yet the emergent patterns clearly support the folk knowledge implied in Murphy's Law. Slightly revised, Murphy's Law could read: anything that can go wrong, will, in the short run. In the long run, the outcome could go equally the way of any of the possible outcomes. Relationships of underlying lean quaternary temporal logic (LQTL) with other four-valued forms of logic are under active investigation.

Acknowledgments

The Ant Farm and associated tools were coded by Jason Chapman to the requirements of the author. The Ant Farm application is available, on request, from the author.

References

[1] P. Sarkar, "A Brief History of Cellular Automata," *ACM Computing Surveys*, **32**(1) (2002) 80–107.

[2] N. Ganguly, *et al.*, "A Survey of Cellular Automata," *Technical Report*, Centre for High Performance Computing (Dresden University of Technology, December, 2003).

[3] S. Wolfram, *A New Kind of Science* (Wolfram Media, Inc., Champaign, IL, 2002).

[4] F. Gray, "Pulse Code Communication," U.S. Patent No. 2,632,058, March 17, 1953.

[5] U. Endriss, "Modal Logic of Ordered Trees," unpublished Ph.D. Thesis, King's College, London, 2003.

[6] J. Chapman, personal communication, 2004.

[7] R. H. Barbour, "LQTL," in *Handbook of the First World Congress and School on Universal Logic, UNILOG 2005*, edited by J-V. Beziau and A. Costa-Leite, Montreux, Switzerland, March 26–April 3, 2005; www.uni-log.org.

[8] R. H. Barbour and L. D. Painter, A094266, *Online Encyclopaedia of Integer Sequences*, (2004); www.research.att.com/projects/OEIS?Anum=A094266.

[9] R. H. Barbour, A099423, *Online Encyclopaedia of Integer Sequences*, (2004); www.research.att.com/projects/OEIS?Anum=A099423.

[10] R. H. Barbour and J. Chapman, A094867, *Online Encyclopaedia of Integer Sequences*, (2004); www.research.att.com/projects/OEIS?Anum=A094867.