Physica 19D (1986) 423–432
North-Holland, Amsterdam

SOLITON-LIKE BEHAVIOR IN AUTOMATA†

James K. PARK‡ and Kenneth STEIGLITZ
Dept. of Computer Science, Princeton University, Princeton, NJ 08544, USA

and

William P. THURSTON
Dept. of Mathematics, Princeton University, Princeton, NJ 08544, USA

Received 13 May 1985
Revised 25 October 1985

We propose a new kind of automaton that uses newly computed site values as soon as they are available. We call them *Filter Automata* (FA); they are analogous to Infinite Impulse Response (IIR) digital filters, whereas the usual Cellular Automata (CA) correspond to Finite Impulse Response (FIR) digital filters. It is shown that as a class the FA's are equivalent to CA's, in the sense that the same array of space-generation values can be produced; they must be generated in a different order, however.

A particular class of irreversible, totalistic FA's are described that support a profusion of persistent structures that move at different speeds, and these particle-like patterns collide in nondestructive ways. They often pass through one another with nothing more than a phase jump, much like the solitons that arise in the solution of certain nonlinear differential equations.

Histograms of speed, displacement, and period are given for neighborhood radii from 2 to 6 and particles with generators up to 16 bits wide. We then present statistics, for neighborhood radii 2 to 9, which show that collisions which preserve the identity of particles are very common.

1. Introduction

Cellular automata have attracted attention recently as non-numerical models for nonlinear physical phenomena [1]. Vichniac [2] points out that they "exhibit behaviors and illustrate concepts that are unmistakably physical...", and he goes on to mention "relaxation to chaos through period doublings," "a conspicuous arrow of time in reversible microscopic dynamics," "causality and light-cone," and others. The purpose of this paper is to describe a new kind of automaton that supports soliton-like structures in a strikingly clear way.

Scott *et al.* [3] propose as a working definition that "a soliton $\phi_s(x-ut)$ is a solitary wave solution of a wave equation which asymptotically preserves its shape and velocity upon collision with other solitary waves." The simplest examples are provided by solutions to the dispersionless linear wave equation. What is remarkable about solitons is that they can be supported by nonlinear equations with dispersion.

The notion of a solitary wave can be carried over in a natural way to the context of automata (either CA or FA) as follows: The term *solitary wave* or *particle* in an automaton will be taken to mean a periodic pattern of non-zero cell values that propagates with fixed finite velocity. A collision between two particles will be said to be a

†This work was supported in part by NSF Grant ECS-8307955, U. S. Army Research–Durham Grant DAAG29-82-K-0095, DARPA Contract N00014-82-K-0549, and ONR Grant N00014-83-K-0275.

‡Now with the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

soliton collision if the particles retain their identities after the collision. (For an example see Fig. 3.)

The particular class of filter automata described here, which we call *parity-rule* FA's, support thousands of particles of relatively small size, are irreversible, and totalistic – that is, they belong to the simple class of automata that depend for their next state only on the number of 1's in the argument field of the next-state function (see below). Furthermore, as we will see in what follows, soliton collisions are quite common, occurring for some next-rule radii and ranges of particle widths 99% of the time. In contrast, in the totalistic one-dimensional CA's studied extensively by Wolfram [5] particles are relatively rare and non-destructive collisions extremely rare. There are also two-dimensional CA's that support particles – for example the gliders in the game of Life [7], or the billiard-ball models in the reversible CA's described by Margolus [4]. But the ease with which particles are supported by parity-rule FA's, and their propensity for passing through one another, appear to be unknown in the study of CA's.

2. Filter automata

We will restrict ourselves here to one-dimensional automata with k-valued site values a_i^t, where the subscript i refers to the space variable ($-\infty \leq i \leq +\infty$), and the superscript t refers to time ($0 \leq t \leq +\infty$). In the usual CA [5], the evolution of the automation is determined by a fixed rule F of the form

$$a_i^{t+1} = F\left(a_{i-r}^t, a_{i-r+1}^t, \ldots, a_i^t, \ldots, a_{i+r}^t\right), \tag{1}$$

with

$$F(0,0,\ldots,0) = 0.$$

The next value of site i is a function of the previous values in a neighborhood of size $2r+1$ that extends from $i-r$ to $i+r$. Given initial states at all the sites, which we assume run from $-\infty$ to $+\infty$, repeated application of the rule F determines the time evolution of the automaton.

In an FA, the next-state rule is of the form

$$a_i^{t+1} = F\left(a_{i-r}^{t+1}, a_{i-r+1}^{t+1}, \ldots, a_{i-1}^{t+1}, a_i^t, \ldots, a_{i+r}^t\right). \tag{2}$$

Now the next state is computed using the newly updated values $a_{i-r}^{t+1}, a_{i-r+1}^{t+1}, \ldots, a_{i-1}^{t+1}$, instead of $a_{i-r}^t, a_{i-r+1}^t, \ldots, a_{i-1}^t$. This is precisely analogous to the operation of an IIR digital filter, whereas a CA corresponds to an FIR digital filter (see, [6], for example).

Although we allow the sites in an FA to extend from $-\infty$ to $+\infty$, we must assume that to the left, anyway, there are only a finite number of sites containing non-zero values. This will then give us an unambiguous way to compute the evolution of the FA, using a left-to-right scan. We always start with an initial configuration that has only a finite number of non-zero site values.

Following Wolfram's terminology [5], when the next-state function F depends only on the sum

$$S(i) = \sum_{j=-r}^{r} a_{i+j} \tag{3}$$

we say an automation is *totalistic*. This class, although small and easy to specify, appears to exhibit all the interesting kinds of behavior found in general automata, and the particular class of FA's described here will be totalistic.

We will focus attention on the class of filter automata with binary-valued sites ($k=2$) defined by the following next-state rule. If $S(i)$ is the number of 1's in the $i-r$ to $i+r$ window at time t, then the new value of site i is

$$a_i^{t+1} = \begin{cases} 1, & S \text{ even but not } 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

These we will call the *parity-rule* filter automata, and we will think of them as parameterized by the single integer $r = 2, 3, \ldots,$ the *radius*.

3. Some examples

Fig. 1 shows a typical particle, one that occurs in the $r=3$ parity-rule FA. The first line corresponds to generation 0 – it indicates the initial values assigned to the array of sites (a black square indicates a site with value 1, and the absence of such a square indicates a site with value 0). Subsequent lines correspond to subsequent generations. It is apparent from this figure that a particle has a well-defined *period* (the number of generations needed for the bit pattern to repeat), and *displacement* (the number of sites moved during a period, with plus measured to the left). In this case the period is 3 and the displacement 1.

We will refer to the succession of states passed through by a particle as its *orbit*. A convenient way to identify a particle uniquely is to view each orbital state as a binary number, and to take the smallest binary number in its orbit as the *canonical code* or *generator* of the particle.

It is not hard to see that the position of the rightmost 1 of a particle can never move right. For this to happen, we must be in the situation where the values of $a_i^t, \ldots, a_{i+r}^t$ are all 0, and there are an even nonzero number of 1's among the values $a_{i-r}^{t+1}, \ldots, a_{i-1}^{t+1}$. As the window slides right, this situation must be repeated, and so an infinite number of 1's would be generated, a contradiction. In fact it has been proved [11] that such situations can never be reached from an initial condition with a finite number of 1's; that is, that the parity-rule FA's are *stable* in this sense.

From the previous observation, we know that particles are either stationary or move left. It is also not hard to see that the maximum speed of a particle is $r-1$, and this is realized by the particle consisting of $r+1$ consecutive 1's, which has period 1.

As Hirota and Suzuki [10] describe, one characteristic of solitons is that "A wave packet at any given position dissolves into many solitons each of which travels at its own velocity." Fig. 2 shows a typical evolution from a disordered state for the $r=3$ parity-rule FA. Exactly the same kind of

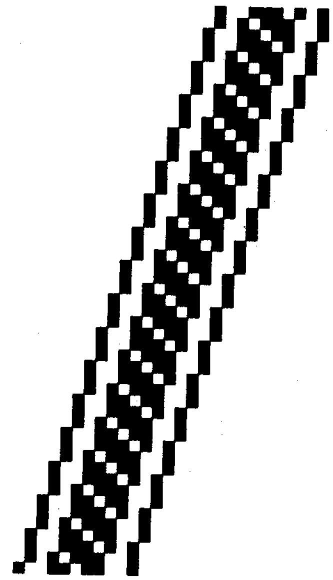

Fig. 1. A typical particle supported by a parity-rule filter automaton, illustrating period and displacement. This example is for $r=3$, has canonical code 629, displacement 1, and period 3. The code sequence in its orbit is 629, 697, 1241.

dissolution into several particles with different velocities can be observed.

Fig. 3 illustrates pairwise collisions that we call *soliton* – those in which the identity of both particles is preserved, while fig. 4 shows examples of non-soliton collisions. Extensive empirical evidence suggests that in a soliton collision the fast particle cannot be shifted to the right, and the slow particle cannot be shifted to the left. That is, the fast particle may only be pushed forward, and the slow only retarded. The latest collision in fig. 4 results in 2 particles moving in parallel and is particularly interesting because it shows that particle collisions are not always reversible. By *reversible* here, we mean that the picture rotated 180° would be a valid evolution of this or some other automaton. If we turn this picture upside down, it becomes clear that spontaneous splitting would be required for reversibility.

We next want to present some statistics for pairwise particle collisions, but first we need to study the maximum possible number of different

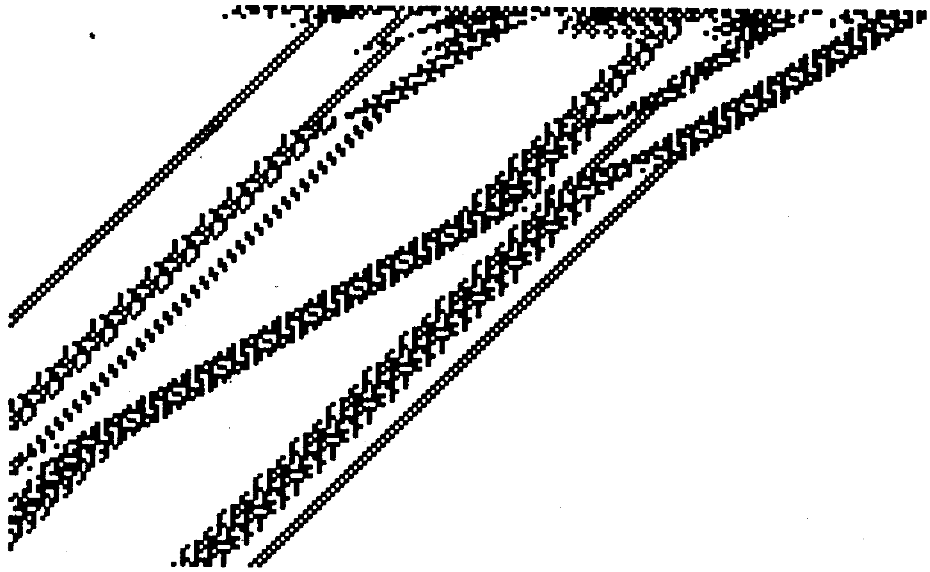

Fig. 2. Typical evolution from a disordered state, for the $r = 3$ parity-rule filter automaton, showing dissolution into several particles with different speeds.

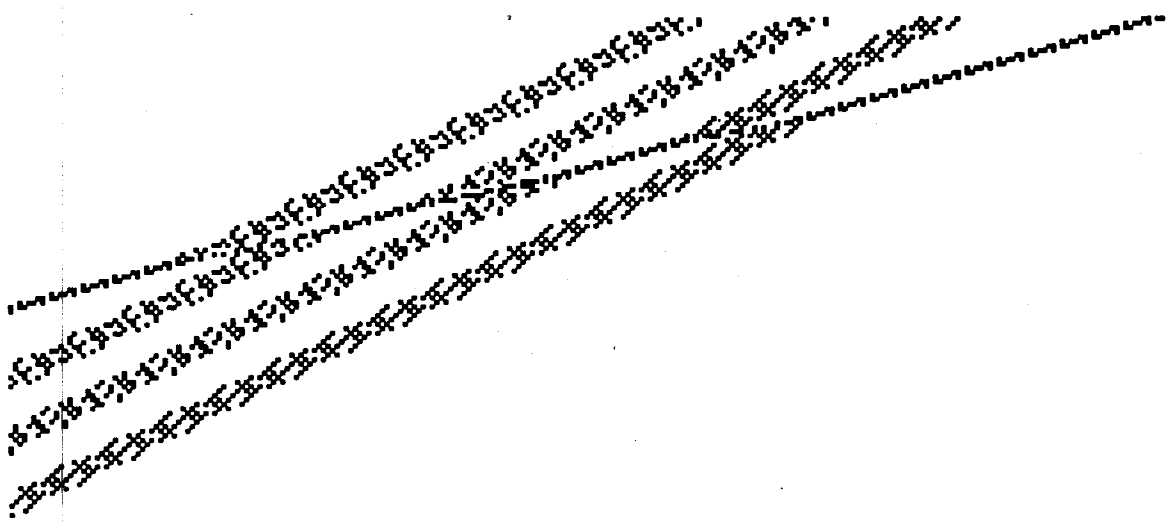

Fig. 3. Some typical soliton collisions, for the $r = 5$ parity-rule filter automaton. The initial particle canonical codes and displacement/periods are, from left to right: 145 (12/6), 201 (12/6), 273 (12/6), and 27 (7/2).

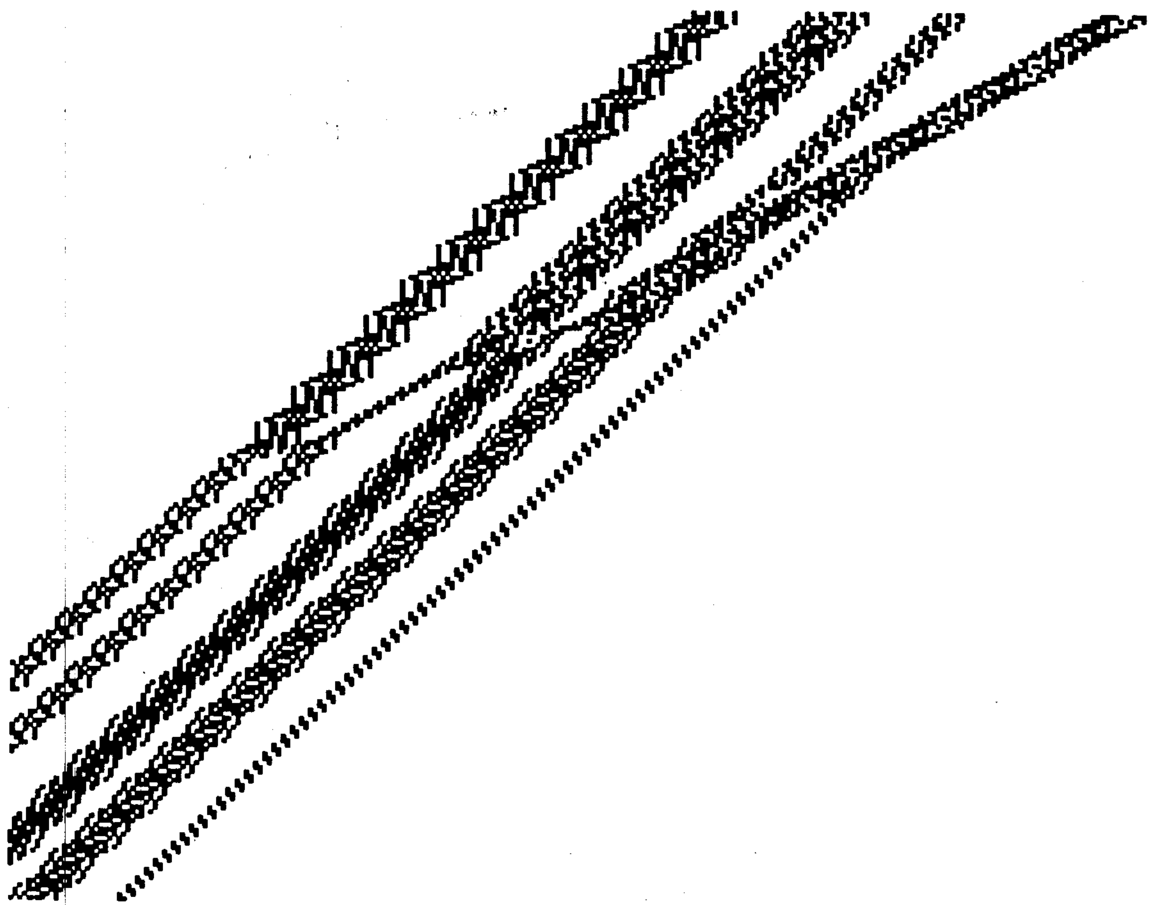

Fig. 4. Some typical collisions in which the identities of both particles are changed, for the $r=3$ parity-rule filter automaton. The initial particle canonical codes and displacement/periods are, from left to right: 601 (8/8), 9451 (10/10), 43 (6/6), and 967 (20/12).

ways two particles can collide, given their periods and displacements.

4. The determinant of a particle pair

Given any two particles that move at different speeds, it is clear that we can arrange a collision between them by choosing an initial configuration with the faster particle to the right of the slower. If the two particles start close enough together, it may happen that they interact in a way that is impossible when they start far apart. In such cases we say the collision is *improper*; otherwise we say it is *proper*. We will restrict our attention to proper collisions, because we will always allow an initial spacing adequate for typical interactions.

We will say that two collisions are the *same* if the bit patterns of their history can be put in concordance by shifts in space and time, and *different* otherwise. We can now calculate a strict limit on the number of different proper collisions possible between two particles.

Theorem 1. Let the two particles have displacements d_1, d_2, periods p_1, p_2, and speeds $d_1/p_1 < d_2/p_2$, so that particle 2 hits 1. Let $q = \mathrm{lcm}(p_1, p_2)$, and the difference in speeds be $\Delta s = d_2/p_2 - d_1/p_1$. Then the number of different proper collisions is no larger than

$$\mathrm{DET} = p_1 p_2 \cdot \Delta s = p_1 d_2 - p_2 d_1.$$

Proof. After q generations, the same relative configuration of bits in the particles' orbits repeats, so we need only cycle through $q \cdot \Delta s$ initial separations to obtain all possible proper collisions.

Case 1: p_1, p_2 relatively prime. In this case those $q \cdot \Delta s = p_1 p_2 \cdot \Delta s$ situations include all combinations of orbital states.

Case 2: p_1, p_2 not relatively prime. Then in $q \cdot \Delta s$ situations we have covered only $q/(p_1 p_2)$ of the total possible. Therefore there are at most $q p_1 p_2 \cdot \Delta s/q = p_1 p_2 \cdot \Delta s$ possible situations, as before.

We call DET the *determinant* of the collision, because it is precisely the 2×2 determinant with rows $p_1 p_2$ and $d_1 d_2$.

5. Particle and collision statistics

Dictionaries of particles were compiled in the following simple way. All bit patterns with width up to 16 were used as starting configurations of the parity-rule FA's, and the automata played forward in time long enough for particles to be generated and separated. The resulting bit strings were than analyzed to catalog the canonical codes, speeds, displacements, and periods of the particles so generated. Table I shows histograms of the results, sorted by speed for each value of r, for all distinct displacement-period pairs, and for all canonical codes of width ≤ 16.

There is a somewhat hazy distinction between two particles with the same speed traveling in parallel, and one particle. For our purposes we insist that a single particle not have a gap of $2r$ or more consecutive 0's in its canonical code.

The number of particle generators increases sharply with r, growing from only 8 for $r = 2$ to 13109 out of a possible 32768 – the number of odd integers up to 2^{16} – for $r = 6$. Another striking fact is the tendency for certain displacement/period pairs to be preferred. For example, for $r = 6$ the two pairs $d/p = 44/12$ and $56/14$ account for 65% of the total.

These particle dictionaries were used to study the question of how often a 2-particle collision is *soliton*; that is, how often the two particles pass through one another without change in their identities. Given a set of particles, all possible pairwise collisions were sampled uniformly as follows: a random pair (a, b) was chosen, with b faster than a and all such pairs equally likely. Then the fast particle was played a random number g generations forward in time, where g was chosen uniformly between 0 and the period of the fast particle less 1; that is, the fast particle was put in a random orbital state. The slow particle was then placed in its canonical orbital state with its right end a random number x spaces to the left of the left end of the fast particle, where x was chosen uniformly between k and $k + \mathrm{DET} - 1$, k being large enough to ensure a proper collision. Finally, the result of the ensuing collision was weighted by its corresponding DET to make the sample uniform over all possible collisions, rather than over all possible pairs of particles. This was done for 2000 collisions, for various values of radius r and for particle dictionaries with code-widths up to 10, 14, and 16. The results are shown in table II.

The general trend is that the estimated probability of soliton collisions increases with r for fixed code-widths, and decreases with code-width, although there are exceptions. What is perhaps most striking is the high level of the observed frequencies, reaching 99% for code-widths up to 10, and $r = 8$ and 9. It was also the case that mutual annihilation was never observed; every collision resulted in at least one particle.

6. Quasi-equivalence of CA's and FA's

A natural question is whether FA's are essentially different from CA's, or whether any FA can be simulated in some sense by a CA. In this

Table I

Speed, displacement, period, and frequency of occurrence for all particles with canonical code width ≤ 16, radius 2 to 6.

Speed	Disp.	Per.	Freq.
Radius = 2, no. of pars. = 8			
0.000	0	1	1
0.500	1	2	1
0.500	2	4	1
0.500	3	6	1
0.500	8	16	1
1.000	1	1	3
Radius = 3, no. of pars. = 198			
0.333	1	3	1
0.500	1	2	1
0.500	4	8	4
1.000	1	1	2
1.000	2	2	1
1.000	4	4	4
1.000	5	5	1
1.000	6	6	3
1.000	8	8	11
1.000	10	10	41
1.000	16	16	8
1.333	8	6	2
1.333	16	12	63
1.400	7	5	3
1.400	14	10	17
1.500	3	2	1
1.500	6	4	1
1.500	12	8	4
1.571	22	14	16
1.667	5	3	4
1.667	10	6	3
1.667	20	12	3
1.750	14	8	1
2.000	2	1	3
Radius = 4, no. of pars. = 682			
0.667	2	3	2
1.500	3	2	4
1.500	6	4	10
1.500	9	6	6
1.500	12	8	44
1.500	15	10	2
2.000	2	1	1
2.000	4	2	1
2.000	10	5	14
2.000	20	10	219
2.125	17	8	19
2.333	7	3	18
2.333	14	6	19
2.333	28	12	216
Radius = 4, no. of pars. = 682			
2.333	35	15	2
2.500	5	2	1
2.500	25	10	14
2.571	18	7	8
2.571	36	14	68
2.750	11	4	5
2.750	22	8	4
2.750	44	16	4
3.000	3	1	1
Radius = 5, no. of pars. = 6534			
1.000	1	1	1
1.000	3	3	3
2.000	2	1	3
2.000	4	2	6
2.000	8	4	20
2.000	10	5	2
2.000	12	6	10
2.000	16	8	110
2.000	20	10	4
2.600	13	5	41
2.600	26	10	926
2.750	11	4	3
2.750	22	8	46
3.000	3	1	2
3.000	6	2	1
3.000	9	3	51
3.000	12	4	4
3.000	18	6	69
3.000	36	12	2251
3.000	45	15	12
3.200	32	10	71
3.286	23	7	53
3.286	46	14	1929
3.500	7	2	2
3.500	14	4	35
3.500	21	6	3
3.500	28	8	32
3.500	35	10	1
3.500	42	12	37
3.500	56	16	693
3.500	70	20	4
3.667	33	9	7
3.667	66	18	96
3.800	19	5	2
3.800	38	10	1
3.800	76	20	2
4.000	4	1	1
Radius = 6, no. of pars. = 13109			
1.333	4	3	4
2.500	5	2	3
2.500	10	4	19
2.500	15	6	15
2.500	20	8	195
2.500	25	10	8
3.200	16	5	30
3.200	32	10	1635
3.375	27	8	95
3.667	11	3	5
3.667	22	6	35
3.667	44	12	4157
3.900	39	10	213
4.000	28	7	21
4.000	56	14	4353
4.250	17	4	5
4.250	34	8	7
4.250	51	12	231
4.250	68	16	1742
4.444	40	9	1
4.444	80	18	227
4.500	9	2	1
4.500	63	14	102
4.600	23	5	3
4.833	29	6	1
5.000	5	1	1

Table II
Frequency of collisions that are soliton, in *per cent*. Based on samples of 2000 collisions.

Radius	Width ≤ 10	Width ≤ 14	Width ≤ 16
2	38.74	38.74	38.74
3	24.63	11.02	9.55
4	65.63	30.50	35.38
5	73.97	45.13	45.20
6	80.35	80.76	52.67
7	80.62	84.51	
8	99.78	83.63	
9	99.42	78.16	

section we will show that the latter is true; in particular, we will show that the space-time array generated by any particular FA can also be generated by some CA, and vice-versa.

First consider the constraints imposed on the order in which site values in FA and CA space-time arrays can be generated. Any site value a_i^t in an FA space-time array is determined by $a_{i-r}^t, \ldots, a_{i-1}^t, a_i^{t-1}, \ldots, a_{i+r}^{t-1}$. These site values are in turn determined by the values of other sites, but all sites that could possibly affect a_i^t lie within the region $\{a_{i+j}^{t-k}\colon k \geq 0,\ j \leq kr\}$ (this is the shaded region in fig. 5). For the space-time array of a CA, the corresponding region of sites whose values can affect a site value a_i^t is depicted in fig. 6 – it is just the region $\{a_{i+j}^{t-k}\colon k \geq 0,\ -kr \leq j \leq kr\}$.

It is clear from figs. 5 and 6 that there exists no 1-to-1 mapping from site values in an FA space-time array to site values in a CA space-time array, with the property that the orientation of the space axis is preserved. Changing the value of some site to the left of a_i^t (but in the same generation) in an FA space-time array can affect this site value, but it cannot in a CA space-time array. Thus, if we wish to simulate an FA with a CA, we must somehow change the orientation of the space axis. This motivates a "tilting" mapping from the FA space-time array to the CA space-time array, a mapping that rotates the region in fig 5 so that it looks more like the region of fig. 6.

Suppose then that we are given an FA, with parameters r and k, and we want to generate the

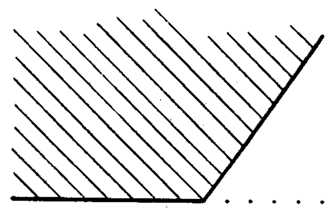

Fig. 5. The region that can affect a point in an FA.

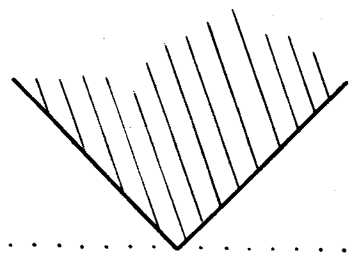

Fig. 6. The region that can affect a point in a CA.

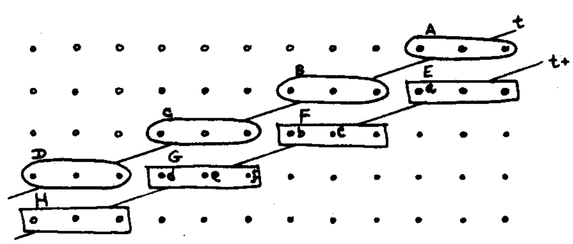

Fig. 7. Illustration of the simulation of an FA by a CA, for the case $r = 2$. The space axis of the CA is tilted, and each cell of the CA is a conglomerate of $(r+1)$ cells of the FA. To compute the value of CA cell G, we compute in turn the value of FA cell a, b, c, d, e and f.

same array of space-time values using a CA, using parameters r' and k'. The mapping we will use is illustrated in fig. 7 for the case $r = 2$. First, groups of three consecutive cell values in a particular generation of the FA array are coalesced to form one cell in the corresponding CA, so that the CA has $k' = 2^{r+1}$. Next, we take the space axis (constant-time axis) of the CA to be tilted so that cells in the same CA generation correspond to cells along a line rotated counterclockwise from horizontal in the FA space-time array.

Using capital letters to represent the cells in the CA, and lower case for the FA, we next show how the value of the cell G in generation $t+1$ can be computed from the values in the neighboring cells A, B, C, D in generation t. First, find the value of the FA cell a from the FA components in A and B. Next find the values of cells b and c from the value of a and the components of B and C. Finally, find the values of FA cells d, e, and f from the values of b, c and the components of C and D. This then gives the value of the CA cell G. We can summarize this in the following

Theorem 2. Every space-time array that is generated by an r, k – FA can also be generated by a CA with $r' = r$ and $k' = 2^{r+1}$, provided that we allow the values to be computed in a different order. The FA cell values are also coded in groups of $r+1$ alphabet symbols.

Next consider the problem of simulating a CA with an FA. A glance at fig. 8 shows that this is easy, provided that we allow a slippage to the left: just compute the value at site $x+r$ at the point x, and choose the FA rule to depend only on the values in the preceding generation. Each generation will therefore be shifted r cells to the left with respect to the CA array, but will otherwise be identical; in this case the time axis is tilted. We summarize this as

Theorem 3. Every space-time array that is generated by a r, k' – CA can also be generated with an FA with $r = 2r'$ and $k = k'$, provided that we allow each successive row in the FA to be displaced r units to the left with respect to the corresponding row in the CA.

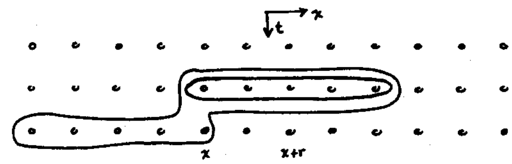

Fig. 8. Illustration of the simulation of a CA by an FA. The value at cell $x+r$ is computed at point x in the FA, so that the time axis of the FA is shifted to the left r cells every generation.

7. Discussion

Many questions about the class of filter automata remain unexplored. Some rules other than the parity rule appear to support particles the way the parity rule does. For example, the following variation of the parity rule seems to support particles and soliton collisions, and the particles are on the average slower:

$$a_i^{t+1} = \begin{cases} 1, & S \text{ even but not 0 or 2}, \\ 0, & \text{otherwise}. \end{cases} \tag{5}$$

Other rules are unstable in the sense described in section 3. While the question of stability in linear Infinite Impulse Response digital filters is settled by the criterion of characteristic values being inside the unit circle, an analogous general technique for FA's is unknown.

One of the motivations for studying cellular automata in general is to gain insight into the nonlinear phenomena that occur in the solution of differential equations, and in the physical systems they are used to model. We have seen that certain simple one-dimensional automata give rise to solitary waves that very often pass through one another non-destructively. Whether such automata capture the mechanism of soliton generation in differential equations is uncertain, but we feel worthy of further study.

Another reason for studying CA's is the possibility of embedding useful computation within such regular and simple structures. One-dimensional cellular automata can be implemented in VLSI in a highly pipelined and efficient way [8], resulting in what amounts to a cellular automaton machine with an almost unlimited degree of parallelism. Even if the embedding of a useful computation is very inefficient, it may still be more than compensated for in certain applications by the parallelism and efficiency of the VLSI implementation. There exist complex CA's that simulate a universal Turing machine, but we still do not know how to construct a simple one-dimensional CA that does useful computation. It appears, however, that it will help to have particles that can pass through one another, because that will make possible communication between different elements of the automaton.

Carter [9] describes the transmission of information in molecules using physically supported solitons. The general notion of processing information in simple, homogeneous media via solitons opens the way for speculation about such computation at the level of the molecule or the biological cell.

Acknowledgements

We thank Irfan Kamal and Arthur Watson, who helped in many ways, especially with the development and testing of programs. Doug West and Stephen Wolfram took the time to offer useful suggestions.

References

[1] D. Farmer, T. Toffoli and S. Wolfram, Cellular Automata (North-Holland, Amsterdam, 1984).

[2] G.Y. Vichniac, "Simulating Physics with Cellular Automata," pp. 96–116 in [1].

[3] A.C. Scott, F.Y.F. Chu, D.W. McLaughlin, "The Soliton: A New Concept in Applied Science," Proc. IEEE, vol. 61, no. 10, pp. 1443–1483, October 1973.

[4] N. Margolus, "Physics-Like Models of Computation," pp. 81–95 in [1].

[5] S. Wolfram, "Universality and Complexity in Cellular Automata," pp. 1–35 in [1].

[6] A.V. Oppenheim, R.W. Schafer, Digital Signal Processing (Prentice-Hall, Englewood Cliffs, NJ, 1975).

[7] E.R. Berlekamp, J.H. Conway and R.K. Guy, Winning Ways for your Mathematical Plays, Vol. 2: Games in Particular, (Chapter 25, "What is Life?") (Academic Press, New York, NY, 1982).

[8] K. Steiglitz and R.R. Morita, "A Multi-Processor Cellular Automaton Chip," Proc. 1985 IEEE Int. Conf. on Acoustics, Speech, and Signal processing, Tampa, Florida, March 1985.

[9] F.L. Carter, "The Molecular Device Computer: Point of Departure for Large Scale Cellular Automata," pp. 175–194 in [1].

[10] R. Hirota and K. Suzuki, "Theoretical and Experimental Studies of Lattice Solitons on Nonlinear Lumped Networks," Proc. IEEE, vol. 61, no. 10, pp. 1483–1491, October 1973.

[11] C.H. Goldberg, manuscript in preparation.