

## Linear reversible second-order cellular automata and their first-order matrix equivalents

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2004 J. Phys. A: Math. Gen. 37 10791

(<http://iopscience.iop.org/0305-4470/37/45/006>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 171.66.16.65

The article was downloaded on 02/06/2010 at 19:43

Please note that [terms and conditions apply](#).

# Linear reversible second-order cellular automata and their first-order matrix equivalents

A J Macfarlane

Centre for Mathematical Sciences, D.A.M.T.P., Wilberforce Road, Cambridge CB3 0WA, UK

E-mail: a.j.macfarlane@damp.cam.ac.uk

Received 27 May 2004

Published 28 October 2004

Online at [stacks.iop.org/JPhysA/37/10791](http://stacks.iop.org/JPhysA/37/10791)

doi:10.1088/0305-4470/37/45/006

## Abstract

Linear or one-dimensional reversible second-order cellular automata, exemplified by three cases named as RCA1–3, are introduced. Displays of their evolution in discrete time steps,  $t=0, 1, 2, \dots$ , from their simplest initial states and on the basis of updating rules in modulo 2 arithmetic, are presented. In these, shaded and unshaded squares denote cells whose cell variables are equal to one and zero respectively. This paper is devoted to finding general formulas for, and explicit numerical evaluations of, the *weights*  $N(t)$  of the states or configurations of RCA1–3, i.e. the total number of shaded cells in  $t$ th line of their displays. This is achieved by means of the replacement of RCA1–3 by the equivalent linear first-order matrix automata MCA1–3, for which the cell variables are  $2 \times 2$  matrices, instead of just numbers ( $\in \mathbb{Z}_2$ ) as for RCA1–3. MCA1–3 are tractable because it has been possible to generalize to them the heavy duty methods already well-developed for ordinary first-order cellular automata like those of Wolfram's Rules 90 and 150. While the automata MCA1–3 are thought to be of genuine interest in their own right, with untapped further mathematical potential, their treatment has been applied here to expediting derivation of a large body of general and explicit results for  $N(t)$  for RCA1–3. Amongst explicit results obtained are formulas also for each of RCA1–3 for the total weight of the configurations of the first  $2^M$  times,  $M = 0, 1, 2, \dots$

PACS numbers: 45.30.+s, 02.10.-v, 02.30.Lt

## 1. Introduction

Since the publication of the pioneering paper [1], there has been sustained interest in the subject of cellular automata, continuing right up to the present time. Evidence for this can be provided by considering a list of books devoted to the topic, which includes [2–9], as well as others

devoted to applications such as fluid flow. From [2] one gains access to many of the important early papers, and from [9] one finds a truly vast body of examples, with extensive discussion and impressive illustration, while in [8] one finds comprehensive referencing of the field in general.

One strong motivation for the continued study of cellular automata stems from the desire to create simulations of characteristic features of real-world processes; see chapters 7–9 of [9]. In the case of physical processes a general property of importance is reversibility or time-reversal invariance. Chapter 9 of [9] gives a broad analysis of this concept in relation to cellular automata; see also chapter 8 of [8] and chapter 14 of [3]. Further many physical processes are governed by second-order partial differential equations within theories that are invariant under time reversal. This applies in discussions of particle behaviour, e.g. scattering and particle production, which has been treated using reversible second-order cellular automata in [10]. Also, [11] considers the use of reversible cellular automata in the modelling of processes in quantum mechanics and quantum field theory. There is much work also on reversible cellular automata which pursues aims in statistical mechanics, or else purely structural aims; we mention a few examples [12–17]. For references to fluid dynamics, see [8]. The work presented in this paper is different in outlook and aims from many of the works cited: we wish here to study the behaviour of second-order reversible cellular automata with a view to exhibiting, and getting a good measure of algebraic control over, some of the interesting mathematical structures which they contain.

Our interest here is confined to one-dimensional or linear cellular automata, life in one dimension, as opposed to Conway's game of Life, see e.g. [18, 19]. The foundations (and more) of the studies of linear cellular automata were laid down already in [1], and we use the language of this paper to refer to the automata of Rule 90 and of Rule 150. We discuss these first-order automata in preparation for the central purpose of this paper, which is to analyse reversible second-order cellular automata. It is well known that second-order differential equations can often with advantage be replaced by equivalent first-order matrix equations. Here we wish to exploit this fact in the context of reversible second-order cellular automata, replacing these by equivalent first-order matrix cellular automata for which the cell variables instead of just being numbers are matrices. The value of this replacement lies in the fact that we can generalize the heavy duty methods [20, 21] already developed for first-order cellular automata like those of Wolfram's Rules 90 and 150 where cell variables take on values 1 and 0, to cases like MCA1–3 for which the cell variables take matrices as values. We think matrix-valued cellular automata are of intrinsic interest and have potential for development. But they have been used here as tools to allow us to carry out our mathematical analysis of RCA1–3, and later perhaps more general studies.

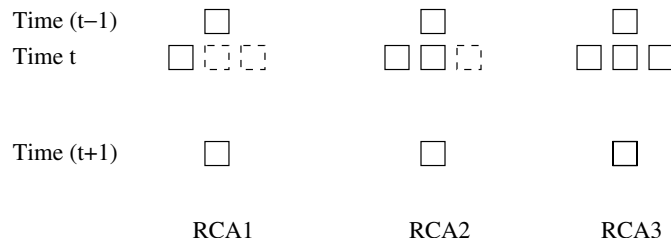
This paper approaches the subject of reversible second-order cellular automata by reference to several examples of (modestly) increasing complexity. We suppose that the cells  $C_n$ ,  $n \in \mathbb{Z}$  or  $\mathbb{Z}_+$ , of the linear arrays of our one-dimensional automata have (numerical) variables  $x_n(t)$  attached to them at each instant of discrete time  $t \in \mathbb{Z}_+$ . The simultaneous updating of all cells  $C_n$  of the arrays at each time  $t$  is specified by the following rules, applied using modulo 2 arithmetic:

$$x_n(t+1) = x_{n-1}(t) + x_n(t-1) \pmod{2}, \quad (1)$$

$$x_n(t+1) = x_{n-1}(t) + x_n(t) + x_n(t-1) \pmod{2}, \quad (2)$$

$$x_n(t+1) = x_{n-1}(t) + x_n(t) + x_{n+1}(t) + x_n(t-1) \pmod{2}, \quad (3)$$

respectively. We refer to the corresponding automata by the names RCA1–3. That these rules are indeed reversible can be seen by using modulo 2 arithmetic to make  $x_n(t-1)$  the subject



**Figure 1.** Diagrams of the rules for the updating of the automata RCA1–3. The dashed boxes in the laterally spread neighbourhoods of cells  $C_{n-1}, C_n, C_{n+1}$  at time  $t$  do not correspond to entries used in the actual rules RCA1–3, but are present to allow easy visual comparison of which cell variables at times  $(t - 1)$  and  $t$  influence cell variables at time  $(t + 1)$ .

of (1)–(3), writing, for RCA1, for example,

$$x_n(t - 1) = x_{n-1}(t) + x_n(t + 1) \pmod 2. \tag{4}$$

Figure 1 gives a diagrammatic representation of (1)–(3).

For RCA1–3 the single seed initial state configuration is specified by

$$x_n(t = -1) = 0 \quad \text{for all } n \in \mathbb{Z},$$

$$\text{and } x_0(t = 0) = 1, \quad x_n(t = 0) = 0 \quad \text{for all } n \neq 0 \in \mathbb{Z}. \tag{5}$$

In fact for RCA1 and RCA2, only  $n \in \mathbb{Z}_+$  is relevant since evolution from (5) can never enter  $n < 0$ .

The evolution of RCA1–3 from the initial states of (5) is displayed in figures 2–4, respectively.

Some explanation of the displays may be helpful. It should be enough to address figure 2 for RCA1. The numbers at the left give the times  $t$  corresponding to the configuration of RCA1 given by the  $t$ th row for  $0 \leq t \leq 31$  with the binary representation  $t_B$  of  $t$  shown alongside for later use. If at any  $t$  the cell variable  $x_n(t), n = 0, 1, 2, \dots$ , takes the value 1 then a shaded square sits at the  $n$ th cell position; if  $x_n(t) = 0$  then the cell position is left blank. The evolution of RCA1 in  $t$  as  $t$  increases discretely in unit steps proceeds down the display in a manner governed by (1). To see exactly how this works consider the times  $t = 3, 4$ . Figure 2 tells us that at  $t = 3$  the only non-zero cell variable is  $x_3(t = 3) = 1$ , while at  $t = 4$  we have only got  $x_{0,2,4}(t = 4) = 1$ . We can now construct the  $t = 5$  line by referring either to (1) or else to the left column of figure 1 for  $t = 4$ . The examples

$$x_1(t = 5) = x_0(t = 4) + x_1(t = 3) = 1 + 0 = 1$$

$$x_3(t = 5) = x_2(t = 4) + x_3(t = 3) = 1 + 1 = 0 \pmod 2$$

$$x_5(t = 5) = x_4(t = 4) + x_5(t = 3) = 1 + 0 = 1$$

tell us why there are shaded boxes at positions 1 and 5 but not at 3 in the  $t = 5$  line, and so on. To get the  $(t + 1)$  line from its two predecessors such steps are done simultaneously for all cells  $n = 0, 1, 2, \dots$ . To initiate the process we set out from the single seed state (5) with the  $t = -1$  line blank and  $x_n(t = 0) = \delta_{n0}$ .

Finally we define the *weight*  $N(t)$  of the configuration of RCA1 at each time  $t$  to be the total number of shaded boxes in the  $t$ th line of figure 2, or the total number of cell variables  $x_n(t)$  that take the value one at time  $t$ . Figure 2 records the  $N(t)$  in its right-hand side column. Figure 2 is obtained from a simple C-program. Similar remarks apply to RCA1–3.



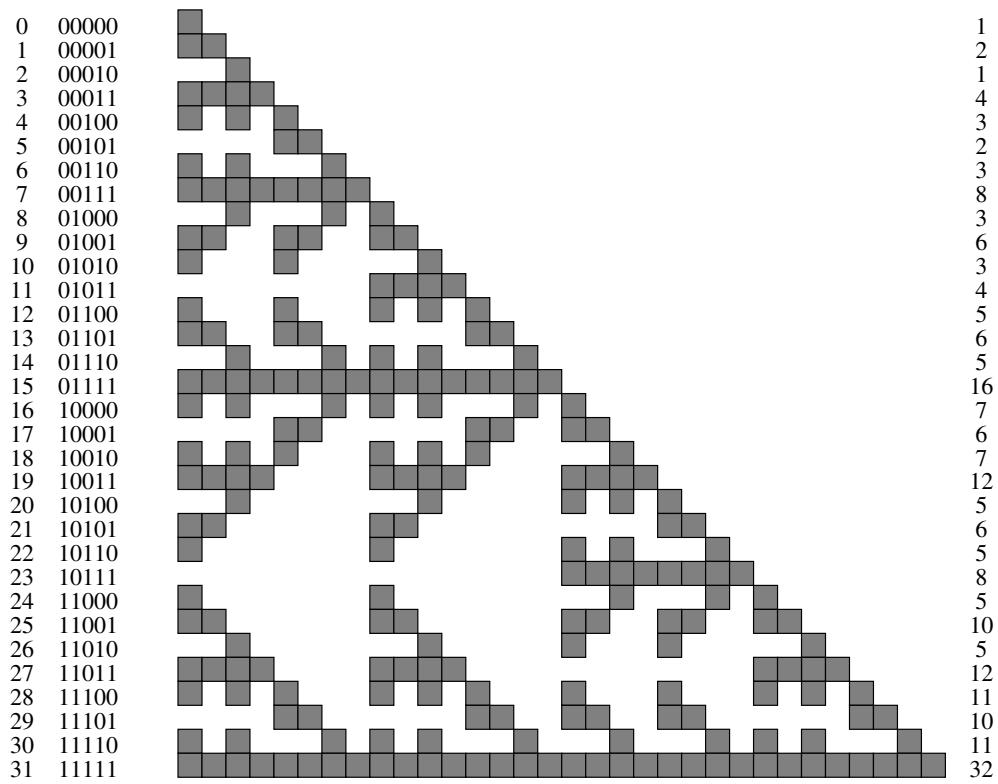


Figure 3. The reversible cellular automaton RCA2 of (2).

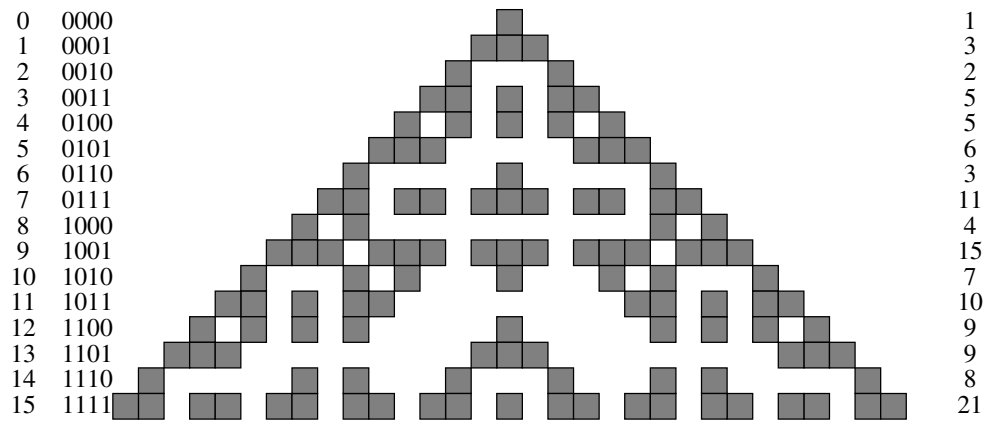


Figure 4. The reversible cellular automaton RCA3 of (3).

$2t$  and  $(2t + 1)$ . In the material we need to present on Rule 150, it is enough to deal with 2-blocks, i.e. blocks of two adjacent cells, whose variables take on values in  $Z_2$ . We describe the relevant material in section 2.2, below, because understanding of it is an essential prerequisite to understanding the generalisation needed to conquer MCA3, for which the cell variables are ones which take values in  $M_{2,2}(Z_2)$ , the set of  $2 \times 2$  matrices whose elements lie in  $Z_2$ .

Once algebraic control of MCA1–3 has been attained, the derivation of general expressions for  $N(t)$  for RCA1–3, and the numerical evaluation of these, can be tackled with good prospects of success.

Our results for the evolution from the single seed initial states of (5) of RCA1–3 include the following. For each of them, we establish a general expression for the weight  $N(t)$  of their configurations at time  $t$ . This involves, in each case, transition matrices  $A_0$  and  $A_1$  which generate algebraically the transition from line  $t$  to lines  $2t$  and  $(2t + 1)$ . (The relevance of these matrices to the binary representation  $t_B$  of the time  $t$ , and hence to our treatment of  $N(t)$  for all cases studied here, is explained in section 2.2 for Rule 150.) For RCA1–3, the transition matrices are respectively  $2 \times 2$ ,  $3 \times 3$  and  $8 \times 8$ .

For RCA1 a systematic approach to the evaluation of  $N(t)$  is available, and it leads to an algorithm that resembles Wick's theorem in quantum field theory for writing down the numerical value of  $N(t)$  directly for any  $t$ . For RCA2 we still have a systematic approach to the evaluation of  $N(t)$ , but have not found an analogue of the algorithm. For RCA3, we quote, with derivation, results for  $N(t)$  for times  $t$  with the binary representations  $t_B = 1^m, 1^m 0^n$ . While we found no real obstacle in pushing our evaluations further, we do not present any more results. The results in question become rather quickly more complicated, although always suggestive of the underlying structure that we are yet to uncover. The quest for a systematic approach to the evaluation of  $N(t)$  for RCA3 however is an ongoing project.

There is another tractable quantity of some interest that can be defined in terms of the  $N(t)$  and evaluated for RCA1–3, namely the total weight of their first  $2^M$  rows:

$$V(M) = \sum_{t=0}^{2^M-1} N(t). \quad (6)$$

For RCA1 and RCA2 we deduce

$$V(M) = \frac{1}{2}(3^M + 1), \quad (7)$$

$$V(M) = 2^{M-1}(F_{M+2} + 1), \quad (8)$$

where  $F_n$  defines the  $n$ th Fibonacci number. For RCA3,  $V(M)$  is given in section 5 below by (71).

The material of this paper is organized as follows. Section 2 reviews for later use some results for Rules 90 and 150. For Rule 90 these are simple and obvious, but give some pointers towards our treatments of MCA1 and MCA2. Rule 150 requires application of some of the  $r$ -block methods of [20, 21], which, for  $r = 2$  and evolution from single seed, we describe in detail just as far as it is needed in preparation for our work on MCA3. It is worth pointing out that the general expression (19) below for  $N(t)$  for Rule 150 has an analogue of similar structure for each of MCA1–3, and indeed for any first-order cellular automata for which the underlying arithmetic is that of  $Z_2$ . Work on MCA1–3 and its application to RCA1–3 occupies sections 3–5. Section 6 offers a few additional remarks.

An appendix deals with certain sequences of numbers: families  $\chi_n$  and  $\phi_n$  defined initially for  $n = 0, 1, 2, \dots$ . These enter our studies because it so happens that many of our explicit results for  $N(t)$  take on a neat and natural appearance in terms of them. The  $\chi_n$  are already well known in the context of Rule 150 [1], but the  $\phi_n$  arose by inspection of the way best to organize computer data as it emerged for RCA3. The  $\chi_n$  satisfy a difference equation from which all its properties stem. Since the  $\phi_n$  were observed to have a close relationship to the  $\chi_n$ , the difference equation they obey and their properties follow too.

## 2. The first-order cellular automata of Rules 90 and 150

### 2.1. Rule 90

Rule 90 is defined by

$$x_n(t + 1) = x_{n-1}(t) + x_{n+1}(t) \pmod 2, \tag{9}$$

and we consider only its evolution from the single seed initial state

$$x_0(t = 0) = 1, \quad x_n(t = 0) = 0 \quad \text{for all } n \neq 0. \tag{10}$$

This gives rise to the well-known picture in figure 5 of its evolution in time. The columns on the right of figure 2 show the weight  $N(t)$  of the configuration for each  $t, 0 \leq t \leq 15$ , with the binary representation  $t_B$  of  $t$  shown alongside, the weight being equal to the number of shaded squares of the configuration. We note some easily seen results. If  $t_B$  contains  $B(t)$  ones, then  $N(t) = 2^{B(t)}$ . Also  $V(M)$  as defined by (6) is given by  $V(M) = 3^M$ .

Needless to say far more can be (and has been) done in the context of Rule 90, see [20, 21].

We note, for later use, since use of it is overkill for Rule 90, that, given knowledge of the  $t$ th line of figure 5, we can write down the lines for times  $2t$  and  $(2t + 1)$  directly: each cell shaded at time  $t$  gives rise to one shaded cell at  $2t$  and two shaded cells at  $(2t + 1)$ . We generalize this reasoning later for RCA1 and RCA2 to useful effect.

### 2.2. Rule 150

Rule 150 is defined by

$$x_n(t + 1) = x_{n-1}(t) + x_n(t) + x_{n+1}(t) \pmod 2, \tag{11}$$

and we consider only its evolution from the single seed initial state (10). This gives rise to the picture figure 6 of its evolution for increasing  $t$  values. The columns at the right of the display again provide relevant data.

A nice general expression for  $N(t)$  for Rule 150 is given without indication of its origin in [1]. Suppose that the binary representation of  $t_B$  of  $t$  contains strings of ones of lengths  $a_\alpha, \alpha = 1, 2, \dots, p$ , separated by strings of zeros whose lengths are irrelevant. Then

$$N(t) = \prod_{\alpha=1}^p \chi_{a_\alpha}. \tag{12}$$

Because the quantities  $\chi_r$  occurring here enter also in several places in our subsequent work, a discussion is given, in the appendix, of their properties. This provides a table of their values. However, the values

$$\chi_1 = 1, \quad \chi_2 = 3, \quad \chi_3 = 5, \quad \chi_4 = 11,$$

enable it to be checked that (12) agrees with all the data in figure 6. For example, for  $t = 423 = t_B = 110100111$  or  $1^2 010^2 1^3$ , we have  $a_1 = 2, a_2 = 1, a_3 = 3$ , so that  $N(t) = 3 \cdot 1 \cdot 5 = 15$ , correctly. Neither the length of the separating strings of zeros in  $t_B$  nor the order of the  $\chi$  factors in (2.3) matters.

To derive a general closed formula for  $N(t)$ , which implies also (12) is a procedure of three stages in which

- (a) one obtains an intuitive view of how the configuration of Rule 150 at time  $t$  determines the configurations at each of the times  $2t$  and  $(2t + 1)$ ,



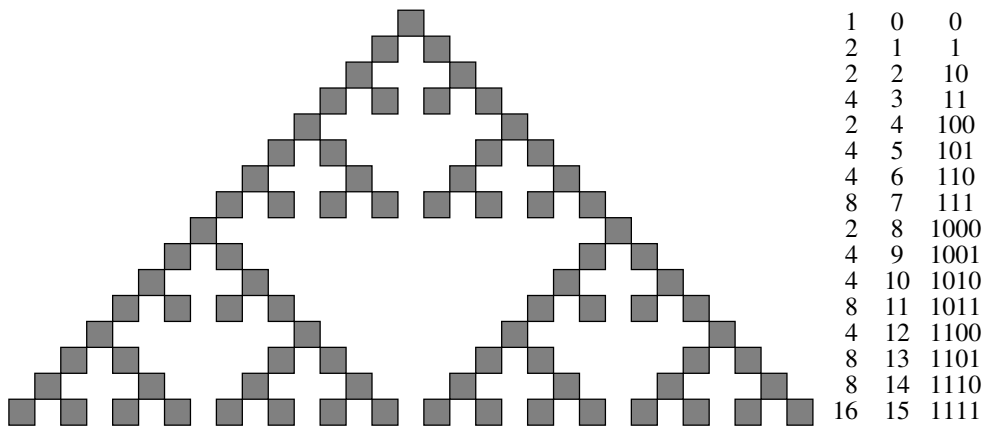


Figure 5. Evolution from single seed of the automaton of Rule 150.

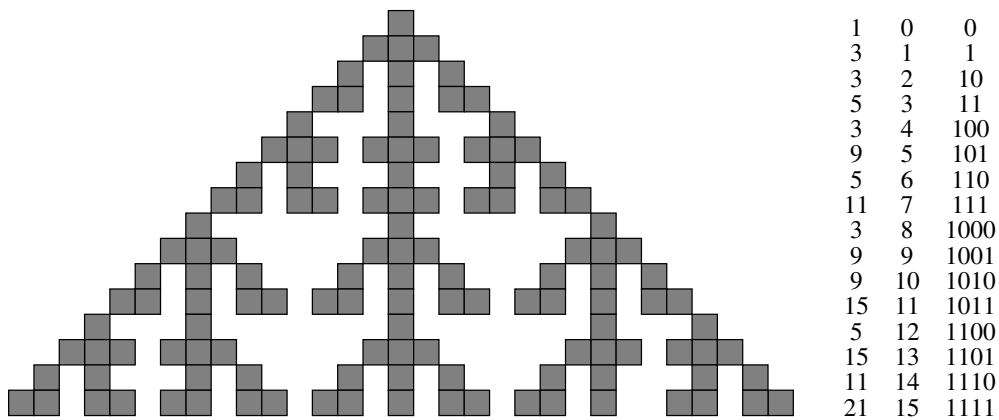


Figure 6. Evolution from single seed of the automaton of Rule 150.

- (b) one finds transition matrices  $A_0$  and  $A_1$  which describe this in algebraic terms, and  
 (c) one uses these matrices and the binary representation  $t_B$  of the time  $t$  to reach the required formula.

A similar procedure is followed for the other cellular automata problems of interest here. A careful exposition of this is given for Rule 150, because it is a prerequisite to understand the discussion in section 5.

By inspection of figure 6, we infer the rule that tell us how, given the line of time  $t$ , we write down the lines at times  $2t$  and  $(2t + 1)$ , for any  $t$ . The *underlying* rules are

$$\begin{aligned} 0 &\longrightarrow 000, & 1 &\longrightarrow 010 & \text{for } t &\longrightarrow 2t, \\ 0 &\longrightarrow 000, & 1 &\longrightarrow 111 & \text{for } t &\longrightarrow (2t + 1). \end{aligned}$$

However, appreciation of this is a subtle matter, because of the problem of overlapping blocks, described and comprehensively resolved by [20, 21]. To explain the present, simplest possible, context, let us look at a portion of time  $t$  line  $\dots xy \dots$ , where the dots indicate cell variables that do not enter our present discussion. This contains the 2-block  $xy$ , i.e. a block of two

consecutive cells  $xy$ . There is an unambiguous rule for finding the unique 3-blocks at each of the times  $2t$  and  $(2t + 1)$  that arise from the 2-block  $xy$ . For  $t \rightarrow 2t$

$$(x \rightarrow 0x0) \quad \text{and} \quad (y \rightarrow 0y0) \quad \text{imply} \quad (\dots xy \dots) \rightarrow (\dots x0y \dots),$$

and for  $t \rightarrow (2t + 1)$

$$(x \rightarrow xxx) \quad \text{and} \quad (y \rightarrow yyy) \quad \text{imply} \quad (\dots xy \dots) \rightarrow (\dots xzy \dots),$$

where  $z = x + y \pmod 2$ . The middle entry of the latter 3-block is all that shows the way the independent effects of  $x$  and  $y$  overlap here at the times  $2t$  and  $(2t + 1)$ ; the result for the former is trivial as  $0 + 0 = 0 \pmod 2$ . In details, for the allowed 2-blocks 00, 10, 01, 11, we have

$$\begin{aligned} (00) &\rightarrow (000) & \text{and} & & (000) & \text{at the respective times } 2t \text{ and } (2t + 1), \\ (10) &\rightarrow (100) & \text{and} & & (110) & \text{at the respective times } 2t \text{ and } (2t + 1), \\ (01) &\rightarrow (001) & \text{and} & & (011) & \text{at the respective times } 2t \text{ and } (2t + 1), \\ (11) &\rightarrow (101) & \text{and} & & (101) & \text{at the respective times } 2t \text{ and } (2t + 1). \end{aligned} \tag{13}$$

Next consider how a complete line at time  $t$  gives rise to the complete lines at times  $2t$  and  $(2t + 1)$ . A single example should clarify this. The line (1101011) of figure 6 for time  $t = 3$  defines a sequence

$$(01), (11), (10), (01), (10), (01), (11), (10)$$

of eight 2-blocks, with (13) thereby determining a corresponding sequence of 3-blocks at time  $t = 7$

$$(011), (101), (110), (011), (110), (011), (101), (110), (011).$$

These can be seen in the  $t = 7$  line of figure 6: (110110111011011).

We can now pass on to counting problems, in which (00) blocks play no role and (01) and (10) enter on the same footing. Further, viewing a given line as a sequence of two overlapping 2-blocks, as in the above example, represents a potential double counting that will need to be compensated for later. We can see from equation (13) that each 2-block of type (01) or (01) present at time  $t$  gives rise at time  $(2t + 1)$  to one 2-block of type (01) or (01) and one of type (11), while each 2-block of type (11) present at time  $t$  gives rise at time  $(2t + 1)$  to two 2-blocks of type (01) or (01). Thus if there are  $a_1$  2-blocks of types (01) or (10) and  $a_2$  of type (11) present at time  $t$  then at time  $(2t + 1)$  there will be  $b_1$  and  $b_2$  of the respective types so that we have

$$b = A_1 a, \quad \text{where} \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}, \quad a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

We have here defined the transition matrix  $A_1$  for transferring the counting of 2-blocks from time  $t$  to time  $(2t + 1)$ . The counting of unit entries or shaded boxes in figure 6 at time  $t$  now gives the weight  $N(t)$  at time  $t$  as

$$N(t) = \frac{1}{2} c^T a = \frac{1}{2} (1 \ 2) \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{1}{2} a_1 + a_2, \tag{14}$$

where the counting vector  $c$  is given by  $c^T = (1 \ 2)$ , since the 2-blocks (01) or type (10) contribute one to the count and (11) two. Also

$$N(2t + 1) = \frac{1}{2} c^T A_1 a = \frac{3}{2} a_1 + a_2. \tag{15}$$

**Table 1.** Values of  $N(t)$  for  $0 \leq t \leq 7$ .

$t$	$t_B$	$v(t)$	$N(t)$
0	0	$u$	1
1	1	$A_1 u$	3
2	10	$A_0 A_1 u$	3
3	11	$A_1^2 u$	5
4	100	$A_0^2 A_1 u$	3
5	101	$A_1 A_0 A_1 u$	9
6	110	$A_0 A_1^2 u$	5
7	111	$A_1^3 u$	11

The answer for  $N(t)$  must of course always be an integer, e.g. the line (1101011) at  $t = 3$  has  $a_1 = 6, a_2 = 2$  giving  $N(3) = 5$  and predicting via (15)  $N(7) = 11$ , as figure 6 also gives. By a similar means, the transition matrix  $A_0$  for the  $t \rightarrow 2t$  transition is found to be

$$A_0 = \begin{pmatrix} 1 & 2 \\ 0 & 0 \end{pmatrix}. \quad (16)$$

A very important result from [20, 21] can now be stated. Let  $u$ , with  $u^T = (2, 0)$  be the initial vector as  $a_1 = 2$  and  $a_2 = 0$  at  $t = 0$ . Let  $c$ , with  $c^T = (1, 2)$  be the counting vector as above. Let the binary representation  $t_B$  of  $t$  be

$$t_B = i_p i_{p-1} \cdots i_2 i_1, \quad (17)$$

so that

$$t = \sum_{\alpha=1}^p i_\alpha 2^{\alpha-1}. \quad (18)$$

Then we have

$$N(t) = \frac{1}{2} c^T A_{i_1} \cdots A_{i_p} u = \frac{1}{2} c^T \prod_{\alpha=1}^p A_{i_\alpha} u. \quad (19)$$

Here the factor  $\frac{1}{2}$  compensates for double counting due to use of overlapping 2-blocks. To see that the building up of the configuration of the automaton with time follows the binary tree structure of  $t_B$ , it is sufficient to note that, if  $t = t_B$  in the binary, then the times represented in the binary as  $t_B 1$  and  $t_B 0$  are equal to  $(2t + 1)$  and  $2t$ . So insertion of a 0 or 1 at the right-hand side end of a binarily represented time, inserts into the previous  $N(t)$ , to the immediate left of  $c^T$ , a matrix  $A_0$  or  $A_1$  to get  $N(t)$  at the new time. Thus we find the results shown in table 1.

A single example illustrates, confirming also the correctness of the ordering of the transition matrices in (19) relative to (17). Consider the  $p = 4$  example

$$t = 13, \quad t_B = 1101 = 1 + 0.2 + 1.2^2 + 1.2^3.$$

To get  $N(13)$  directly from the initial vector  $u$  at time  $t = 0$ , we form in turn the vectors

$$A_1 u, \quad A_1 A_1 u, \quad A_0 A_1 A_1 u, \quad A_1 A_0 A_1 A_1 u.$$

thereby reaching the vectors for times 1, 3, 6, 13. Then  $N(t) = \frac{1}{2} c^T A_1 A_0 A_1 A_1 u = 15$  comes out correctly.

We repeat that (17)–(19) (apart possibly from the external factor one-half in (19)) apply to all cases which depend on the use of the binary representation of  $t$ . In any example, the problem is to identify appropriately the vectors  $c, u$  and the matrices  $A_0, A_1$ .

2.3. Proof of (12)

Here we proceed from the formula (19) to proof, new here, of the result (2.3) first given in [1]. Firstly, we note that the facts

$$A_0^r = A_0, \quad A_0 u = u, \tag{20}$$

which tell us that any string of zeros in  $t_B = t$  can be replaced by a single zero regardless of its actual length, and that zeros in  $t_B$  to the left of all ones can be ignored (of course). The key step of writing  $A_0 = \frac{1}{2}uc^T$  directs us towards a general procedure. In this we write, for example for  $t_B = 1^x 0^y 1^z$ ,

$$\frac{1}{2}c^T A_1^z A_0^y A_1^x u = \frac{1}{2}c^T A_1^z A_0 A_1^x u = \left(\frac{1}{2}c^T A_1^z u\right) \left(\frac{1}{2}c^T A_1^x u\right). \tag{21}$$

For any string of ones of length  $n$  in  $t_B$ , (21) indicates that we should define and evaluate the quantities

$$w_n = \frac{1}{2}c^T A_1^n u, \tag{22}$$

for then the right-hand side of (21) has a product structure of the type  $w_z w_x$  required by (12). Since  $A_1^2 = A_1 + 2I$ , we find that

$$w_{n+2} = w_{n+1} + 2w_n. \tag{23}$$

The initial values of the sequence of  $w_n$  are given by  $w_0 = 1$  and  $w_1 = 3$ . It follows that the  $w_n$  can be identified with the  $\chi_n$  defined in the appendix, and that proof of (12) can be completed.

2.4. Evaluation of the quantity  $V(M)$  of (6)

To obtain the total weight (6) for the first  $2^M$  rows of figure 6, we use the result

$$V(M) = \frac{1}{2}c^T A^M u, \quad A = A_0 + A_1 = \begin{pmatrix} 2 & 4 \\ 1 & 0 \end{pmatrix}. \tag{24}$$

To see that this is correct, we expand  $A^M$  as a sum of  $2^M$  ordered products of the transition matrices  $A_0$  and  $A_1$ . Recalling that powers of  $A_0$  on the right of any term can be omitted by reason of (20), we see the required sum emerges.

From the characteristic equation of  $A$  we find the difference equation

$$V(M + 1) = 2V(M) + 4V(M - 1). \tag{25}$$

Since  $V(0) = 1$  and  $V(1) = 4$ , we can show that

$$V(M) = 2^M F_{M+2}, \tag{26}$$

where  $F_n$  denotes the  $n$ th Fibonacci number.

**3. The reversible cellular automaton RCA1**

3.1. Passage to the matrix cellular automaton MCA1

Our work on RCA1 is confined to evolution in discrete time from the single seed initial state of (5) on the basis of the updating rule (1), namely

$$x_n(t + 1) = x_{n-1}(t) + x_n(t - 1) \pmod{2}. \tag{27}$$

Here we wish to analyse the time evolution of RCA1 with the aid of an equivalent matrix cellular automaton whose time evolution is governed by a rule that is first order in time like Rules 90 and 150. The motivation is that we can thereby apply appropriate generalization of techniques already developed systematically for rules like these.

We start by defining

$$y_n(t) = x_n(t - 1), \quad \text{for all } t \text{ and all } n \geq 0. \quad (28)$$

Then (27) and (28) imply

$$\begin{pmatrix} y_n(t+1) \\ x_n(t+1) \end{pmatrix} = \begin{pmatrix} x_n(t) \\ x_{n-1}(t) + y_n(t) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} y_n(t=0) \\ x_n(t=0) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \delta_{n0}. \quad (29)$$

Forming a generating function

$$\Pi(t, \lambda) = \sum_{n=0}^{\infty} \lambda^n \begin{pmatrix} y_n(t) \\ x_n(t) \end{pmatrix} = \begin{pmatrix} Y(t, \lambda) \\ X(t, \lambda) \end{pmatrix}, \quad (30)$$

we find

$$\Pi(t+1, \lambda) = \begin{pmatrix} X(t, \lambda) \\ Y(t, \lambda) + \lambda X(t, \lambda) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & \lambda \end{pmatrix} \Pi(t, \lambda). \quad (31)$$

Equations (31) and (29) then give

$$\Pi(t+1, \lambda) = M^t \Pi(t, \lambda), \quad M = \begin{pmatrix} 0 & 1 \\ 1 & \lambda \end{pmatrix}, \quad \Pi(0, \lambda) = \begin{pmatrix} Y(0, \lambda) \\ X(0, \lambda) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (32)$$

and hence

$$X(t, \lambda) = \text{Tr } M^t P = \text{Tr}(G + \lambda P)^t P, \quad (33)$$

where

$$G = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad P = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \quad (34)$$

This will enable us to evaluate the weight  $N(t)$  of the state of RCA1 at time  $t$ , since

$$X(t, \lambda = 1) = \sum_{n=0}^{\infty} x_n(t) = N(t). \quad (35)$$

We stress the fact modulo 2 arithmetic governs the steps of the calculation of  $x_n(t)$  throughout the discussion.

Noting the role of the two-by-two matrix  $M^t P$  in (32) and hence (33), we make the definition

$$M^t P = \sum_{n=0}^{\infty} c_n(t) \lambda^n. \quad (36)$$

This is the generating function of matrix-valued cell variables  $c_n(t)$ . Multiplication of (36) on the left by  $M$  then yields

$$c_n(t+1) = P c_{n-1}(t) + G c_n(t), \quad \text{for all } n, t, \quad (37)$$

together with  $c_n(t=0) = \delta_{n0}$ .

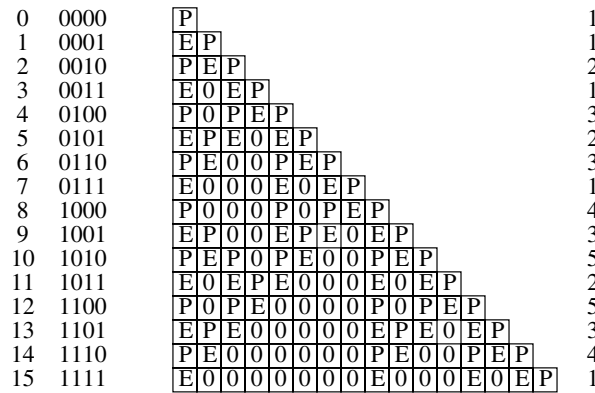


Figure 7. The matrix cellular automaton MCA1.

Thus we have reached the sought after definition of the first-order matrix cellular automaton MCA1 to which RCA1 is equivalent. Its cell variables  $c_n(t)$  take values in the space  $M_{2,2}(Z_2)$  of two-by-two matrices whose elements belong to  $Z_2$ . Equation (37) allows the generation, from the given initial state, by a C-program, of the display of figure 7 for times  $0 \leq t \leq 15$ . Note that figure 7 involves non-trivially only the matrices  $P$  and the matrix  $E$

$$e = GP = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \tag{38}$$

where  $P$  and  $G$  are given in (34). One may, as we show by examples, check entries in figure 7 by hand. In doing so it is necessary but easy both to take account of the non-commutativity of matrix multiplication, and also to appreciate the role of modulo 2 arithmetic in the process. Two examples should suffice

$$c_3(t = 4) = Pc_2(t = 3) + Gc_3(t = 3) = PE + GP = E + 0 = E,$$

$$c_5(t = 7) = Pc_4(t = 6) + Gc_5(t = 6) = PP + GE = P + P = 0 \pmod 2.$$

The  $t$ th line of figure 7 gives directly the configuration of MCA1 at time  $t$ . It also gives the matrix  $M^t P$  as a matrix polynomial in  $\lambda$  if one associates  $\lambda^n$  with the  $n$ th cell of the line for each  $n > 0$ . By taking the trace of this polynomial at  $\lambda = 1$  we should recover (35). We can see otherwise this happens because  $\text{Tr } c_n(t) = x_n(t)$  with values 0 and 1 coming from  $c_n(t) = E$  and  $P$ . This corresponds to the observation that replacing of  $E$  and  $P$  by 0 and 1 in figure 7 for MCA1 puts it into exact coincidence with figure 2 for RCA1.

Inspection of figure 7 provides the key to obtaining expressions for  $N(t)$ . The following observation holds for all times  $t$ . Given row  $t$  of the display, one can directly write down

- (a) row  $2t$  by replacing each entry of row  $t$  by non-overlapping two cell blocks, according to  $P, E, 0 \longrightarrow P0, PE, 00$  respectively,
- (b) row  $(2t+1)$  by replacing each entry of row  $t$  similarly according to  $P, E, 0 \longrightarrow EP, E0, 00$ .

For example the  $t = 3$  line  $E0EP$  gives the  $t = 6$  line  $(PE)(00)(PE)(P0)$  or  $PE00PEP$ , and the  $t = 7$  line  $(E0)(00)(E0)(P0)$  or  $E000E0EP$ , just as computer-produced figure 7 tells us. It follows, as in section 2, that we have transition matrices whose rows and columns are labelled in the order  $E, P$

$$A_0 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = A_0^T. \tag{39}$$

where, e.g., the first column of  $A_0$  tells us that from each entry  $E$  at time  $t$  there arises at time  $2t$  one entry of each of  $E$  and  $P$ . As usual zero entries are irrelevant in counting studies and are ignored.

The method using these matrices is the same as was followed in section 2, in the work on Rule 150. Corresponding to  $t = t_B$  given in binary by (17), we can use

$$N(t) = c^T A_{i_1} \cdots A_{i_p} u = c^T \prod_{\alpha=1}^p A_{i_\alpha} u. \quad (40)$$

Since happily one does not encounter here an overlap problem like that which complicated the treatment of Rule 150, in section 2.2, there is no factor of one-half in (4.9). Also the initial and the counting vector are each given by

$$c = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = u, \quad (41)$$

and the matrices of (39) must be used. It is easy to check that the results which (40) gives for  $N(t)$  agree with those displayed on the right side of figure 7.

### 3.2. An algorithm for writing down any $N(t)$ directly

To make progress with evaluating  $N(t)$  explicitly, we first note the results

$$c^T A_1^y = c^T, \quad A_0^y u = u, \quad (42)$$

which tells us that any string of ones at the right-hand end of  $t_B$  can be ignored, which is useful, and so can any string of zeros at the left-hand end, which is trivial.

Anticipating that strings of ones in  $t_B$  separated by non-trivial strings of zeros are to be as significant in the analysis as they were in section 2.2, consider in order the cases

$$N(1^{b_1} 0^{a_1}) = c^T A_0^{a_1} A_1^{b_1} u, \quad N(1^{b_2} 0^{a_2} 1^{b_1} 0^{a_1}) = c^T A_0^{a_1} A_1^{b_1} A_0^{a_2} A_1^{b_2} u.$$

From

$$A_1^r = \begin{pmatrix} 1 & r \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad A_0^r = A_1^{rT},$$

we find easily

$$N(1^{b_1} 0^{a_1}) = 1 + a_1 b_1 = K_1, \quad (43)$$

$$N(1^{b_2} 0^{a_2} 1^{b_1} 0^{a_1}) = K_1 K_2 + K_{12}, \quad (44)$$

where we have defined the *contraction* symbols

$$K_j = 1 + a_j b_j, \quad \text{and for } j < k \text{ only} \quad K_{jk} = a_j b_k. \quad (45)$$

The symbols have been called contractions because an algorithm resembling Wick's theorem in quantum field theory can be established for writing down directly and explicitly the value of  $N(t)$ , for  $t = L_n L_{n-1} \cdots L_2 L_1$ , where we have used the abbreviation  $L_r = 1^{b_r} 0^{a_r}$ . The required answer is a sum of terms one for each ordered partition of  $n$ . Each term is a product of non-overlapping contraction of the two types given in (45) that exhaust the integers 1 to  $n$ , with the qualification that the presence of  $K_{rs}$  dictates that there can also be as a factor of the same term no contraction either of type  $K_u$  or  $K_{uv}$  for  $r < u < s$  and  $1 \leq v \leq n$ .

We note that the two terms of (44) which correspond to the partitions 11 and 2 of  $n = 2$  do illustrate this.

Similarly we can write

$$N(L_4L_3L_2L_1) = K_1K_2K_3K_4 + K_1K_2K_3K_4 + K_1K_2K_3K_4 + K_{12}K_3K_4 + K_1K_{24} + K_{13}K_4 + K_{12}K_{34} + K_{14}. \tag{46}$$

It is not hard to verify directly that this is correct. Note that (46) illustrates the language ‘exhaustive and non-overlapping’ of our general statement. For example, none of  $K_{12}K_3$ ,  $K_{13}K_{24}$ ,  $K_{13}K_2K_4$  are allowed in (46): the first is not exhaustive, the remaining ones show what the sense of non-overlapping is. The terms of (46) correspond to the eight ordered partitions of four

$$1111, 112, 121, 211, 13, 31, 22, 4.$$

One can either set about general evaluations by an iterative procedure or use what follows to provide a proof of the algorithm. Since

$$A_0^r = I + ru_2u_1^T, \quad (u_i)_j = \delta_{ij}, \quad 1 \leq i, j \leq 2, \quad u_2 = u,$$

we can derive for  $N(t) = N(L_nL_{n-1} \cdots L_2L_1)$ , the result

$$N(t) = c^T R_1R_2 \cdots R_{n-1}(1 + a_nuu_1^T)A_1^{b_n}u = N(L'_{n-1}L_{n-2} \cdots L_2L_1) + a_nb_nN(L_{n-1}L_{n-2} \cdots L_2L_1), \tag{47}$$

where  $R_r = A_0^{ar}A_1^{br}$  and  $L'_{n-1} = 1^{b_n+b_{n-1}}0^{a_{n-1}}$ . Here the result  $u_1^T A_1^{b_n} u = b_n$  has been used. Equation (47) can be used as the basis of an iterative procedure.

### 3.3. Derivation of the result (7)

As in section 2.4, we have

$$V(M) = c^T A^M u, \quad \text{where } A = A_0 + A_1 = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \pmod 2.$$

The transition matrix  $A$  obeys the equation

$$A^2 - 4A + 3I = 0.$$

Thus  $V(M)$  obeys a difference equation with solution of the type  $\alpha + \beta 3^M$ , and initial conditions  $V(0) = 1, V(1) = 2$ , so that the answer quoted in (7) follows.

## 4. The reversible cellular automaton RCA2

### 4.1. Passage to the matrix cellular automaton MCA2

The study progresses as in section 3.1, with evolution from the single seed initial state (5) according to the updating rule

$$x_n(t + 1) = x_{n-1}(t) + x_n(t) + x_n(t - 1). \tag{48}$$

The definitions (28)–(30) may be retained, but now the quantity  $\Pi(t, \lambda)$  of (30) is specified by an equation very much like, but not identical to (32), namely

$$\Pi(t, \lambda) = M^t \Pi(0, \lambda), \quad M = \begin{pmatrix} 0 & 1 \\ 1 & 1 + \lambda \end{pmatrix} = G + (1 + \lambda)P, \quad \Pi(t = 0, \lambda) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{49}$$



0	0000	P	1
1	0001	VP	2
2	0010	EEP	1
3	0011	PPVP	4
4	0100	VEVEP	3
5	0101	E0E0VP	2
6	0110	P0P0EEP	3
7	0111	VPVPVPVP	8
8	1000	EEVEEEVEP	3
9	1001	PEE0PEE0VP	6
10	1010	VE00VE00EEP	3
11	1011	E000E000PPVP	4
12	1100	P000P000VEVEP	5
13	1101	VP00VP00EPE0EP	6
14	1110	EEP0EEP0P0P0EEP	5
15	1111	PPVPVPVPVPVPVPVPVP	16

Figure 8. The matrix cellular automaton MCA2.

As in section 3.1, we write,

$$X(t, \lambda) = \text{Tr } M^t P = \text{Tr}[G + (1 + \lambda)P]^t P, \tag{50}$$

and pass to the matrix cellular automata MCA2 defined by writing

$$M^t P = \sum_{r=0}^{\infty} c_r(t) \lambda^r. \tag{51}$$

The coefficients in (51) satisfy

$$c_r(t + 1) = P c_{r-1}(t) + (G + P) c_r(t), \tag{52}$$

with  $c_n(t = 0) = P \delta_{n0}$ ,  $n \geq 0$ . The evolution of MCA2 follows and the first 16 lines of this is given, with some other data, in figure 8.

The notation here is as before plus the definition

$$V = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} = P + E. \tag{53}$$

It can be seen that the passage from any row  $t$  to row  $2t$  is achieved by the replacements  $P, E, V \rightarrow P0, VE, EE$ . Similarly, for passage from row  $t$  to row  $(2t + 1)$ ,  $P, E, V \rightarrow VP, E0, PP$ . It follows that as in section 3, we have transition matrices with rows and columns labelled in the order  $E, V, P$ :

$$A_0 = \begin{pmatrix} 1 & 2 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 1 \end{pmatrix} = A_0^T. \tag{54}$$

With  $t$  given in binary by (17), (4.9) here also yields  $N(t)$ , using the transition matrices of (54), plus appropriate initial and counting vectors  $c, u$ , given by  $c^T = (0, 1, 1)$ , and  $u^T = (0, 0, 1)$ .

4.2. Evaluation of the result (40) for RCA2

Actions of  $A_0^n$  and  $A_1^n$  on the basis  $u_i$ ,  $(u_i)_j = \delta_{ij}$ ,  $1 \leq i, j \leq 3$ , gives rise to results that may be proved by induction with use of table 1 for  $n = 0, 1$ . This yields

$$A_0^n = \begin{pmatrix} \chi_{n-1} & 2\chi_{n-2} & 0 \\ \chi_{n-2} & 2\chi_{n-3} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad A_1^n = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2\chi_{n-3} & \chi_{n-2} \\ 0 & 2\chi_{n-2} & \chi_{n-1} \end{pmatrix}. \quad (55)$$

To read these results for low  $n$  values requires the extension of the definition of  $\chi_r$  for negative  $r$  given, together with explicit values in table A.1 of the appendix.

One simple consequence of (55) holds for the counting vector  $c = u_2 + u_3$ . It is

$$c^T A_1^n = 2^n c^T, \quad \text{and hence} \quad N(1^n) = c^T A_1^n u = 2^n, \quad (56)$$

obvious by inspection of figure 3. Further consequences are

$$N(L_{ab}) = \chi_{a-1} + 2\chi_{a-2}\chi_{b-3}, \quad L_{ab} = 1^a 0^b \quad (57)$$

$$N(L_{ab}L_{cd}) = 2\chi_{a-2}\chi_{b-2}\chi_{d-2} + \chi_{a-1}(\chi_{c-1} + 2\chi_{c-2}\chi_{d-3}) + 4\chi_{a-2}\chi_{b-3}(\chi_{c-2} + 2\chi_{c-3}\chi_{d-3}). \quad (58)$$

To see that (58) for  $c = 0$  agrees with (57) requires use of (A.5) of the appendix.

There is no difficulty in getting similar results for  $N(L_3L_2L_1)$  etc, but no algorithm of Wick-theorem type for organizing the increasing complication has been found. The first entry of (56) enables an easy extension (factor  $2^c$ ) of (57) from  $L_{ab}$  to  $L_{ab}1^c$ , and from  $L_{ab}L_{cd}$  to  $L_{ab}L_{cd}1^c$ .

4.3. Derivation of the result (8)

Again, as in section 2.4, we have

$$V(M) = c^T A^M u, \quad \text{where} \quad A = A_0 + A_1 = \begin{pmatrix} 2 & 2 & 0 \\ 1 & 0 & 1 \\ 0 & 2 & 2 \end{pmatrix}. \quad (59)$$

The transition matrix  $A$  here obeys the equation  $A^3 - 4A^2 + 8I = 0$ , and has the eigenvalues  $(1 \pm \sqrt{5})$ , 2. It follows that  $V^M$  obeys the difference equation  $V(M+3) - 4V(M+2) + 8V(M) = 0$ , and that the solution for  $V(M)$  is of the form

$$V(M) = a(1 + \sqrt{5})^M + b(1 - \sqrt{5})^M + c2^M.$$

To determine  $a$ ,  $b$  and  $c$  here, we calculate and use the initial conditions  $V(0) = 1$ ,  $V(1) = 3$ ,  $V(2) = 8$ . Comparing the first two terms of the solution so obtained with the solution of the difference equation for the Fibonacci numbers

$$F_{M+2} = F_{M+1} + F_M, \quad F_0 = 0, \quad F_1 = 1,$$

which is

$$F_M = \frac{\alpha^M - \beta^M}{\alpha - \beta}, \quad \alpha, \beta = \frac{1}{2}(1 \pm \sqrt{5}),$$

we are led, after attending to some detail, to the answer (8) for  $V(M)$ .

### 5. The reversible cellular automaton RCA3

In this case, the cell variables  $x_n(t)$  at each instant of time are defined for all  $n \in \mathbb{Z}$ . We consider evolution from single seed using the updating rule (3)

$$x_n(t+1) = x_{n-1}(t) + x_n(t) + x_{n+1}(t) + x_n(t-1) \pmod{2}. \quad (60)$$

Figure 4 displays the evolution for  $0 \leq t \leq 15$ .

As in section 3.1, we reach the first-order matrix cellular automaton associated with RCA3. It is defined by

$$\Pi(t, \lambda) = M^t \Pi(0, \lambda), \quad M = G + (\lambda + 1 + 1/\lambda)P, \quad \Pi(t=0, \lambda) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (61)$$

The matrices  $G, P$  occurring here are shown in (34). Using (36)

$$M^t P = \sum_{r=-\infty}^{\infty} c_r(t) \lambda^r, \quad (62)$$

we are led to

$$c_r(t+1) = P c_{r-1}(t) + (G + P) c_r(t) + P c_{r+1}(t), \quad c_n(t=0) = P \delta_{n0}, \quad \text{for all } n, t. \quad (63)$$

Using (63) as the definition of MCA3, we obtain the display of figure 9.

The symbols in figure 9 have the same meaning as those of figure 6, and each  $P$  and each  $V$  contribute one to the counting of the weight of  $N(t)$  for each  $t$ .

#### 5.1. Derivation of formulas

It is possible, by scrutiny of figure 9, to determine the rules that underlie the passage from the  $t$ -line to the lines for times  $2t$  and  $(2t+1)$ . These are

$$P \longrightarrow 0P0, \quad V \longrightarrow EEE, \quad E \longrightarrow EVE, \quad \text{for } t \text{ to } 2t, \quad (64)$$

$$P \longrightarrow PVP, \quad V \longrightarrow PPP, \quad E \longrightarrow 0E0, \quad \text{for } t \text{ to } (2t+1), \quad (65)$$

as well, in each case, as  $0 \longrightarrow 000$ . However, the problem of overlapping blocks appears at this point somewhat as it did, see section 2.2, for Rule 150. To see how this affects our progress consider either one of the passages (64) or (65), say the former. Suppose the pair of entries  $xy$  occurs in the time  $t$ -line, and that in the passage to time  $2t$  we have  $x \rightarrow aba$  and  $y \rightarrow cdc$ . Then the  $xy$  pair gives rise at time  $2t$  to the sequence  $(.)bed(.)$ , where  $e = a + c$  modulo 2. The entries  $(.)$  depend on the (here unspecified) left-hand side neighbour of  $x$  and the (likewise unspecified) right-hand side neighbour of  $y$  at time  $t$ . However, the central entries  $bed$  are uniquely determined by the 2-block  $xy$  in a way that respects the dictates of modulo 2 arithmetic.

To proceed, it is necessary to consider overlapping 2-blocks in the spirit of the work from [20, 21] for first-order cellular automata like Rule 150 reviewed in section 2.2. There are in fact eight of these to consider when we follow the most straightforward approach, so that the matrices  $A_0$  and  $A_1$  that govern the transitions  $t$  to  $2t$  and  $t$  to  $(2t+1)$  are  $8 \times 8$  (sparse) matrices. The 2-blocks, taken in the order that we use to label the rows and columns of  $A_0, A_1$  and  $A = A_0 + A_1$ , are

$$0P, PV, PE, EE, EV, V0, E0, PP.$$

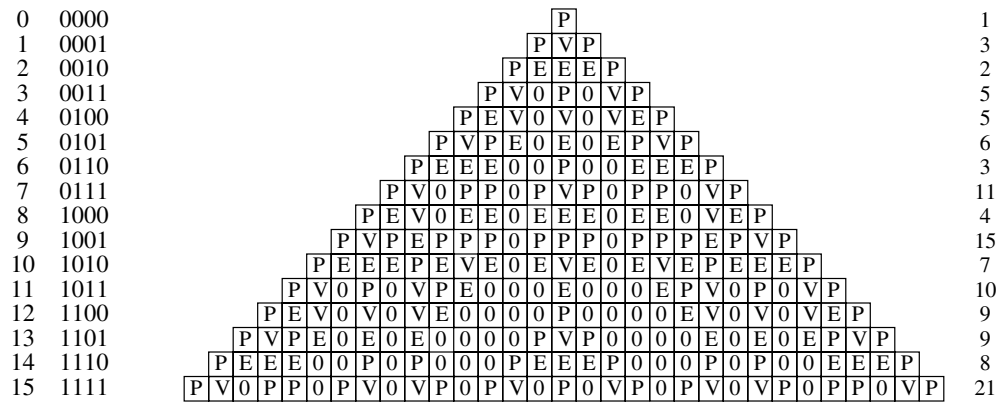


Figure 9. The matrix cellular automaton MCA3.

Table 2. Data for deducing the EV column of  $A_0$  and  $A_1$ .

Time $t$	$E$	$V$
Time $2t$	? $V$ 0 $E$ ?	
Time $(2t + 1)$	? $E$ $P$ $P$ ?	

We do not need to distinguish between 2-blocks  $ab$  and  $ba$ . We ignore the 2-block 00 as it does not contribute to the counting, and omit the 2-block  $VV$  which does not occur even in any extension of figure 9 to later  $t$ .

The correct treatment of each of these 2-blocks follows lines seen in table 2 for a typical 2-block.

The line for time  $2t$  here uses (64) while the line for time  $(2t + 1)$  uses (65). The key fact is that the central entries  $V0E$  and  $EPP$  are uniquely determined. These entries tell us what values to put into the  $EV$  or fifth columns of  $A_0$  and  $A_1$ : the 2-block  $EV$  at time  $t$  is seen to give rise, at time  $2t$  to the 2-blocks  $V0$  and  $E0$ , and, at time  $(2t + 1)$ , to the 2-blocks  $EP$  and  $PP$ . This accounts for the entries one in the sixth and seventh place in the fifth column of  $A_0$ , and in third and eighth place for  $A_1$ . Hence, treating the other 2-blocks similarly, we obtain the matrices  $A_0$  and  $A_1$ :

$$A_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}. \tag{66}$$

Once these matrices are written down all the modulo two arithmetic has been done and we are ready to turn to the evaluation of the weights of the lines of figure 4 for general times. The initial vector could be taken to be  $(2, 0, 0, 0, 0, 0, 0, 0)$  since each  $P$  and each  $V$  contribute

one to the counting, and we have 2-blocks  $0P$  and  $P0$  initially. However, it is clear that our treatment of 2-blocks consistently double counts all contributions, and so we use the initial vector  $u$  given by

$$u^T = (1, 0, 0, 0, 0, 0, 0), \quad (67)$$

and do not include an external factor two. The counting vector  $c$  is given by

$$c^T = (1, 2, 1, 0, 1, 1, 0, 2). \quad (68)$$

The formula for evaluation of  $N(t)$  for  $t = t_B$  given by (17) again takes the form (4.9) except that now the matrices  $A_0$  and  $A_1$  are given by (66). It is simple if somewhat tedious to check by hand that all the counting results in figure 4 can be reproduced exactly. To go any further it is essential to employ a computer algebra package, e.g. MAPLE, as used here.

To obtain a formula for  $V(M)$ , defined in (6) that counts the total weight of all configurations of RCA3 for times from  $t = 0$  up to the time  $(2^M - 1)$ , we introduce  $A = A_0 + A_1$ . Since its distinct eigenvalues are  $2, 1, -1, -2, \rho_{\pm} = \frac{1}{2}(3 \pm \sqrt{17})$ , we try, successfully, to evaluate  $V(M)$  using the ansatz

$$V(M) = 2^M a + b + (-1)^M c + (-2)^M d + e_+ \rho_+^M + e_- \rho_-^M, \quad (69)$$

and data computed for  $V(M)$  for  $M = 0, \dots, 5$ . This gives rise to the formula, checked also for  $M = 6, 7, 8, \dots$ :

$$V(M) = \frac{2}{3} 2^M - (-1)^M \frac{1}{6} + e_+ \rho_+^M + e_- \rho_-^M, \quad (70)$$

with  $e_{\pm} = \pm(\rho_{\pm} + 2)/(2\sqrt{17})$ . It may be thought preferable to give this result in the form

$$V(M) = \frac{1}{2}(\chi_M + x_M), \quad (71)$$

where  $\chi_M$  is given by (A.2) from the appendix, and  $x_M$  is the solution of the difference equation

$$x_{M+2} = 3x_{M+1} + 2x_M, \quad x_0 = 1, \quad x_1 = 5,$$

although no deep significance is implied by doing so.

The large  $n \propto 2^M$  behaviour of  $V(M)$  can be seen from (70) (see e.g. [8], p 56) to correspond to a fractal dimension  $d = 1.8325$ . This may be compared with the numbers  $d = 1.5850$  and  $d = 1.6942$  that arise similarly from (7) and (8).

The result

$$N(t_m) = c^T A_1^m u = \chi_m, \quad (72)$$

obvious in figure 4 for  $t_m = t_B = 1^m$  can be proved. One observes that  $A_1^r u$  for  $r = 0, 1, 2, 3$  spans the set of vectors  $A_1^r u$  for all integers  $r \geq 0$ . In fact (MAPLE),

$$(A_1^4 - A_1^3 - 3A_1^2 + A_1 + 2)u = 0. \quad (73)$$

Since the equation  $x^4 - x^3 - 3x^2 + x + 2 = 0$  has roots  $2, 1, -1, -1$ , we employ the ansatz

$$N(t_m) = a2^m + b + (-1)^m(c + dm), \quad (74)$$

and computed values for  $m = 0, 1, 2, 3$  to obtain  $a = \frac{4}{3}$ ,  $b = 0$ ,  $c = \frac{1}{3}$ ,  $d = 0$ . This allows us to identify the  $N(t_m)$  with the numbers  $\chi_m$  given by (A.2) in the appendix.

More can definitely be done often along suggestive lines. In the absence of a systematic general approach, however, we confine ourselves to one further example, the evaluation of  $N(t)$  for  $t = t_B = 1^m 0^n$ . Consider first the case  $m = 1$ . Setting  $w = A_1 u$ , we find, cf (73),

$$(A_0^4 - A_0^3 - 3A_0^2 + A_0 + 2)w = 0. \tag{75}$$

We therefore employ the ansatz (74), and computed values for  $n = 0, 1, 2, 3$ , to get the formula

$$N(t = 10^n) = \frac{4}{9}2^n + 2 + \frac{1}{9}(-1)^n(5 + 3n). \tag{76}$$

It can be checked for various  $n \geq 4$  that (76) does yield a (correct) integer value.

In fact (76) is a special case of a more general result, one whose proof depends on the fact that (75) holds also for  $w = A_1^m u$  for all positive integers  $m$ . This result can be cast eventually into the form

$$N(t = 1^m 0^n) = c^T A_0^n A_1^m u = \phi_m(\phi_n + (-1)^n) + \phi_{m+1}, \tag{77}$$

where, for  $n = 0, 1, 2, \dots$ , the  $\phi_n$  belong to the sequence

$$0, 1, 2, 3, 8, 13, 30, 55, 116, \dots \tag{78}$$

A large number of spot checks against a large amount of computer output have been performed. It is seen by inspection of these numbers that the  $\phi_n$  are related to the  $\chi_n$  by means of  $\chi_n = \phi_n + \phi_{n+1}$ . This enables a systematic discussion of them, given in the appendix.

We note also that (77) reduces to (76) for  $m = 1$ , using (A.10) from the appendix, and to (72) for  $n = 0$  using (A.8) there.

### 6. Additional remarks

In section 5, we have shown that the  $r$ -block methods introduced in [20, 21], for linear first-order cellular automata with cell variables taking values in  $Z_2$ , can be extended to first-order matrix cellular automata whose cell variables take values in  $M_{2,2}(Z_2)$ . And of course these were introduced to allow treatment of reversible second-order linear cellular automata whose cell variables take values in  $Z_2$ . We have explicitly used only two-blocks, but there is no obstacle in principle in going to higher  $r$ . This will be essential when algebraic work on evolution of RCA1–3 from states more complicated than single seed initial states is undertaken.

The cellular automata RCA1–3 are specially simple examples, based on the updating rules (1–3). There is, beyond these, see chapter 9 of [9] for an extensive discussion, a wide diversity of interesting reversible cellular automata.

Further, if one wishes to consider reversible cellular automata for which cell variables take values in, say,  $Z_3$ , it should be clear enough that methods like those described here can be applied. Transition matrices  $A_r$  for the transitions from time  $t$  to times  $(3t + r)$ ,  $r = 0, 1, 2$ , arise; their sum  $A = \sum_{i=1}^3 A_i$  can be applied to finding results for

$$W(M) = \sum_{t=0}^{3^M-1} M(t),$$

analogous to those found for  $V(M)$  above.

### Appendix. The quantities $\chi_n$ and $\phi_n$

The set of quantities  $\chi_n$  appear first in section 2.2. They acquire significance here simply because it turns out that many useful formulas for the  $N(t)$  of various cellular automata involve them. See e.g. (12) for Rule 150, (57) and (58) for RCA2, and (72) for RCA3. A related set of quantities  $\phi_n$  likewise draw themselves to our attention because results for RCA3, see (77), appear to be most naturally presented in terms of them.

The  $\chi_n$  are defined [1], initially for  $n = 0, 1, \dots$ , by the relations

$$\chi_{n+1} = 2\chi_n + (-1)^n, \quad \chi_0 = 1, \quad (\text{A.1})$$

so that easily

$$\chi_n = \frac{1}{3}(2^{n+2} + (-1)^{n+1}), \quad n = 0, 1, 2, \dots \quad (\text{A.2})$$

The formula (A.2) can also be derived by solving the difference equation

$$\chi_{n+1} = \chi_n + 2\chi_{n-1}, \quad \chi_0 = 1, \quad \chi_1 = 3, \quad (\text{A.3})$$

which can also be seen to be implied by (A.1). We have also these results

$$\chi_{n+1} + \chi_n = 2^{n+2}, \quad \chi_{n+2} = 4\chi_n + (-1)^n. \quad (\text{A.4})$$

In section 4, various consistency checks require further identities, all easily proved. Examples include

$$\chi_n \chi_m + 2\chi_{n-1} \chi_{m-1} = \chi_{n+m+1}, \quad (\text{A.5})$$

$$\chi_{n-1} \chi_{m+1} - \chi_n \chi_m = (-1)^m \chi_{n-1} + (-1)^n \chi_m, \quad (\text{A.6})$$

and the  $m = n$  case of (A.6) gives  $\chi_n^2 - \chi_{n+1} \chi_{n-1} = (-2)^{n+1}$ .

One can also use the data in (A.3) to obtain the generating function

$$F(y) = \sum_{n=0}^{\infty} \chi_n y^n = \frac{2y+1}{(1-2y)(y+1)}, \quad (\text{A.7})$$

from which the solution (A.2) follows easily. Also, (A.5) can be established by means of the identity

$$F(y)^2 + 2(yF(y) + 1)^2 = F'(y),$$

itself easily verified by use of (A.7).

We note here also, since the need arises later, that (A.3) can be used to assign values to  $\chi_n$  for negative integral values of  $n$ . Some values are shown in table A.1, together with those of the sequence  $\phi_n$ , to which we turn next.

The values of  $\phi_n$  for positive  $n$  shown in table A.1 emerged, see (78), from computer data for RCA3. By inspection it was seen that they are related to the  $\chi_n$  by means of

$$\phi_n + \phi_{n+1} = \chi_n. \quad (\text{A.8})$$

As a consequence, they obey the difference equation

$$\phi_{n+3} = 3\phi_{n+1} + 2\phi_n. \quad (\text{A.9})$$

**Table A.1.** Values of  $\chi_n$  and  $\phi_n$  for  $-5 \leq n \leq 8$ .

$n$	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
$\chi_n$	$\frac{3}{8}$	$-\frac{1}{4}$	$\frac{1}{2}$	0	1	1	3	5	11	21	43	85	171	341
$\phi_n$	$\frac{17}{8}$	$-\frac{7}{4}$	$\frac{3}{2}$	-1	1	0	1	2	3	8	13	30	55	116

Since  $x^3 - 3x - 2 = (x + 1)^2(x - 2)$ , we use the ansatz

$$\phi_n = a2^n + (-1)^n(b + cn),$$

to solve (A.9). Using the values for  $\phi_n$  for  $n = 0, 1, 2$  from table A.1 as initial conditions, we find

$$\phi_n = \frac{1}{9}[2^{n+2} + (-1)^n(3n - 4)]. \tag{A.10}$$

Equation (A.10) affords the quickest way of building up the sequence  $\phi_n$  via its consequence

$$\phi_n = 2\phi_{n-1} + (-1)^n(n - 2). \tag{A.11}$$

We can also evaluate the generating function  $G(y)$  of the  $\phi_n$ :

$$G(y) = \sum_{n=0}^{\infty} \phi_n y^n = \frac{y(2y + 1)}{(y + 1)^2(1 - 2y)},$$

and, by use of partial fractions, etc, check that one recovers from it the solution (A.10). One can also check that substitution of (A.10) into (A.8) reproduces the result (A.2) for  $\chi_n$ .

**References**

[1] Wolfram S 1983 Statistical mechanics of cellular automata *Rev. Mod. Phys.* **55** 601–44  
 [2] Wolfram S 1986 *Theory and Applications of Cellular Automata* (Singapore: World Scientific)  
 [3] Toffoli T and Margolus N 1988 *Cellular Automata Machines: A New Environment for Modelling* (Cambridge, MA: MIT Press)  
 [4] Gutowitz H A 1990 *Cellular Automata: Theory and Experiment* (Amsterdam: Elsevier)  
 [5] Wolfram S 1994 *Cellular Automata and Complexity, Collected Papers* (New York: Addison-Wesley)  
 [6] Voorhees B 1996 *Computational Analysis of One-dimensional Cellular Automata* (Singapore: World Scientific)  
 [7] Chaudhuri P P 1997 *Additive Cellular Automata: Theory and Experiment* (Los Alamitos, CA: IEEE Computer Society Press)  
 [8] Ilachinski A 2001 *Cellular Automata: A Discrete Universe* (Singapore: World Scientific)  
 [9] Wolfram S 2002 *A New Kind of Science* (Champaign, IL: Wolfram Media)  
 [10] Ablowitz M A, Kreiser J M and Tahktajan L A 1991 Class of stable multistate time-reversible cellular automata with rich particle content *Phys. Rev. A* **44** 6909–12  
 Ablowitz M A, Kreiser J M and Tahktajan L A 1991 Stable multistate time-reversible cellular automata with rich particle content *Quaestiones Mathematicae* **15** 325–43  
 [11] 't Hooft G 1988 Equivalence relations between deterministic and quantum mechanical systems *J. Stat. Phys.* **53** 323–44  
 [12] Guan P and He Y 1986 Exact results for deterministic cellular automata with additive rules *J. Stat. Phys.* **43** 463–78  
 [13] Takesue S 1987 Reversible cellular automata and statistical mechanics *Phys. Rev. Lett.* **59** 2499–502  
 [14] Takesue S 1989 Ergodic properties and thermodynamics of elementary cellular automata *J. Stat. Phys.* **56** 371–402  
 [15] Vatan F 1988 Intractible problems in reversible cellular automata *Int. J. Theor. Phys.* **27** 801–18  
 [16] Hillman D 1991 The structure of reversible cellular automata *Physica D* **52** 277–92  
 [17] Urias J 1993 Arithmetic representation of cellular automata *Physica D* **68** 437–46  
 [18] Poundstone W 1985 *The Recursive Universe* (Oxford: Oxford University Press)



- 
- [19] Sigmund K 1993 *The Game of Life* (Oxford: Oxford University Press)
- [20] Willson S J 1987 The equality of the fractal dimensions for certain cellular automata *Physica D* **24** 179–89  
Willson S J 1987 Computing fractal dimensions for certain cellular automata *Physica D* **24** 190–206
- [21] Takahashi S 1990 Cellular automata and multifractals: dimension spectra of linear cellular automata *Physica D* **45** 36–48