On the Existence and the Number of Fixed Points of Dynamical Systems Defined by Cellular Automata.

S. P.Banks and S. Djirar

Department of Automatic Control and Systems Engineering

University of Sheffield

Mappin Street

SHEFFIELD S1 4DU

Research report No 491

November, 1993.

Abstract

A technique for determining fixed points and their number in one- and two-dimensional cellular automata based on graph theory is given. The method is simple to apply and can easily be implemented on a computer.

Keywords:Cellular automata, Fixed points.

1 Introduction In this paper we shall consider fixed points in one and two dimensional cellular automata [1, 3]. A formula for the number of fixed points of the majority rule in one dimension and its applications has been given in [2, 3]. The one dimensional case has recently been considered in [4] where certain operators are constructed to determine the fixed points. The method, however, is complicated and is difficult to generalize to two-dimensions. Here we give a simple algorithm which finds all fixed points of one or two dimensional systems, based on graph theory. The method produces an easily computable result in both the one and two dimensional cases and we give several examples. All the fixed points are found by applying the adjacency matrix of our graphs. Graphs as finite state machines have been used in the computation theory of cellular automata [6], but not previously for finding fixed points. In section 2 we consider one dimensional problems for a rule of any length and section 3 we consider the case of 9 bit two dimensional rule with periodic boundary conditions.

Note that the existing methods cited above are all quite complicated and difficult to generalize, to other rules or higher dimensions, and the number of fixed points given in [2, 3] is only for a specific rule using methods particular to that rule. Here we emphasize that our method applies to **any** rule in one and two dimensions.

2 One Dimensional Systems For notational simplicity we shall consider only 3-bit rules. The generalizations to rules of arbitrary length is trivial (this is a main advantage of our method). Consider a dynamical system with a binary state vector of the form

$$x = (x_1, x_2, x_3.., x_n), \quad x_i \in Z_2, \quad 1 \leq i \leq n. \tag{2.1}$$

which is defined by

$$x(k+1) = F(x(k)) \quad (k \geq 1) \tag{2.2}$$

and F is given by a local rule of order 3. Thus, $(Fx_n)_i = y_i$, where

$$y_i = \begin{cases} R(x_n, x_1, x_2) & \text{if } i = 1 \\ R(x_{n-1}, x_n, x_1) & \text{if } i = n \\ R(x_{i-1}, x_i, x_{i+1}) & \text{if } i \neq 1,\ n \end{cases}$$

2.1 Example If we consider the majority rule R defined by the table

x	R
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

then if

$$x = (01100101)$$

we have

$$Fx = (11100010)$$ □

We require to find the fixed points of F for any given rule R, i.e., the points which is defined by

$$x(k+1) = F(x(k)) \quad (k \geq 1) \tag{2.2}$$

and F is given by a local rule of order 3. Thus, $(Fx_n)_i = y_i$, where

$$y_i = \begin{cases} R(x_n, x_1, x_2) & \text{if } i = 1 \\ R(x_{n-1}, x_n, x_1) & \text{if } i = n \\ R(x_{i-1}, x_i, x_{i+1}) & \text{if } i \neq 1, n \end{cases}$$

2.1 Example If we consider the majority rule R defined by the table

x	R
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

then if

$$x = (01100101)$$

we have

$$Fx = (11100010)$$ □

We require to find the fixed points of F for any given rule R, i.e., the points

$B_i = x_{i-1}x_ix_{i+1}, \quad i \neq 1, \; n$

(i.e. we use periodic boundary conditions). Conversely, any such sequence for which the last bit of the strings B_i are the same as the first bit of B_{i+1} (for all i) defines a state x in the obvious way. This idea of overlapping strings is formalized in the following definition.

2.2 Definition We shall say that for two elements $B_1 = (\beta_1, \beta_2, \beta_3)$ and $B_2 = (\gamma_1, \gamma_2, \gamma_3) \in R_{\mathcal{F}}$ we may **put** B_2 to the **right** of B_1 if

$$\beta_i = \gamma_{i-1} \quad i = 2, \; 3.$$

We also say that B_1 **can be put to the left of** B_2. We next form a directed graph G with vertices $\{v_i\}$ which are the elements of $R_{\mathcal{F}}$. If $v_1, v_2 \in R_{\mathcal{F}}$ then the graph contains the directed edge (v_1, v_2) if and only if v_2 can be put to the right of v_1 (or equivalently, if v_1 can be put to the left of v_2). As above, let a state x be written in the form (2.1) and associated with it the sequence of binary strings

$$\{B_1, B_2, \ldots B_n\}$$

as above where each $B_i \in R_{\mathcal{F}}, \; i \in \; n$. Clearly the subgraph containing the vertices B_i is connected and so we can restrict attention to connected subgraphs of G.

2.3 Definition A cycle of length K in a graph G is a sequence of vertices $v_1v_2v_3 \ldots v_Kv_1$ such that $v_Kv_1 \in E$ and $v_iv_{i+1} \in E, \, i \neq K$. □

2.4 Theorem If $n > 3$ then an n-dimensional system has a fixed point if and only if G has a cycle of length n.

Proof This is clear since if v_i is the vertex of G associated with B_i, then $v_1v_2 \ldots v_nv_1$ is a cycle in G of length n. □

2.5 Definition The adjacency or **transfer matrix** $S = (s_{ij}) \; 1 \leq i, j \leq n$, where $n = \#(V)$, of a graph G=(V,E) is defined by

$$s_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in E \\ 0 & \text{otherwise} \end{cases}$$

The following result is proved in [7].

2.6 Theorem The number of cycles of length n in a graph G is given by $tr(S)$, where S is the adjacency matrix of G and tr(A) denotes the trace of the matrix A. □

Combining theorems 2.4 and 2.6 we immediately obtain the remarkable result below; valid for any rule.

2.7 Theorem Let R be any 3-bit rule, let $R_{\mathcal{F}}$ be the fixed set as above and let G=(V,E) be the graph associated with R as before (where $\#(V) = \#(R_{\mathcal{F}})$). If S is the adjacency matrix of G, then the number of fixed points of the dynamical system (2.2) is given by $tr(S)$. □

2.8 Remark (a) Note that the assumption of a 3-bit rule is unecessary. The result is true for a p-bit rule for any $p > 2$.

(b) From [3] we have

$$tr(S) = \phi_n + \hat{\phi}_n + (-1)^{[(n+1)/3]}(2 - [nmod3/2]) \qquad (2.3)$$

where $[x]$ is the largest integer $\leq x$, $\phi = (1+\sqrt{5})/2$ and $\hat{\phi} = 1 - \phi$.

2.9 Example Consider again the majority rule of exampl 2.1. The graph associated with this rule is shown in figure2.1. For example, if n=5 the cycles are (using periodic boundary conditions):

$v_1 v_1 v_1 v_1 v_1 v_1$ 00000

$$v_1v_2v_3v_5v_6v_1 \quad 00110$$

$$v_2v_3v_4v_5v_6v_2 \quad 01110$$

$$v_2v_3v_5v_6v_1v_2 \quad 01100$$

$$v_3v_5v_6v_1v_2v_3 \quad 11000$$

$$v_3v_4v_5v_6v_2v_3 \quad 11100$$

$$v_6v_1v_2v_3v_5v_6 \quad 00011$$

$$v_6v_2v_3v_4v_5v_6 \quad 00111$$

$$v_5v_6v_1v_2v_3v_5 \quad 10001$$

$$v_5v_6v_1v_2v_3v_5 \quad 10011$$

$$v_4v_5v_6v_2v_3v_4 \quad 11001$$

$$v_4v_4v_4v_4v_4v_4 \quad 11111.$$

Note that there are 12 in agreement with (2.3).

3 Two-Dimensional Systems In this section we shall show that the one dimensional results obtained above can be easily generalized to the two-dimensional case. Here we shall consider the case of a 9-bit rule which determines a new value for a given pixel in terms of its old value and the values of its eight nearest horizontal and vertical neighbours fig.3.1. Also, we shall restrict attention to periodic boundary conditions.

We can write

$$b_0' = R(b_1b_2b_3b_4b_5b_6b_7b_8b_9)$$

3.1 Definition The fixed point set $R_{\mathcal{F}}$ of the rule R is the subset of $\{0,1,\ldots,512\}$ consisting of numbers $K = b_1b_2b_3b_4b_5b_6b_7b_8b_9$ for which $R(K) = b_5$.

3.2 Definition If $K_1 = b_1b_2b_3b_4b_5b_6b_7b_8b_9$ and $K_2 = c_1c_2c_3c_4c_5c_6c_7c_8c_9$ are two binary representations of a nine-bit neighbourhood then we say that K_1 can be **put above** (respectively **below, to the left of, to the right of**) K_2 if

$b_4 = c_1, b_5 = c_2, b_6 = c_3, b_7 = c_4, b_8 = c_5, b_9 = c_6$

(resp. $b_4 = c_1, b_5 = c_2, b_6 = c_3, b_7 = c_4, b_8 = c_5, b_9 = c_6$;

$b_2 = c_1, b_3 = c_2, b_5 = c_4, b_6 = c_5, b_8 = c_7, b_9 = c_8$;

$b_1 = c_2, b_2 = c_3, b_4 = c_5, b_5 = c_6, b_7 = c_8, b_8 = c_9$;)

In contrast to the one-dimensional case we now form two directed graphs G_{UD}, G_{RL} each containing the vertices $R_{\mathcal{F}}$ and such that G_{UD} contains an edge (v_1, v_2) (for $v_1, v_2 \in R_{\mathcal{F}}$) if and only if v_2 can be put above v_1 and G_{RL} contains an edge (v_1, v_2) if and only if v_2 can be put to the right of v_1. Suppose our state vector is of the form

$$x = (x_{ij}), \quad 1 \leq i, j \leq K$$

We will show how to determine all K-length cycles in G_{RL}. These are finite in number and we write

$$C_{RL}^K = \{c : \quad c \text{ } is \text{ } a \text{ } K - length \text{ } cycle \text{ } in \text{ } G_{RL}\}$$

for the set of such K-length cycles. Now form a new graph $\mathcal{G}_{RL}$ with vertices in a one to one correspondence with C_{RL}^K. Two vertices c_1 and c_2 in C_{RL}^K will be joined by directed edge (and we say that c_2 can be put above c_1) if the following holds: suppose that c_1 and c_2 represent the K-length cycles

$$c_1 = v_1 \ldots v_K$$

$$c_2 = w_1 \ldots w_K$$

and that c_1 can be cyclically permuted to obtain

$$c_1' = v_i v_{i+1} \dots v_K v_1 \dots v_{i-1}$$

so that w_j can be put above:

$$\begin{cases} v_{j+i-1} & \text{if } j+i-1 \leq k \\ v_{j+1-1-k} & \text{if } j+i-1 > k. \end{cases}$$

3.3 Theorem A KxK two-dimensional system has a fixed point if and only if $\mathcal{G}_{RL}$ has a K-length cycle.

Proof This follows inexactly the same way as theorem 2.10. □

3.4 Remark We could also define the graph $\mathcal{G}_{UD}$ in an obvious way.

3.5 Example We shall illustrate the above theory with the game of life. The rule in this case is given in the following way. The set $R_{\mathcal{F}}$ can be shown to contain 284 nine-bit strings. The following are examples

$$\{000111001, 000111010, 000111100\}.$$

Then all horizontal and vertical cycles can be determined, and an algorithm has been implemented on a Sun workstation using the adjacency matrix of a graph where we have taken $K = 15$. This will find all fixed points for any given K. One particular solution is shown in fig. 3.2.

4 Conclusions A simple algorithm which finds all fixed points and their number of one or two dimensional systems. A technique has been given for the determination of fixed point in one and two-dimensional cellular automata. It is specified in terms of graph theory and provides an easily computable method in both cases. The algorithm has been applied to find fixed points of one- and two-dimensional cellular automata.

References

[1] Theory and Applications of Cellular Automata by S. Wolfram, World Scientific, Singapore, 1986.

[2] Fixed Points of Majority Rule Cellular Automata with its Applications to Plasticity and Precision of the Immune System by Z. Agur, Complex Systems, 5, 1991, 351-357.

[3] Number of Fixed Points of the Majority Rule by Z. Agur, A.S. Frankel, and S.T. Klein, Discrete Mathematics, 70, 1988, 295-302.

[4] Determination of Fixed Points and Shift Cycles for Nearest Neighbour Cellular Automata by B. Voorhees, J.S. P., Vol. 66, 1992.

[5] Nearest Neighbour Cellular Automata over Z_2 with periodic Boundary conditions by b. Voorhees,Physica D, 26-35, 1990.

[6] Computation Theory of Cellular Automata by S. Wolfram, Commun. Math. Phys. 96, 15-57, 1984

[7] Enumerative Combinatorics by R.P. Stanley, Vol. 1, The Wadsworth - Brooks/Cole Mathematics series, 1986.

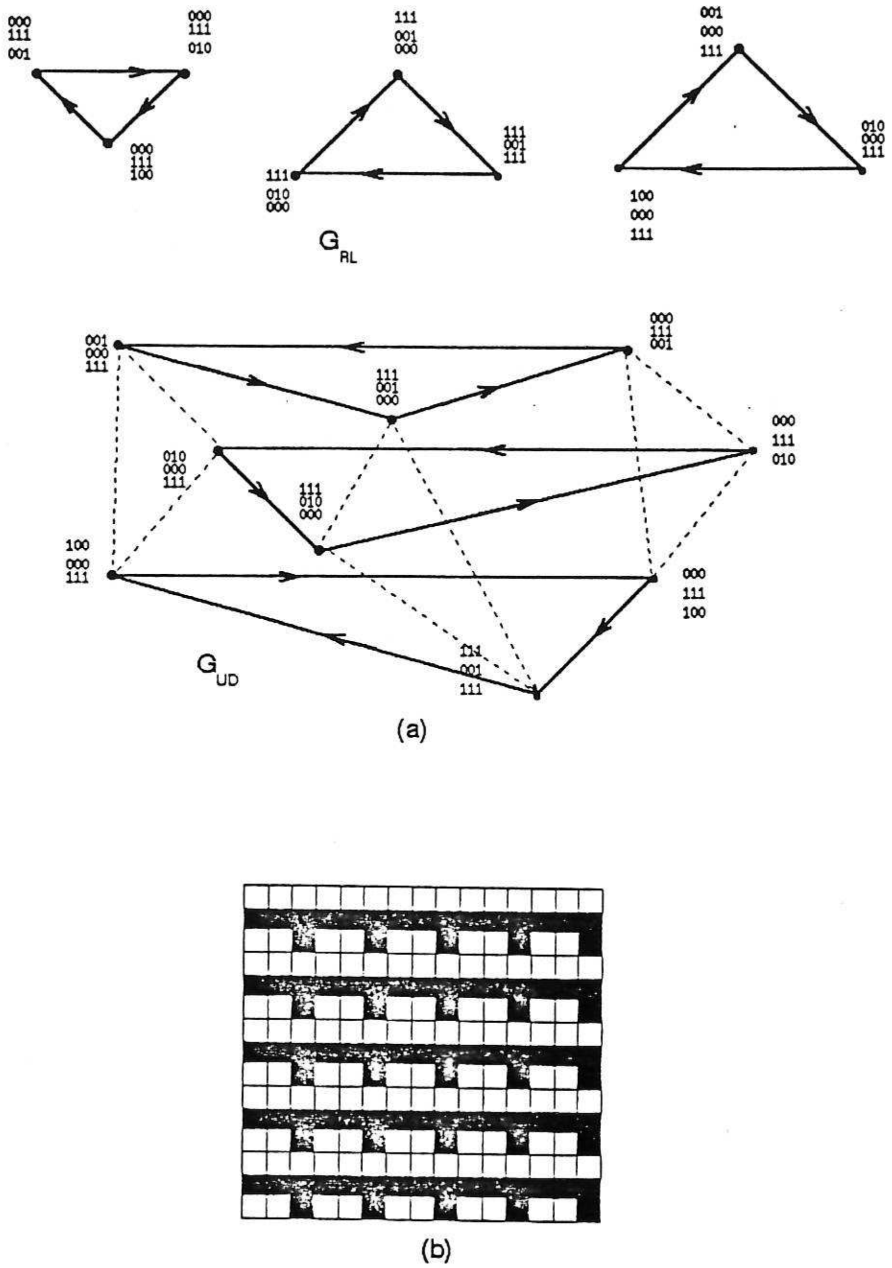

Fig.3.2. Particular Solution For K=15

(a) Graphs G_{UD} and G_{RL} (b) picture (15x15)

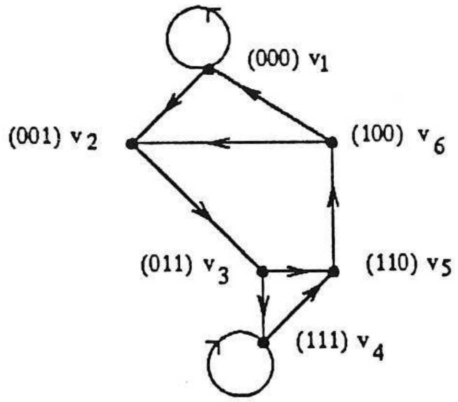

Figure 2.1. Graph of the Majority Rule.

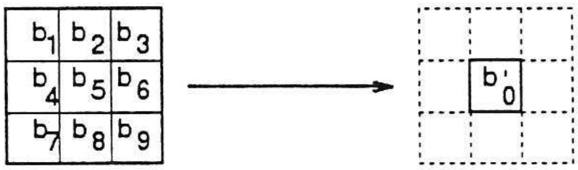

Figure 3.1. Neighbourhood Structure for 9-bit Rule.