This is a repository copy of *Fixed Points and Shift Cycles in Cellular Automata*.

White Rose Research Online URL for this paper: http://eprints.whiterose.ac.uk/79286/

---

**Monograph:**

---

**Reuse**
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

**Takedown**
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.
Fixed Points and Shift Cycles in Cellular Automata

by

S. P. Banks and S. Djirar

Department of Automatic Control and Systems Engineering
University of Sheffield
Mappin Street
SHEFFIELD S1 4DU

Research Report No. 459
September 1992
Abstract

A technique for determining fixed points and shift cycles 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 (and shift cycles) in one- and two-dimensional cellular automata [1,3]. The one dimensional case has recently been considered in [2] where certain operators are constructed to determine the fixed points. The method, however, is complicated and is difficult to generalise to two-dimensions. Here we give a very simple technique which uses the theory of graphs and applies to both one and two dimensional systems. The method produces an easily computable result which can be implemented on a computer.

Graphs as finite state machines have been used in the computation theory of cellular automata [4], but not previously for detecting fixed points. In section 2 we consider one-dimensional problems for a rule of any length and section 3 we consider the case of a 5 bit two-dimensional rule with periodic boundary conditions.

2 One Dimensional Systems

We shall first consider systems of doubly infinite length for simplicity. Thus we consider a dynamical system with a binary state vector of the form

\[ x = (\ldots, x_{-3}, x_{-2}, x_{-1}, x_0, x_1, x_2, x_3, \ldots), \quad x \in \mathbb{Z}_2 \]

which is defined

\[ x(n + 1) = F(x(n)) \]
and $F$ is given by a local rule of order $p$ (odd). Thus,

$$(Fx(n))_i = x_i(n+1)$$

$$= R(x_{i-\lfloor p/2\rfloor}(n), x_{i-\lfloor p/2\rfloor+1}(n), \ldots, x_i(n), \ldots, x_{i+\lfloor p/2\rfloor}(n))$$

For example if $p=3$ consider the rule $R$ defined by truth table:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
</tr>
<tr>
<td>111</td>
<td>0</td>
</tr>
</tbody>
</table>

Then if

$$x = (\ldots0001100101110001000\ldots)$$

(with leading and trailing zeros) we have

$$Fx = (\ldots001111111011111100\ldots)$$

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

$$x = Fx,$$  \hspace{1cm} (2.1)
A p-bit rule assigns a binary bit to each p-bit binary number which can be represented by its equivalent natural number. Thus a p-bit rule $R$ is a map

$$R : 2^p = \{0, 1, 2, \ldots, 2^p - 1\} \rightarrow \{0, 1\}$$

In the above example, $R$ is the 3-bit rule given by

$$R(0) = 0, R(1) = 1, R(2) = 1, \ldots, R(7) = 0.$$  

2.1 Definition The fixed point set $R_\mathcal{F}$ of the rule $R$ is the subset of $\{0, 1, 2, \ldots, 2^p - 1\}$ consisting of all numbers whose central binary bit is fixed by $R$. Thus, if

$$K = b_1b_2\ldots b_{x+1}\ldots b_p \in 2^p$$

then $K$ is a fixed point of $R$ if

$$R(K) = b_{x+1}$$

Again in the above example, $R_\mathcal{F} = \{0, 2, 3, 6\}$.

Consider the state $x$ to be made up of successive strings of p-bit binary numbers:

$$x = \ldots b_{-2}b_{-1}b_0b_1b_2\ldots$$

where each $b_i \in 2^p$ and is to be considered as being written in binary form. In order that $x$ be a fixed point of $F$ it is clearly necessary that

$$b_i \in R_\mathcal{F}, i \in \mathbb{Z}.$$
This is obviously not sufficient, however, since substrings of \( b_i b_{i+1} \) may not be in \( R_\mathcal{F} \).

**2.2 Definition** We shall say that for two elements \( b_1 = (\beta_1, \ldots, \beta_p) \) and \( b_2 = (\gamma_1, \ldots, \gamma_p) \in R_\mathcal{F} \), we may put \( b_2 \) to the right of \( b_1 \) if

\[
\beta_i = \gamma_{i-1}, \quad 2 \leq i \leq p.
\]

We also say that \( b_1 \) can be put to the left of \( b_2 \). We next form a directed graph \( G \) with vertices 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 \( v_2 \)). We shall write \( V \) (or \( V_G \)) for the vertices of \( G \) and by \( E \) (or \( E_G \)) the edges of \( G \). As above, let a state \( x \) be written in the form

\[
x = \ldots b_{-2} b_{-1} b_0 b_1 b_2 \ldots
\]

where each \( b_i \in 2^p \) is \( p \)-bit binary string and \( b_i \in R_\mathcal{F}, i \in \mathbb{Z} \).

**2.3 Lemma** If the state \( x \) is a fixed point of \( F \) then the set

\[
B_x = \{b_i : i \in \mathbb{Z}\} \subseteq 2^p
\]

is a connected subgraph of \( G \).

**Proof** If \( b_k \) and \( b_\ell \) are in different connected subgraphs of \( G \) and \( k < \ell \) consider the subsequence

\[
b_k b_{k+1} \ldots b_\ell
\]
on $x$. Since

$$(b_i, b_{i+1}) \in E_G$$

for $k \leq i < \ell$ we have a contradiction. \hfill \Box

It follows from lemma 2.3 that we can restrict attention to connected subgraphs of $G$.

2.4 Lemma If $(v_1, v_2, v_3, \ldots, v_n, v_1)$ is a circuit in $G$ then

$$x = (v_1 v_2 \ldots v_n v_1 v_2 \ldots v_n v_1 v_2 \ldots)$$

is a fixed point of $F$.

Proof The proof is trivial. \hfill \Box

We now describe an algorithm which reduces the graph $G$ to a tree from which all possible fixed points $x$ can be determined. Let $e$ be any edge in $G$ which is on a loop and let $M(e)$ denote the maximal connected subgraph of $G$ containing $e$ such that every vertex of $M(e)$ is on a loop. Clearly we have

$$M(e) = M(e_1)$$

if and only if $e$ and $e_1$ are on a loop so that

$$e, e_1 \in M(e), M(e_1).$$

Hence $M(e)$ is independent of the choice of $e$ in $M(e)$. Otherwise $M(e)$ and $M(e)$ are disjoint if $\overline{e} \not\in M(e)$. Let $\overline{G}$ be the graph obtained from $G$ by shrinking each subgraph $M(e)$ to a point and regarding it as a vertex of $\overline{G}$. All other vertices
and edges in $G$ remain unchanged.

2.5 Example Consider the graph $G$ if fig. 2.1.

Then $\overline{G}$ is the graph in fig. 2.2.

2.6 Lemma For any directed graph $G$, $\overline{G}$ is tree.

Proof Suppose that $(\overline{v_1}, \overline{v_2}), (\overline{v_2}, \overline{v_3}), \ldots, (\overline{v_L}, \overline{v_1})$ is a circuit in $\overline{G}$. Each vertex $\overline{v_i}$ in $\overline{G}$ corresponds to a (nonunique) vertex $v_i$ in $G$. Then $(v_1, v_2), \ldots, (v_L, v_1)$ is a circuit in $G$ contradicting the definition of the vertices of $\overline{G}$. $\square$

2.7 Theorem Denote by $V_1 \subseteq \overline{V}$ the vertices in $\overline{G}$ which are obtained by shrinking a maximal connected set of circuits as described above. Consider the set of all paths in $\overline{G}$ and ending in $V_1$. These are clearly finite in length and finite in number. Then any fixed point of $F$ is given by

$$z = s_1 v_{11} \ldots v_{1K_1} \ldots s_2 v_{21} \ldots v_{2K_2} \ldots s_L v_{L1} \ldots v_{LK_L} s_{L+1}$$

where $L$ is the number of edges in the path, $v_{ij}$ are vertices of $G$ and $s_1, \ldots, s_{L+1}$ are strings obtained from the maximal circuit subgraphs corresponding to vertices of $V_1$ along the path. Note that $s_1, s_{L+1}$ are infinite strings while $s_2, \ldots, s_L$ are finite.

Proof The proof is trivial from the definition of $\overline{G}$. $\square$

In order to determine the structure of the strings $s_i$ in theorem 2.7 in more detail we introduce the following terminology. In $G$ consider a maximal circuit subgraph $C$ and let $V_C \subseteq V$ be the vertices of $G$ in $C$. Suppose that $v \in V$ and $v \notin V_C$, but the edge $(v, v_1) \in E_G$ for some $v_1 \in V_C$. Then $v_1$ is called an entry point of $C$. Similarly we can define an exit point of $C$ in the corresponding
way.

Clearly, each of the strings $s_1, \ldots, s_L$ must start and finish with an entry point and an exit point. Similarly, $s_1$ must end with an exit point and $s_{L+1}$ must start with an entry point.

We shall say that a vertex $\bar{v} \in \bar{V}$ is a peripheral in $\bar{G}$ if it has no entry point which is connected to another element of $\bar{V}$ or no exit point similarly connected. Then $s_1$ and $s_{L+1}$ can be peripheral (although not necessarily) and $s_2, \ldots, s_L$ cannot. Within $s_2, \ldots, s_L$ we can have any path leading from an entry point to an exit point possibly containing an arbitrary number of loops. Similar remarks apply to $s_1$ and $s_{L+1}$. We therefore see that all the fixed points of $F$ can be read from $\bar{G}$ and $G$.

2.8 Example We shall determine all the fixed points associated with the graph in fig.1. From the above results we clearly can have fixed points of only four types:

$$s_1 v_0 s_2$$

where $s_1$ is an infinite string in $\bar{v}_1$ or $\bar{v}_4$ and $s_2$ is an infinite string in $\bar{v}_2$ or $\bar{v}_3$.

Note that fixed points cannot contain $v_5$ or $v_7$. There is only one infinite string in $\bar{v}_1$, namely

$$\cdots v_{13} v_{11} v_{12} v_{13} v_{11} v_{12} \cdots$$

Note that $v_{12}$ is an exit point for $\bar{v}_1$. Similarly, $\bar{v}_4$ has only one infinite string, i.e.

$$\cdots v_{42} v_{41} v_{42} v_{41}$$
with \( v_{41} \) as an exit point. The vertex \( \bar{v}_3 \) also has only one string with \( v_{31} \) as an entry point:

\[
v_{31}v_{32}v_{33}v_{34}v_{31}v_{32} \cdots
\]

Finally, \( \bar{v}_2 \) has an infinite number of strings with entry point \( v_{21} \). The most obvious one is

\[
v_{21}v_{22}v_{23}v_{24}v_{25}v_{21}v_{22}v_{23} \cdots
\]

However at any point \( v_{22} \) along this string we can insert the loop \( v_{22}v_{24}v_{27}v_{22} \) any number of times. Hence all the strings in \( \bar{v}_2 \) are of the form

\[
v_{21}v'_{22}v_{23}v_{24}v_{25}v_{21}v'_{22}v_{23} \cdots
\]

where \( v'_{22} = v_{22} \) or \( v'_{22} = v_{22}v_{24}v_{27}v_{22}v_{24}v_{27} \cdots v_{22} \) and similarly for \( v''_{22} \), etc.

**2.9 Example** As a concrete example consider the five-bit rule with fixed point set \( R_F \) given by

\[
R_F = \{00000, 00011, 00100, 00101, 00110, 01000, 01001, 01010, 01011, 01100, 10010, 10011, 11001, 11010, 11011, 11111\}
\]

Then \( G \) is the graph in fig. 2.3.

Thus, a fixed point is of one of the forms:

\[
\cdots 010010010011001100110011001 \cdots
\]

\[
\cdots 000000000 \cdots
\]

\[
\cdots 11111111 \cdots
\]
Consider next the case of finite dimensional dynamical systems with the vector

$$\mathbf{z} = (z_1, z_2, \ldots, z_K)$$

with periodic boundary conditions. We can form the graphs $G$ and $\bar{G}$ just as before and we obtain the following theorem.

2.10 Theorem If $K > p$ then the system has a fixed point if and only if $G$ has a cycle of length $K$. (The length of a cycle $(v_1v_2v_3 \cdots v_mv_1)$ in $G$ is $m$.)

Proof Since we have periodic boundary conditions, if

$$\mathbf{z} = (z_1, z_2, \ldots, z_K)$$

is a fixed point then so is

$$z_1z_2 \cdots z_p \cdots z_Kz_1z_2 \cdots z_p \cdots$$

The result is now obvious. □

2.11 Example Consider the 5-bit system of example 2.9. Clearly arbitrary dimensional systems have fixed points containing just 0's or 1's respectively since $G$ has cycles of arbitrary length in these vertices. However the only other cycles have length 3 and 4. Hence only systems of dimensions $3m$ and $4m$ for $m \geq 2$ will have fixed points. For example, 001001001001001001 is a fixed point of an 18-dimensional system while 10011001 is a fixed point of an 8-dimensional system.

Note finally that shift cycles can be treated in exactly the same way as fixed points if we replace the elements the set $R_x$ by the set $R_C$ given by the elements
of $R$ which satisfy then property

$$R(b_1 b_2 \ldots b_p) = b_i, \ 1 \leq i \leq p.$$  

This will give a shift of magnitude $|\frac{i+1}{2} - i|$.

## 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. For simplicity, we shall consider only the case of a five-bit rule which determines a new value for a given pixel in terms of its old value and the values of its four nearest horizontal and vertical neighbours (fig. 3.1). Also, we shall restrict attention to periodic boundary conditions.

We can write

$$b'_0 = R(b_1 b_2 b_3 b_4 b_5)$$

### 3.1 Definition

The fixed point set $R_x$ of the rule $R$ is the subset of $\{0, 1, \ldots, 31\}$ consisting of numbers $K = b_1 b_2 b_3 b_4 b_5$ for which $R(K) = b_3$.

**bf 3.2 Definition** If $K_1 = b_1 b_2 b_3 b_4 b_5$ and $K_2 = c_1 c_2 c_3 c_4 c_5$ are two binary representations of a five-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

$$b_3 = c_1, \ b_5 = c_3$$

(resp. $b_1 = c_3, b_3 = c_5; c_4 = b_3, c_3 = b_2; b_4 = c_3, b_2 = c_3$)
In contrast to the one-dimensional case we now form two directed graphs \( G_{UD}, G_{RL} \) each containing the vertices \( R \) and such that \( G_{UD} \) contains an edge \((v_1, v_2)\) (for \( v_1, v_2 \in R \)) 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

\[
z = (z_{ij}), \quad 1 \leq i, j \leq K.
\]

Determine all \( K \)-length cycles in \( G_{RL} \). These are finite in number and we write

\[
C^K_{RL} = \{ c : c \text{ is a } K\text{-length cycle in } G_{RL} \}
\]

for the set of such \( K \)-length cycles. Now form a new graph \( G_{RL} \) with vertices in a one to one correspondence with \( C^K_{RL} \). Two vertices \( c_1 \) and \( c_2 \) in \( C^K_{RL} \) will be joined by a 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

\[
c_1 = v_1 \cdots v_K
\]

\[
c_2 = w_1 \cdots w_K
\]

and that \( c_1 \) can be cyclically permuted to obtain

\[
c'_1 = v_{i}v_{i+1}\cdots v_Kv_1\cdots v_{i-1}
\]

so that

\[
w_j \text{ can be put above } \begin{cases} v_{j+i-1} & \text{if } j+i-1 \leq K \\ v_{j+i-1-K} & \text{if } j+i-1 > K \end{cases}
\]
3.3 Theorem A $K \times K$ two-dimensional system has a fixed point if and only if $G_{RL}$ has a $K$-length cycle.

Proof This follows in exactly the same way as theorem 2.10. 

3.4 Remark We could also define the graph $G_{UD}$ in an obvious way.

3.5 Example We shall illustrate the above theory with a simple five-bit rule.

The rule in this case is given in the following way:

We shall represent the cells surrounding a given cell $c$ as follows:

\[
\begin{array}{ccc}
  a \\
 d & c & b \\
  e
\end{array}
\]

and the rule is defined on such a set of the cells by

\[ R(abcde) = c' \]

where $R$ is given fully in fig. 3.2.

If

\[ S = \{(abcde) : R(abcde) = c\} \]

then we clearly have

\[ S = \{00000, 00001, 00010, 00111, 01000, 01101, 10000, 10110, 11100\}. \]

First we form $G_{RL}$ as above. This gives the graph in fig. 3.3. Number the vertices $v_1 \cdots v_9$ as above. Suppose we wish to find periodic fixed points in a

13
10 \times 10 'image'. We must first determine all cycles of length 10 in \( G_{RL} \). These can be found, in general, by computer from the incidence matrix of the graph, but here we can read them off quite easily. For simplicity and for the purposes of illustration we shall only determine a small part of \( G_{RL} \). Thus, consider the following 10-bit cycles in \( G_{RL} \):

\[
\begin{align*}
v_1 &= u_1 v_1 v_1 v_1 v_1 \cdots v_1 \\
v_2 &= u_1 v_1 v_1 v_2 v_2 v_2 \cdots v_1 \\
v_3 &= u_1 v_1 v_2 v_3 v_4 v_5 v_6 \cdots v_1 \\
v_4 &= u_1 v_1 v_2 v_3 v_7 v_8 v_9 \cdots v_1 \\
v_5 &= u_1 v_1 v_1 v_9 v_9 v_1 v_1 \cdots v_1
\end{align*}
\]

By considering \( G_{UD} \) it is easy to see that

\[
v_1 v_1 v_2 v_3 v_4 v_5 v_1 v_1 v_1 v_1 v_1 v_1 v_1 v_1
\]

is a length-10 cycle in \( G_{RL} \). This cycle corresponds to the fixed point shown in fig. 3.4.

4 Conclusions

A simple technique has been given for the determination of fixed point (and shift cycles) in one and two-dimensional cellular automata. It is specified in terms of graph theory and provides an easily computable method in both cases. Since a
limit cycle is a fixed point of a rule applied several times we anticipate that the technique will also be useful in finding limit cycles. This will be examined in a future paper.

References


Fig. 2.1. A Simple Directed Graph
Fig. 2.2. Simplified Graph of the Graph G in Fig. 2.1.

Fig. 2.3. Graph of a simple 5-bit Rule
Fig. 3.1. Neighbourhood Structure for a 5-bit Rule

<table>
<thead>
<tr>
<th>a b c d e</th>
<th>c'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 1 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>0 0 1 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1 0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1 1 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1 1 1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0 1 0</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 1 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 1 1 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1 1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 0 1 0</td>
<td>1</td>
</tr>
<tr>
<td>1 0 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 1 0 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1 0 1</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>1 0 1 1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 1 0 0</td>
<td>1</td>
</tr>
<tr>
<td>1 1 1 0 1</td>
<td>0</td>
</tr>
<tr>
<td>1 1 1 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 1 1 1 1</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 3.2. A Simple 5-Bit Rule
Fig. 3.3. The Graph $G_{RL}$ for a Simple 2-Dimensional System

Fig. 3.4. A Two-Dimensional Fixed Point