# Drawdown Automata, Part 2:
# Neighborhoods and State-Transition Rules

In the first article on drawdown automata [1], we described basic concepts and how cellular automata can be used to produce drawdown patterns. In this article, we'll look more closely at neighborhoods and state-transition rules.

## Neighborhoods

The neighborhood of a cell, called the *core cell*, consists of the core cell and those surrounding cells whose states determine the next state of the core cell. The term neighborhood implies proximity. Every cell in a cellular automaton has the same kind of neighborhood.

Figures 1 and 2 show the most commonly used neighborhoods. Core cells in neighborhood diagrams are outlined to emphasize their location.

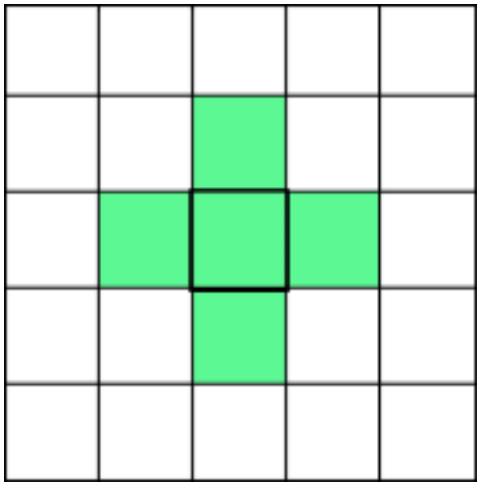

**Figure 1. Von Neumann Neighborhood**

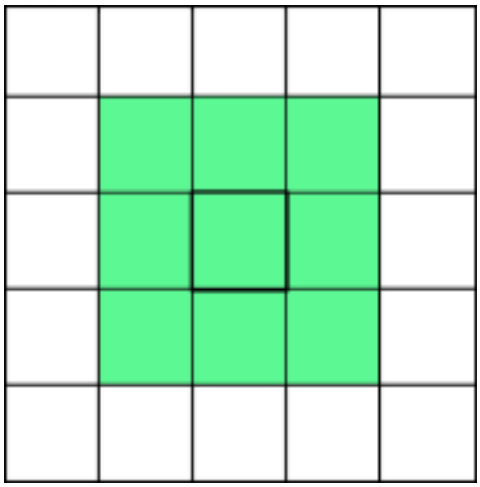

**Figure 2. Moore Neighborhood**

Many other neighborhoods are possible. Examples are shown in Figures 3-9.

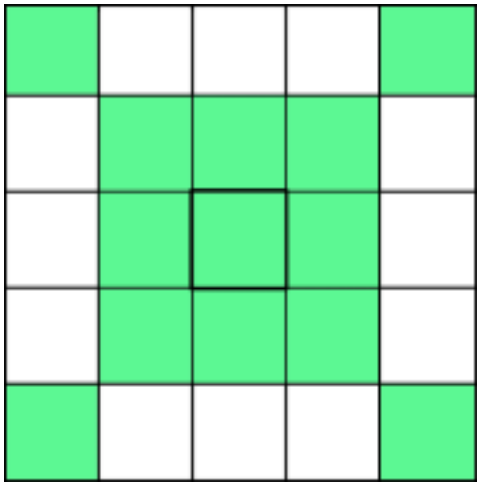

**Figure 3. Moore Neighborhood with "Ears"**



February 13, 2002; last revised August 1. 2004

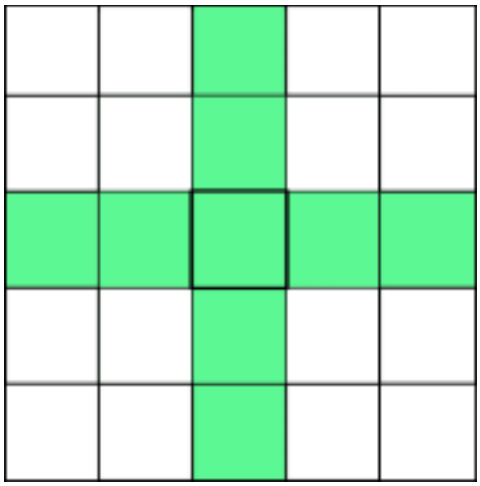

**Figure 4. 9-Cell Cross**

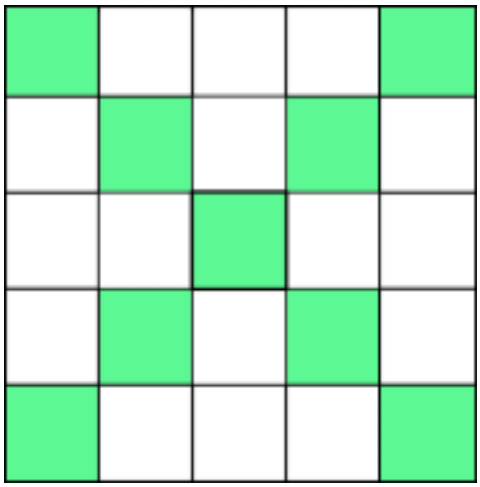

**Figure 5. 9-Cell Star**

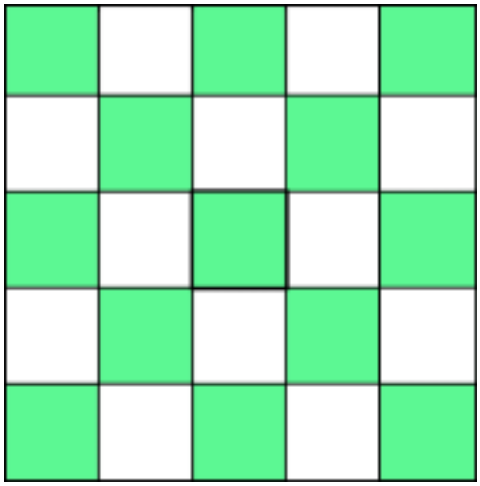

**Figure 6. 13-Cell Checkerboard**

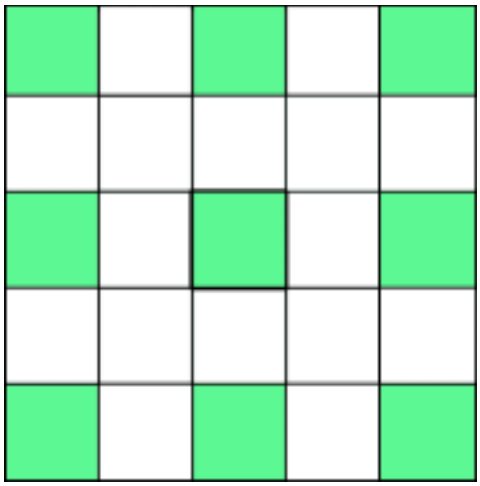

**Figure 7. 9-Cell Spread**

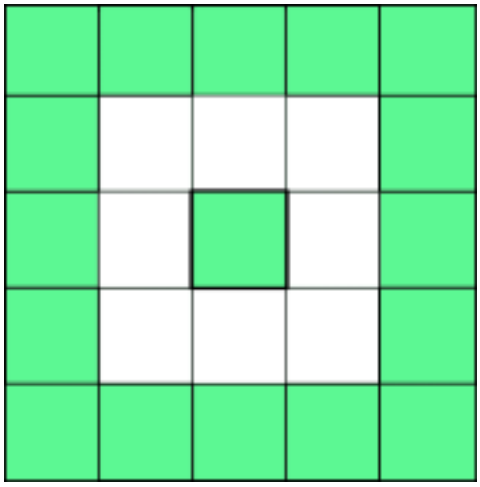

**Figure 8. 17-Cell Compound**

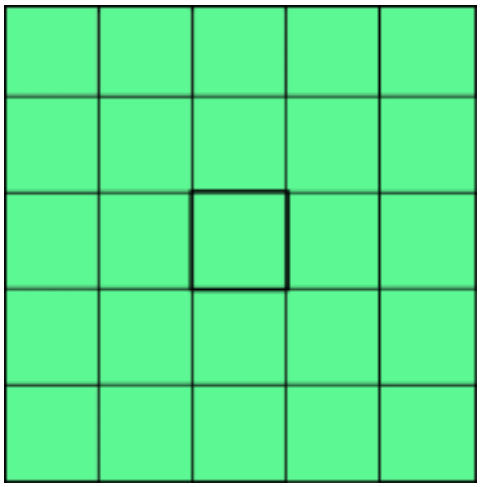

**Figure 9. Extended Moore Neighborhood**

All the neighborhoods shown so far are symmetric with respect to rotation. This need not be the case. See Figures 10-12.

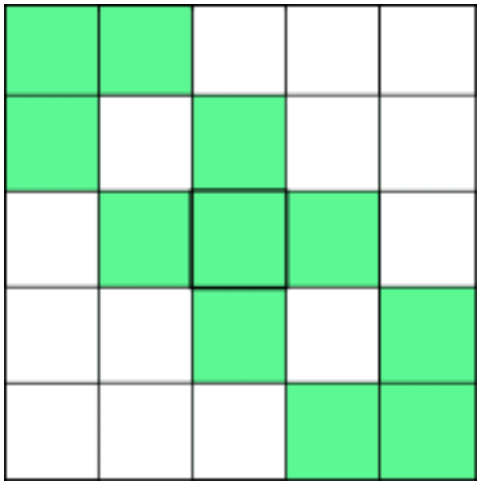

**Figure 10. 11-Cell Diagonal**

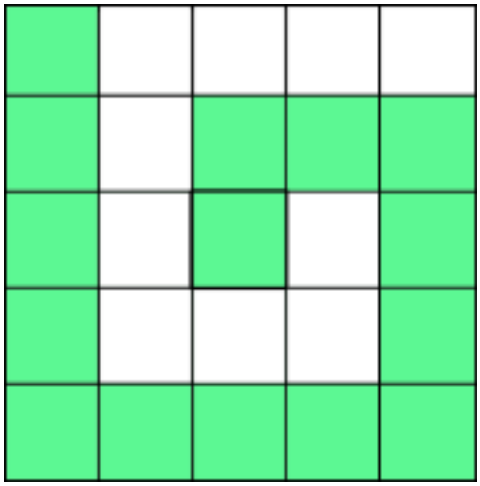

**Figure 11. 15-Cell Spiral**

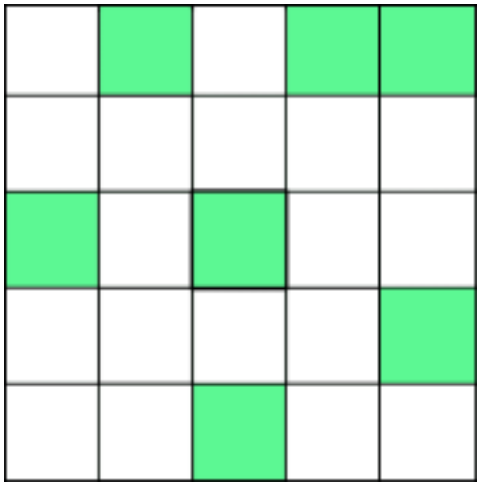

**Figure 12. 7-Cell Scatter**

The *radius* of a neighborhood is defined to be the maximum distance from the core cell, horizontally or vertically, to cells in the neighborhood. The neighborhoods in Figures 1 and 2 have radius 1. The other neighborhoods have radius 2.

The number of cells in a neighborhood, their relative positions, and radius of the neighborhood all figure into the possibilities for pattern generation.

Given what is possible even with the von Neumann and Moore neighborhoods, the use of other neighborhoods needs to be justified. We'll look into this in a subsequent article.

## State-Transition Rules

A state-transition rule specifies how the state of a core cell changes as a function of the states of the cells in its neighborhood.

Even for small neighborhoods, the number of possible rules is very large. For the 5-cell von Neumann neighborhood, there are more than four billion possible rules.

### General Rules

In general, each cell in the neighbor can figure into a state-transition rule independently of all other cells. This is illustrated in Figures 13 and 14 for the von Neumann and Moore neighborhoods, where each cell is shown in a different color.

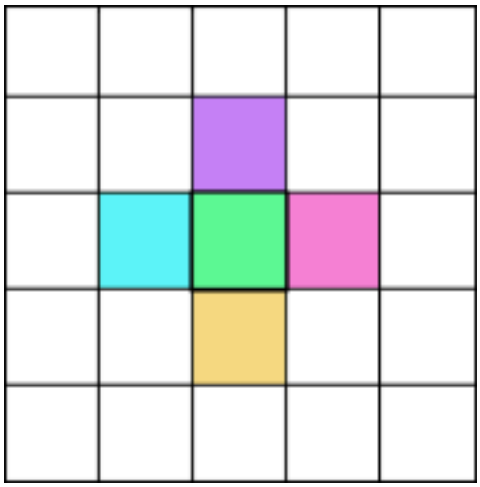

**Figure 13. Von Neumann General Rules**

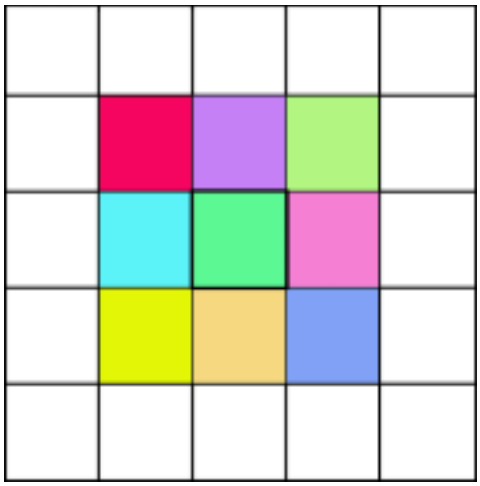

**Figure 14. Moore General Rules**

A general rule can be described by a table whose rows show all possible combinations of the states of individual cells along with the corresponding new state for the core cell.

For example, the parity rule for the von Neumann neighborhood, for which the new state of the core cell is 1 if the sum of all the cells in the neighborhood is odd but 0 otherwise, has the following table. The cells are identified by position as in Reference 1 and C' is the new state of the core cell.

| NESWC | C' | NESWC | C' |
|---|---|---|---|
| 00000 | 0 | 10000 | 1 |
| 00001 | 1 | 10001 | 0 |
| 00010 | 1 | 10010 | 0 |
| 00011 | 0 | 10011 | 1 |
| 00100 | 1 | 10100 | 0 |
| 00101 | 0 | 10101 | 1 |
| 00110 | 0 | 10110 | 1 |
| 00111 | 1 | 10111 | 0 |
| 01000 | 1 | 11000 | 0 |
| 01001 | 0 | 11001 | 1 |
| 01010 | 0 | 11010 | 1 |
| 01011 | 1 | 11011 | 0 |
| 01100 | 0 | 11100 | 0 |
| 01101 | 1 | 11101 | 0 |
| 01110 | 1 | 11110 | 0 |
| 01111 | 0 | 11111 | 1 |

Since there are five cells in the neighborhood, there are $2^5$ rows in the table. Notice that the values for NESWC are given in increasing order, considered as base-2 integers. For each of the rows, there are 2 possibilities for C'. Thus there are $2^{32}$ = 4,294,967,296 possible rules.

For a 9-cell neighborhood, rule tables have $2^9$ = 512 rows and there are $2^{512}$ possible rules. Multiplied out, this number is

13,407,807,929,942,597,099,574,024,998,
205,846,127,479,365,820,592,393,377,723,
561,443,721,764,030,073,546,976,801,874,
298,166,903,427,690,031,858,186,486,050,
853,753,882,811,946,569,946,433,649,006,
084,096

Don't even try to think about this.

The rule for a table can be encoded as an integer in the following way. First write out the values in the C' column in reverse order (to correspond to a base-2 integer, in which the most significant digit is at the left.) Then convert this base-2 number to base 10.

For the parity rule, the base-2 integer is

10000110011010010110100110010110

which in base 10 is 1771476585.

Integer encodings of state-transition rules are unintuitive, but they are useful for some cataloging and programming purposes.

### Totalistic Rules

Many interesting state-transition rules depend only on the sum (total) of the values of cells in the neighborhood. Such rules are termed *totalistic*.

Two kinds of totalistic rules have been studied extensively: purely totalistic and *outer-totalistic*, also called *semi totalistic*. In a purely totalistic rule, the next state of the core cell depends only on the sum of the states of all cells in the neighborhood, including the core cell. In outer-totalistic rules, the state of the core cell depends on the sum of all the cells in the neighborhood except the core cell, and the state of the core cell figures in separately. These rules are illustrated in Figures 15-18, where cells with the same color are taken together for the purpose of forming sums.

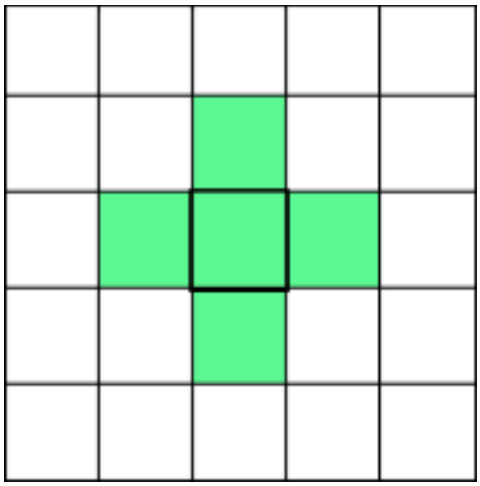

**Figure 15. Von Neumann Totalistic Rules**

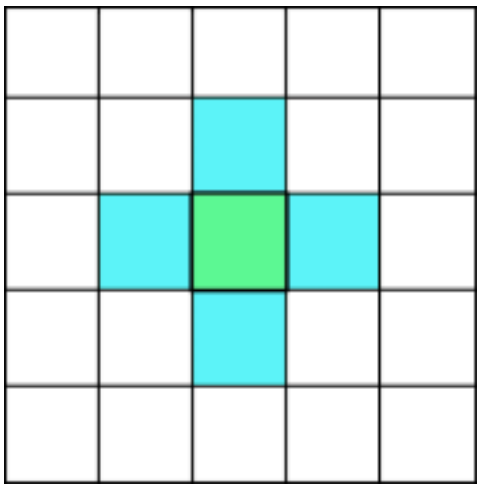

**Figure 16. Von Neumann Outer-Totalistic Rules**

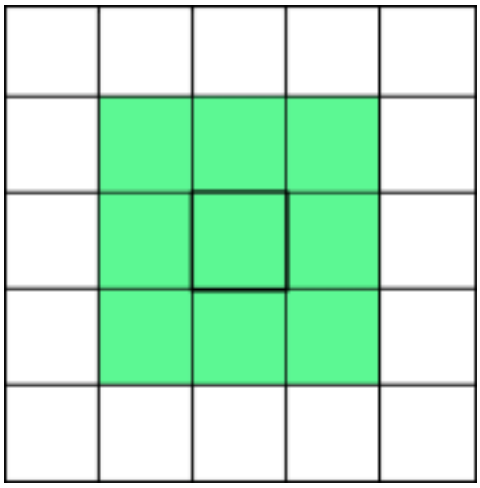

**Figure 17. Moore Totalistic Rules**

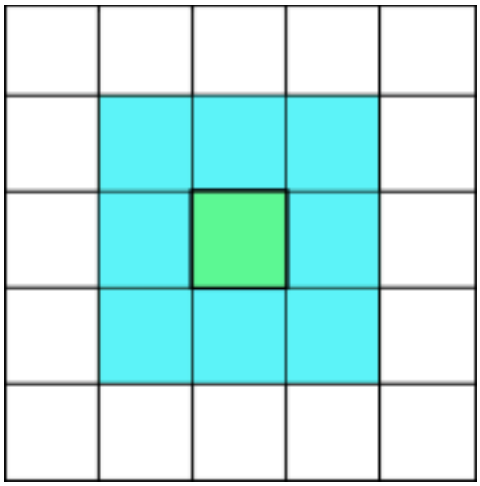

**Figure 18. Moore Outer-Totalistic Rules**

For purely totalistic rules, the number of possible sums is one more than the number of cells in the neighborhood: 0 through *n* for an *n*-cell neighborhood. For example, rule tables for 5-cell neighborhoods have only 6 rows.

The parity rule, which is totalistic, has the following table for the von Neumann neighborhood, where Σ indicates the sum:

| Σ | C' |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 0 |
| 3 | 1 |
| 4 | 0 |
| 5 | 1 |

The totalistic integer code for this rule is 42.

There are $2^6 = 64$ purely totalistic rules for 5-cell neighborhoods and $2^{10} = 1{,}024$ purely totalistic rules for 9-cell neighborhoods.

For outer-totalistic rules, the number of possible sums of the cell states equals the number of cells: 0 through *n*–1. For each, there are two core cell states to consider, so rule tables for 5-cell neighborhoods have 10 rows and rule tables for 9-cell neighborhoods have 18 rows.

The Game of Life [1,2] uses an outer-totalistic rule for the Moore neighborhood. In the terminology of the Game of Life, a live cell (C = 1) remains alive (C' = 1) only when

surrounded by 2 or 3 live neighbors. Otherwise it dies (C' = 0). A dead cell (C = 0) comes to life (C' = 1) only when it is surrounded by exactly 3 live neighbors. The corresponding table is:

| ΣC | C' | ΣC | C' |
|---|---|---|---|
| 00 | 0 | 41 | 0 |
| 01 | 0 | 50 | 0 |
| 10 | 0 | 51 | 0 |
| 11 | 0 | 60 | 0 |
| 20 | 0 | 60 | 0 |
| 21 | 1 | 70 | 0 |
| 30 | 1 | 71 | 0 |
| 31 | 1 | 80 | 0 |
| 40 | 0 | 81 | 0 |

The base-2 code for this rule is

0000000011100000

which is 224 in base 10.

There are $2^{10}$ = 1,024 outer-totalistic rules for 5-cell neighborhoods and $2^{18}$ = 262,144 outer-totalistic rules for 9-cell neighborhoods.

### Life-Type Rules

There are many variations on the Game of Life. They follow the pattern of the Game of Life rules, but the numbers of surrounding live cells necessary to keep a cell alive or restore a dead cell to life vary.

These rules can be represented compactly by listing the numbers of live cells for the two cases. For the game of life itself, these numbers are 2 and 3 to maintain life and 3 to restore life.

One encoding method lists the two sets of numbers with a separating slash. Thus, the Game of Life rule is 23/3.

### Other Ways of Expressing Rules

As if the situation was not already sufficiently confusing, some cellular automata applications have their own ways of specifying state-transition rules.

Many rules that are cumbersome when cast as tables or incomprehensible when given as integer codess are nonetheless fundamentally simple. For example, the parity rule is easy to express and understand in plain English: It's just a matter of whether the sum of the states of all neighboring cells is odd or even.

For precision and as a basis for implementation in programs, pseudo-code will do. The parity rule for the von Neumann neighborhood can be written as

(N + E + S + W + C) mod 2

and the 1-of-8 rule can be written as

if (C = 1) and
(NW + N + NE + E + SE + S + SW + W)
= 1 then 1 else 0

A few simple conventions can simplify code like this. For example, if T(N) is the sum of all states in neighborhood N and O(N) is the sum of all outer states in N, then the parity rule becomes

T(N) mod 2

and the 1-of-8 rule becomes

if (C = 1) and (O(N) = 1) then 1 else 0

## References

1. *Drawdown Automata, Part 1: Basic Concepts*, Ralph E. Griswold, 2004:
http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_dda1.pdf

2. *The Recursive Universe; Cosmic Complexity and the Limits of Scientific Knowledge*, William Poundstone, Contemporary Books, 1985.

Ralph E. Griswold
Department of Computer Science
The University of Arizona
Tucson, Arizona