

Rulespace Combinatorics: Calculating the Amount of Cellular Automata that Golly v3.0b1 Can Simulate

Introduction

In the context of Golly, a cellular automaton (CA) is a system where an unbounded 2D square grid of integers ('cells') ranging from 0 to n ($n < 256$) inclusive changes over the course of discrete time steps ('generations') according to cells' own states and the states of their surroundings (their 'neighbourhoods'). Each CA has its own value for n .

In this document, two CA are considered separate if one or both of these is true:

1. Given at least one possible initial condition, the next generation of the two CA from the same initial condition has at least one cell which is different. If one CA is simply a mirrored or rotated version of another, they are still considered separate.
2. The two CA have a different number of possible states (n).

In practice, the CA Golly can run have 'neighbourhoods', defined as a pattern of cells around which a certain cell is influenced by. With the criteria provided, two CA with different 'neighbourhoods' are not necessarily unequal.

Golly contains various algorithms to simulate CA, but most of these algorithms can be ignored because every CA with the range 1 Moore neighbourhood (that is, a 3x3 square around the center cell) or some subset of it is covered by the RuleLoader algorithm. The rest fall under the Larger than Life algorithm.

RuleLoader

RuleLoader can simulate any CA with a range 1 Moore neighbourhood and a number of states from 2 to 256. Where n is the number of states, there are n^9 possible initial conditions for a cell and its neighbourhood. Each possible initial condition will have an outcome; thus we have n^{n^9} n -state CA. We use summation to account for the varying number of states, which gives us this equation, where R_L is the amount of CA RuleLoader can run:

$$R_L = \sum_{n=2}^{256} n^{n^9}$$

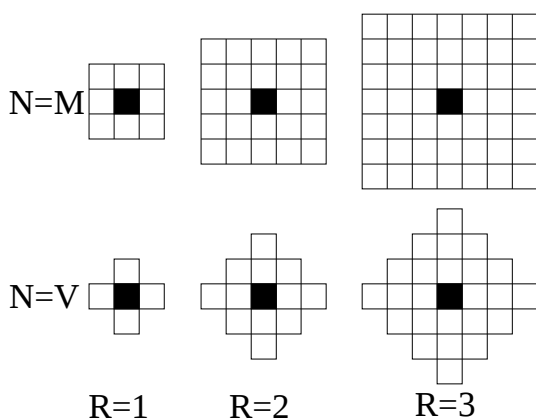
Larger than Life

What follows is a description of Larger than Life (LtL) CA as implemented in Golly version 3.0b1. Since RuleLoader is incapable of simulating CA where neighbours outside the Moore neighbourhood can effect the state of a cell in the next generation, and LtL CA can, the set of all LtL rules is not a subset of the set of all RuleLoader rules.

Definition

The LtL rulespace is a group of CA which operate under a consistent set of rules, where each LtL CA is defined by a set of values for the following parameters:

- **R** – $50 \geq R \geq 1$ – The Range; radius of the neighbourhood.
- **C** – $255 \geq C \geq 2$ – The amount of states.
- **M** – 0 or 1 – Whether or not the center cell is part of the neighbourhood. 1 if so, else 0.
- **N** – V (Von Neumann) or M (Moore) – The shape of the neighbourhood. (Normally, the letter N is used instead of V but I found that confusing)
- **S_{min} , S_{max} , B_{min} , and B_{max}** – Minimum and maximum survival and birth cases. $S_{min} \leq S_{max}$ and $B_{min} \leq B_{max}$.



The neighbourhood is defined by R, N, and M. If M=1, it will include the center cell, otherwise it won't. If N=V, the neighbourhood is defined as the set of cells with a taxicab distance (sum of the absolute differences between the two pairs of coordinates) from the center cell $\leq R$.

If N=M, the neighbourhood is the set of cells where the highest of the two absolute differences in coordinates from the center cell $\leq R$. The result is that M creates a square, while V creates a diamond.

Illustration 1: Neighbourhood shapes and sizes

The state a cell will take in the next generation then depends on the survival and birth conditions. If the cell is live (1), then it will die if, where L is the amount of live cells in the neighbourhood, $L < S_{min}$ or $L > S_{max}$. The same goes for births; if the center cell is dead (0), it will become alive if $L \geq B_{min}$ and $L \leq B_{max}$. Any other cells less than 2 will stay the same.

If $C=2$, live cells become dead cells when they die. Otherwise, they become state 2. A cell of state $a \geq 2$ will become $a+1$ in the next generation unless $a+1 \geq C$, in which case it becomes 0.

It is possible to have a rule where all live cells die regardless of their neighbourhood, every dead cell stays dead, or both. Mathematically, this can be expressed by S_{max} and/or B_{max} being less than 0, or S_{min} and/or B_{min} being greater than the neighbourhood size.

Reduction

We will now attempt to simplify the problem. M is redundant, because it's simply equivalent to adding one to the survival conditions, and as such all LtL CA have $M=1$ and $M=0$ equivalents. C can be accounted for simply by multiplying our result by 254, for all the possible values it can have. The range of values of the survival and birth conditions is dependant on the area (amount of cells) of the neighbourhood, so we must compute that first.

Neighbourhoods

I will use m and v as function names for convenience here. For the moore neighbourhood ($N=M$), the area is $m(R) = (2R+1)^2$, as demonstrated by this visual proof:

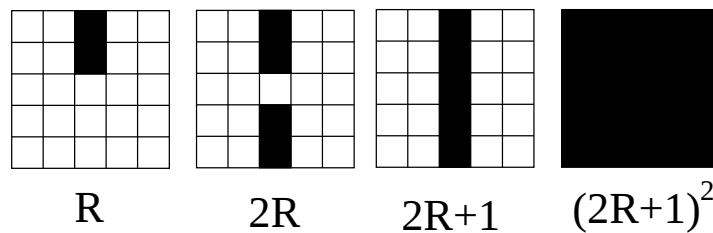


Illustration 2: Visual proof for $m(R)$ with example of $R=2$

For a Von Neumann neighbourhood ($N=V$), it is equal to $v(R) = R^2 + (R+1)^2$.

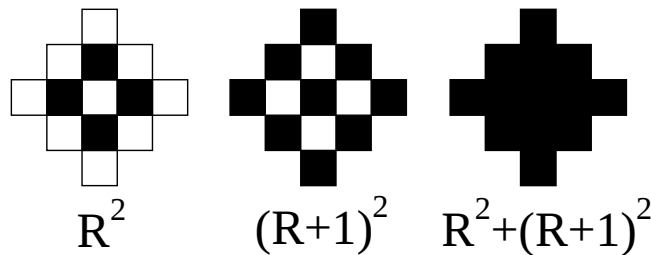


Illustration 3: Visual proof for $v(R)$ with example of $R=2$

The area of the diamond formed by the Von Neumann neighbourhood can be shown to be equal to that of two opposing checkerboard patterns, one of which contains the rim of the diamond and the other containing the insides.

The birth and survival conditions already know the state of the center cell, and as such we can subtract 1. However, there are $x+1$ possibilities for the amount of live cells in a set of x cells, from 0 to x , so we add 1. This means that, for the combinatorics, we can use $m(R)$ and $v(R)$ as the amount of possible states the neighbour count can take.

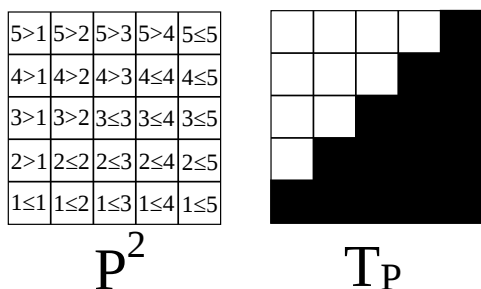


Illustration 4: Visual aid for the correlation to triangular numbers using T_5

Survival and Birth

The amount of *min/max* combinations, where P is $m(R)$ or $v(R)$, is the amount of unique pairs of numbers from 1 to P inclusive, which is a triangular number, T_P (the P th

triangular number). Since it is possible to have no survival/birth conditions, we use T_{P+1} for a *min/max* combination.

There are two *min/max* combinations, with S and B, thus we use $(T_{P+1})^2$ to represent the amount of rules with neighbourhood size P.

LtL Conclusion

Combining all of this, we get the amount of unique LtL CA that exist, which we will call L_T , but first we will recall m and v :

$$v(R) = R^2 + (R+1)^2$$

$$m(R) = (2R+1)^2$$

$$L_T = 254 \left(\sum_{R=1}^{50} \left(T_{m(R)+1}^2 + T_{v(R)+1}^2 \right) \right)$$

Q.E.D.

We can use the fact that $T_n = \frac{n(n+1)}{2}$ to expand the equation out.

$$L_T = 254 \left(\sum_{R=1}^{50} \left(\left(\frac{m(R)(m(R)+1)}{2} + 1 \right)^2 + \left(\frac{v(R)(v(R)+1)}{2} + 1 \right)^2 \right) \right)$$

$$L_T = 254 \left(\sum_{R=1}^{50} \left(\left(\frac{(2R+1)^2((2R+1)^2+1)}{2} + 1 \right)^2 + \left(\frac{(R^2+(R+1)^2)(R^2+(R+1)^2+1)}{2} + 1 \right)^2 \right) \right)$$

Total Conclusion

To add up all these CA, we will define the amount of LtL CA which are not covered by RuleLoader, which we'll call L_t (lowercase t); it would just exclude the case of $R=1$, so the summation will start at $R=2$. As such, the total number of CA that Golly v3.0b1 can simulate equals $R_L + L_t$.