Conway's Game of Life on Finite Boards without Boundaries

A. V. Lelechenko

Well-known Conway's Game of Life is usually played on the infinite plane board. We introduce it on finite surfaces without boundaries: the torus, the Klein bottle and the projective plane. An effective algorithm for the exhaustive search of stable and repeating patterns is suggested. We present patterns which have much longer periods of oscillating than patterns of the comparable size in the classic Game of Life.

Introduction

Conway's Game of Life is a cellular automaton on the two-dimensional orthogonal grid of square cells, each of which is *alive* or *dead.* Its evolution is fully determined by the initial state. Each cell has eight neighbours in horizontally, vertically, or diagonally adjacent cells. At each step any cell with ≤ 2 or ≥ 3 neighbours become dead and any cell with exactly 3 neighbours become alive.

In spite of such simplicity of rules many patterns have long and executable evolution. Generally one can classify all non-vanishing patterns as evolving to *still lives* which are stable, *oscillators* which are periodically oscillating, and aperiodic patterns (including so-called *gliders, guns, puffers, rakes* and others).

The infinite size of the board causes that most of the combinatorial-optimization problems are very hard. The aim of this paper is to study patterns on finite boards without boundaries. The case of a surface with boundaries (e. g., square board) is just cropped Game of Life with border effects, so it seems to be not very interesting.

Namely we start with square $N \times N$ board and glue opposite sides. If we glue them without twist we obtain board-torus S^2_N; if we twist before gluing one pair of sides we obtain board-Klein bottle K_N; and in the case of the both pairs of sides twisted we get board-projective plane RP^2_N. We shall refer to these types as to board's topology.

Some very tentative results of our study were published in [1].

Taking into account a finite number of states no aperiodic patterns exists on the finite boards. So we are interested in stable (having period $T = 1$) and periodical ($T \geq 2$) patterns. Our goal is to list by the exhaustive search all such patterns on S^2_N, K_N and RP^2_N for small values of N.

Algorithm

For fixed board $N \times N$ let $B = \{0, 2^{N^2} - 1\}$ be the set of all possible patterns and let $E\colon B \to B$ be the *evolution operator,* which maps pattern to its state on the next turn. So search for oscillators is a search for patterns p such that $p = E^T p$ for some $T > 0$.

We call operator $M\colon B \to B$ as a *move operator* if it commutes with E. In other words $MEp = EMp$ for all $p \in B$. We have proved the following statement.

Proposition. Pattern $p \in B$ is periodic if and only if (not necessary distinct) move operators $M_1, \ldots, M_k$ exist such that

$$M_1 E \cdots M_k E p = p.$$

In this case k is called a *quasi period* of p.

It is hard to determine analytically the complete set of move operators, but for the need of our algorithm it is enough to use even incomplete one. (Though in general the more the better.)

In the case of S_N^2 shifts by both dimensions, rotations at 90°, reflections and their combinations are move operators. On K_N move operators include shifts by one dimension, rotation at 180° and reflections. And the only known move operators for RP_N^2 are rotations at 90° and reflections.

For the given set of move operators $\{M_k\}_{k=1}^r$ we define *norm operator* $||\cdot||$ such that

$$||p|| = \min\{M_k p\}_{k=1}^r.$$

Now we are ready to write down our algorithm in pseudocode:

```
stack ← ()
periods ← {}
for p ← 1..2^(N^2)
    p' ← ||p||
    if p' < p ∨ p' ∈ periods then
        continue;
    while p' > p ∧ p' ∉ periods ∧ p' ∉ stack
        push(stack, p');
        p' ← ||Ep'||
    if p' ∉ stack then
        stack ← ()
    else
        while stack ≠ ()
            k ← pop(stack)
            periods ← periods ∪ {k}
            if k = j then
                stack = ()
```

Implementation

The suggested algorithm was implemented in the following way.

First, a program written in PHP determines an evolution operator E for a given board topology and size. Then it generates all known move operators, combines them and removes duplicates. For the obtained set the norm operator is built. Finally all this data is translated into optimal C-code and saved to file. Language PHP was chosen for this task because of the easy string operating and the abilities of the functional programming.

Secondly, a C-program is compiled and run. It is the heart of the computational process. It performs an exhaustive search of all positions due to the algorithm above and can be parallelized easily. It uses self-written implementation of single-linked lists and Glib for b-tree manipulating. This program have to work for a long time, so it dumps its state to disk periodically and these dumps can be loaded in. Finally it returns a list of periodic patterns.

Thirdly, a utility written in C postprocess the list of patterns to remove duplicates and determine periods. A statistical report is generated by a simple AWK-script.

Results

An exhaustive search for $2 \le N \le 6$ was completed. Calculations for $N = 7$ are still evaluating, for now only tentative results are available. Results are presented in table 1.

Case of board's size $N = 8$ seems to be far beyond our computation abilities.

One can compare table 1 with the smallest known oscillators on the infinite board [2] and see that in the case of finite boards without boundaries there are much more compact and small patterns almost for all values of T. For example, there is no known oscillator with $T = 38$ on the infinite board — and we have one on RP_7^2.

Table 1. Periodical patterns and their periods

N	S^2_N	K_N	RP^2_N
2	$T = 1$	$T = 1$	$T = 1, 2$
3	$T = 1$	$T = 1$	$T = 1$
4	$T = 1, 2, 4, 8$	$T = 1, 2, 4, 8$	$T = 1, 2, 4, 18$
5	$T = 1, 2, 3, 4, 5, 10,$ 20	$T = 1, 2, 3, 4, 5, 10,$ 20, 40	$T = 1, 2, 3, 4, 14$
6	$T = 1, 2, 3, 4, 6, 8,$ 12, 24	$T = 1, 2, 4, 6, 8, 10,$ 12, 15, 48, 60	$T = 1, 2, 3, 4, 5, 6,$ 8, 28
7	$T = 1, 2, 3, 4, 6, 7,$ 8, 12, 14, 28	$T = 1, 2, 3, 4, 5, 6,$ 7, 8, 9, 14, 28, 56	$T = 1, 2, 3, 4, 5, 6,$ 7, 8, 14, 38, 56

Fork me on Github

Source code and lists of found periods will be available at

`https://github.com/Bodigrim/finite-life`

References

[1] Lelechenko A. V. On finite Conway games // International summer mathematical school in memory of V. A. Plotnikov: Book of abstracts. — Odessa, Astroprint, 2010. — p. 72 (in Russian).

[2] Oscillator // LifeWiki — URL: http://www.conwaylife.com/wiki/Oscillator

Authors

Andrew Vladimirovich Lelechenko — the 2nd year postgraduate student, Department of computer algebra and discrete mathematics, Faculty of mathematics, Institute of mathematics, economics and mechanics, I. I. Mechnikov Odessa National University; E-mail: *1@dxdy.ru*