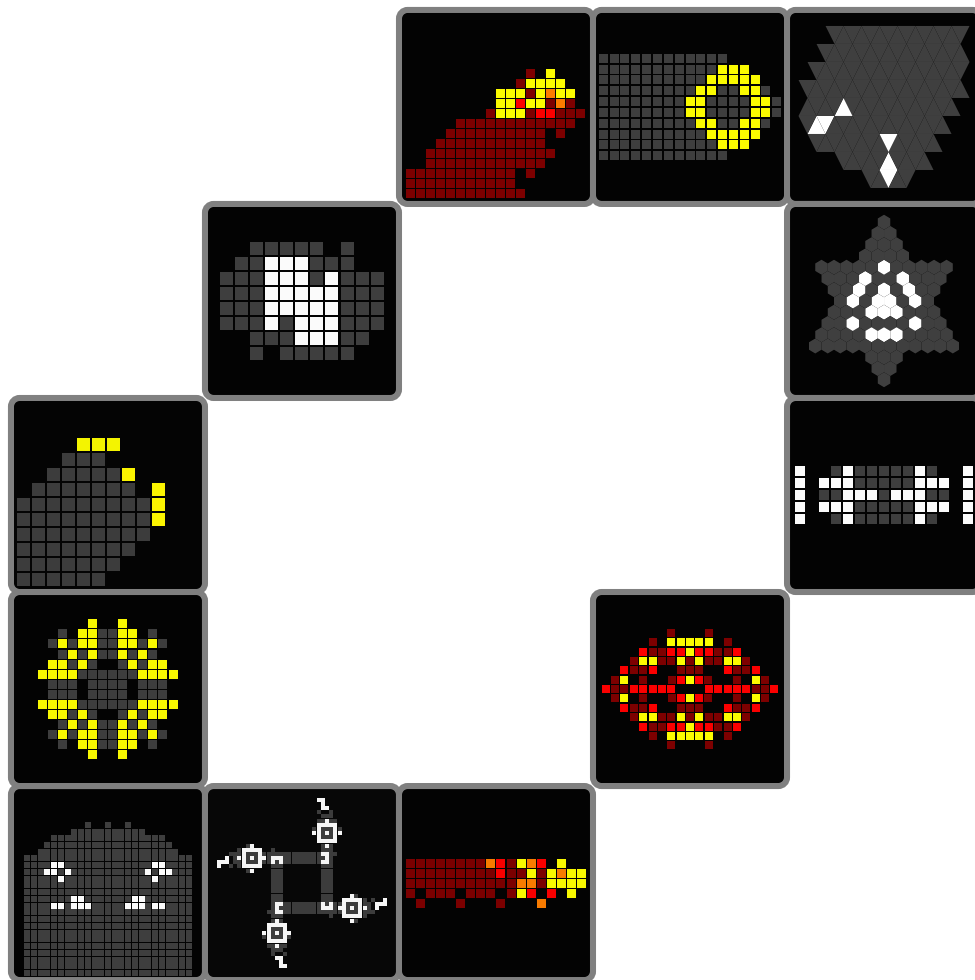


A World of Cellular Automata

Beneath the Surface

iniquity816



Copyright © 2026 iniquity816

PUBLISHED BY MYSELF :3

[HTTPS://GENEVEI.GITHUB.IO/OCABOOK/](https://genevei.github.io/ocabook/)

The Destruction of The World November 20th 2026, 233 Days After The Rapture

Contents

Preface	i
Intentions	i
Prerequisites	i
Usage	i
Exercises	ii
Acknowledgements	ii

I

Square tiling

1	One-dimensional rules	3
1.1	Elementary cellular automata	3
2	The von Neumann neighbourhood	5
2.1	Two-state von Neumann rules	5
2.2	Three-state von Neumann rules	5
2.3	Circuitry rules and higher-state rules	6
	The Moore neighbourhood	7
2.4	Outer-totalistic rules	7
2.4.1	Life variants	7
2.4.2	Other interesting rules	7
2.5	Isotropic non-totalistic rules	7

II

Hexagonal tiling

III

Triangular tiling

2.6	Emulation of rules on the hexagonal tiling	15
------------	---	-----------

IV	Other tilings	
	Generalisations	21
2.7	Extensions	21
	BSFKL	21
2.8	Alternating rules	21
2.9	Generations	21
2.10	Larger than Life	21
2.10.1	Higher-range outer-totalistic	21
2.11	Cyclic cellular automata	21
V	Meta-analysis	
2.12	History	25
2.12.1	Genealogy	25
2.13	Cosmology	25
2.14	Software and computation	25
2.14.1	Implementation of rulespaces	25
2.14.2	Pattern identification	25
2.14.3	Pattern searching	25
2.14.4	Rulegolfing	25
	Bibliography	27
	Articles	27
	Books	27
	Index	29
	Appendix	31
	Pattern identification	33
2.15	Object formats	33
	Appendix Chapter Title	35
2.16	Appendix Section Title	35

Preface

Intentions

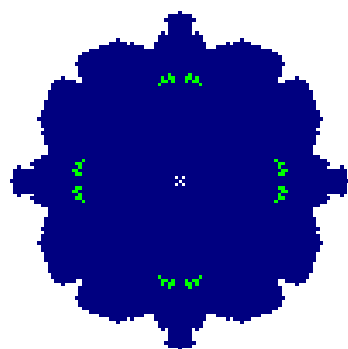
This book is intended to be an unofficial counterpart to *Conway's Game of Life: Mathematics and Construction* by Nathaniel Johnston and Dave Greene (hereinafter "the Life book"), focusing on the large variety of cellular automata that allow for elementary objects and mechanisms that Life simply does not have.¹ Rules with a square tiling, being the most ubiquitous manifestation of cellular automata rules, are well-explored and are thus the main object of observation of this book, but rules on other, more novel tilings (such as the hexagonal and triangular tilings) are also included to present the world of cellular automata more objectively, and to perhaps foster exploration and a better understanding of these rules. Rules with larger ranges, such as those explored by Kellie Evans, are also present, but note that these rules become less easier to simulate as the range of the neighborhood, or the area of cells that can affect one cell, increases. The maximum range that can be simulated in most CA simulation software is 500.

Prerequisites

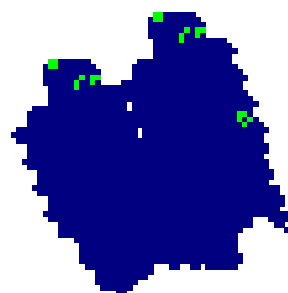
It is assumed that you have the program Golly installed, available [here](#), which natively facilitates the simulation of almost all rulespaces discussed in this book and enables the creation of scripts to achieve certain goals. Additionally, it is assumed that you are familiar with the Life book, which discusses ideas that mostly overlap with this book. Nothing else is required to use each program comfortably, but web resources can be used to broaden your knowledge if necessary.

Usage

Much like the Life book, this book is reliant on figures to illustrate patterns in a non-changing manner, which makes the visualisation of spaceships and other moving objects more difficult. Hence, cells that were alive in previous generations but are now dead (the "envelope" of the pattern) are highlighted blue to indicate change. This ensures the appearance of motion in moving objects and volatility in oscillators. Still lives, showing no movement or change from one generation to the next, are presented "as-is", without highlighting of any kind.



(a) Period-661 oscillator in B356/S23.



(b) (5,2)c/190 spaceship in B38/S23, run for 760 generations.

Examples of envelopes for an oscillator and a spaceship (each in their minimal phase, in terms of population), coloured using LifeHistory state colours.

Cells may also be given different colours as part of an exercise, which we will discuss below.

¹Any object in any rule with a range-1 Moore neighbourhood, like Conway's Game of Life, can be emulated in Life using OEOP metacells.

Exercises

Exercises are also used to test one's knowledge and are placed at the end of each chapter. They are assigned difficulty ratings, ranging from Difficulty-1 (straightforward and often completed without much effort) to Difficulty-5 (projects that may take a day or two to complete). It is hoped that these exercises improve the breadth of one's knowledge and experience.

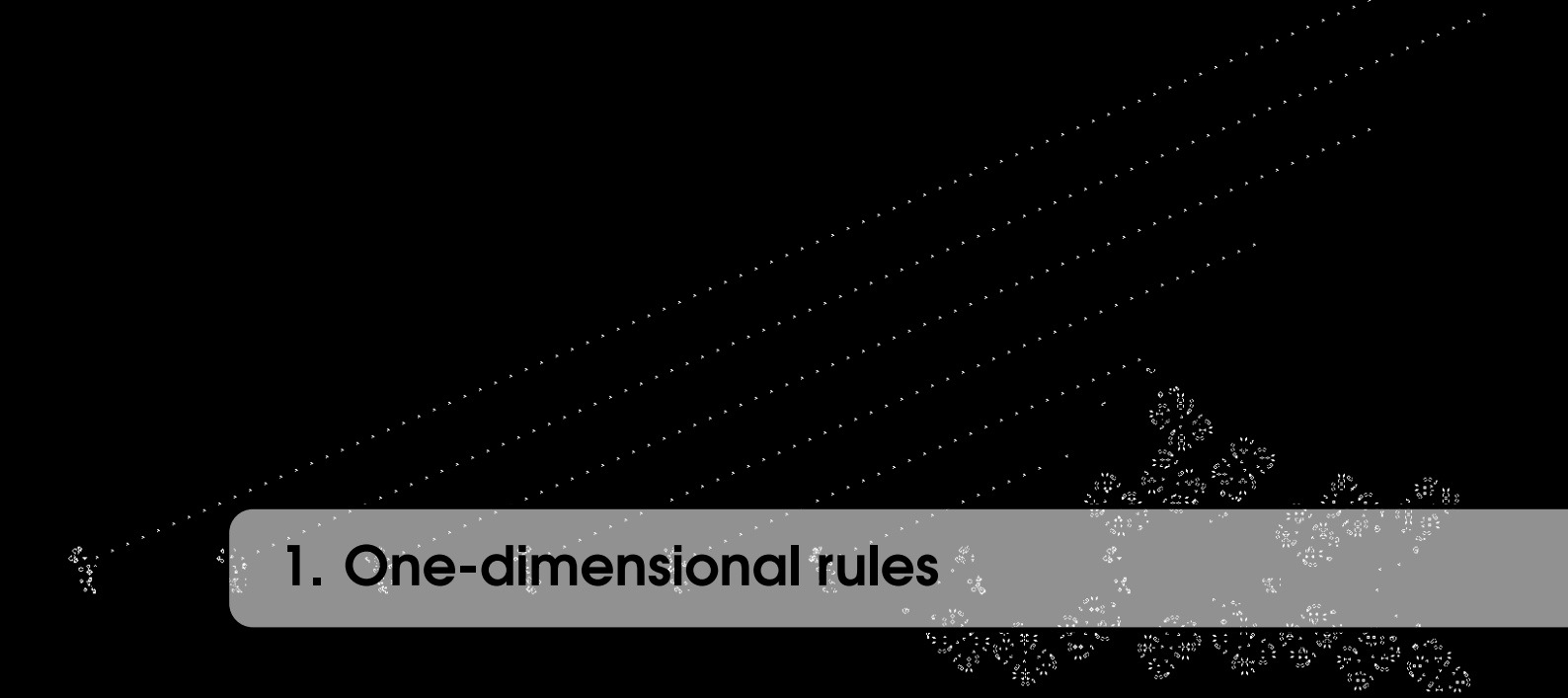
Acknowledgements

First and foremost, we would like to thank Nathaniel Johnston and Dave Greene, who directly inspired this book and indirectly motivated me to write it. We extend thanks to David S. Miller, the forum user "FWKnightship", and the creators of Golly (Andrew Trevorrow and Tomas Rokicki). Finally, we would like to thank Mathias Legrand and Velimir Gayevskiy, creators of the template *Legrand Orange Book*, available at LaTeXTemplates.com.



Square tiling

1	One-dimensional rules	3
1.1	Elementary cellular automata	3
2	The von Neumann neighbourhood	5
2.1	Two-state von Neumann rules	5
2.2	Three-state von Neumann rules	5
2.3	Circuitry rules and higher-state rules	6
	The Moore neighbourhood	7
2.4	Outer-totalistic rules	7
2.5	Isotropic non-totalistic rules	7



1. One-dimensional rules

1.1 Elementary cellular automata

As the name *one-dimensional cellular automata* suggests, the mechanisms that allow for cells to interact with each other are limited to the cell itself and other cells on the same row. Elementary cellular automata limit the neighbours of a cell further, to just the cells horizontally adjacent to it and itself. They were first considered in 1983 by **Stephen Wolfram**. 256 ($= 2^8$) elementary cellular automata exist; an elementary cellular automaton's **Wolfram code** is a decimal representation of the automaton's transition rules written in binary.

2. The von Neumann neighbourhood

The von Neumann neighbourhood, named after none other than John von Neumann (1903–1957), limits the number of neighbours a single cell can have to just the cells orthogonally adjacent to it. A 29-state rule with the von Neumann neighbourhood, devised by von Neumann in the 1940s with suggestions from Stanisław Ulam (1909–1984) to realise a **universal constructor** in the form of a **self-replicating machine**, was the first example of a cellular automaton using this neighbourhood (and the first cellular automaton in general). The first implementation of his machine, using his original 29-state specification, would only appear in 2007.¹

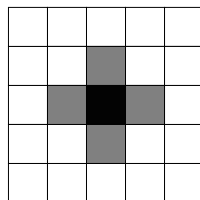


Figure 2.1: The von Neumann neighbourhood (in grey) of the central cell (in **black**).

2.1 Two-state von Neumann rules

Two-state von Neumann rules are extremely limited; they require birth on 1 neighbour to expand outside their initial bounding box (but this, in turn, makes rules hopelessly explosive). A simple proof of this is provided below. [add proof] Such rules are customarily suffixed with the "V" character to denote their neighbourhood. This practice can and has been extended to other types of von Neumann rules.

2.2 Three-state von Neumann rules

Attention to such rules has been paid since at least 2007, when **Artem Dergachev** stumbled upon the rule he termed B1x2/S23V. More recently, in October 2025, a notation emerged for writing rules of this particular type. 3⁴⁵ (2954312706550833698643) rules belong to this family.

¹An earlier implementation by Renato Nobili and Umberto Pesavento in 1994 used a version of von Neumann's rule with 32 states. [10.1162/artl.1995.2.4.337]

2.3 Circuitry rules and higher-state rules

All four Banks rules, Codd, Devore, Flow6, and von Neumann's constructor rule all use the von Neumann neighbourhood. [add pictures of Banks-I to Banks-IV, Codd, Devore, Flow6, and the Sphinx in action]



The Moore neighbourhood

2.4 Outer-totalistic rules

In outer-totalistic rules, only the number of alive cells a cell has matters (cells are born without consideration of the positions of the neighbours of the cell, relative to each other). The Game of Life belongs to this family; 2^{18} (= 262144) rules belong to this family. Out of these 262144 rules:

- 2^{16} (65536) of them have the lowest birth condition as one neighbour, meaning no stable objects except replicators exist.
- 2^{15} (32768) of them have the lowest birth condition as two neighbours, meaning lightspeed frontends made of dominos can propagate through the universe and subsequently cause **explosions**.
- 2^{14} (16384) of them have the lowest birth condition as three neighbours, meaning growth is often non-explosive, or at the very least predictably so. The aforementioned rules listed above are in this particular family.
- 2^{14} of them have the lowest birth condition as four or more neighbours, meaning growth is impossible and patterns localize in boxes that are unable to grow or expand by themselves.
- 2^{17} (131072) of them have the lowest birth condition as zero neighbours. If the highest survival condition in a rule in this family is eight neighbours (the maximum), then no finite patterns can exist, as the background becomes filled with infinitely many alive cells. If the highest survival condition is not eight neighbours, as is the case in 2^{16} of them, then the background flickers between dead and alive each generation; this is often circumvented by CA simulation software in order to avoid simulating infinite cells, by simulating the rule internally as an **alternating rule**.

Starting in 1998, **David Eppstein** began maintaining a database of spaceships, or gliders as all spaceships were termed, in outer-totalistic rules.

2.4.1 Life variants

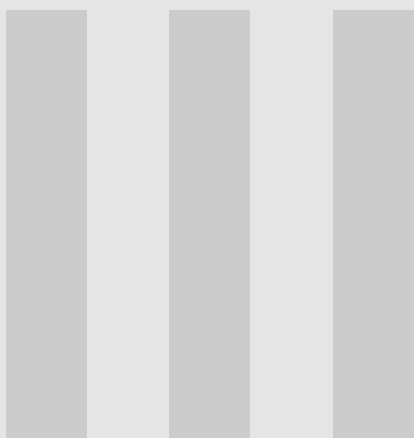
HighLife (B36/S23)

2.4.2 Other interesting rules

2.5 Isotropic non-totalistic rules

Isotropic non-totalistic rules 2^{102} (5070602400912917605986812821504) rules belong to this family.

The



Triangular tiling

2.6 Emulation of rules on the hexagonal tiling

IV

Other tilings

	Generalisations	21
2.7	Extensions	21
2.8	Alternating rules	21
2.9	Generations	21
2.10	Larger than Life	21
2.11	Cyclic cellular automata	21

The



Generalisations

2.7 Extensions

The

BSFKL

The

2.8 Alternating rules

The

2.9 Generations

The

2.10 Larger than Life

2.10.1 Higher-range outer-totalistic

2.11 Cyclic cellular automata



Meta-analysis

2.12	History	25
2.13	Cosmology	25
2.14	Software and computation	25

The

2.12 History

2.12.1 Genealogy

2.13 Cosmology

Cosmology is the study of the ultimate progression of patterns in cellular automata over time. Depending on how patterns in rules interact and co-evolve with each other, the rule may be classified as one of the following:

- **Stable.** Patterns will typically localize themselves to small, independent portions of their original size that remain stagnant over thousands of generations.
- **Explosive.** Patterns will typically grow beyond their original size, growing unpredictably and often coagulating into a single, volatile clump of cells. Spaceships may escape from such a cluster and can evade the explosion, or in some cases, become assimilated if its speed is too slow to outmaneuver the growth behind it.

This is a popular extension of Wolfram's classification scheme devised in 1983:

- **Class one.** Patterns tend towards the vacuum in a highly predictable manner.
- **Class two.** Patterns tend towards small, non-interfering stationary ash.
- **Class three.** Patterns grow beyond predictability in a chaotic, seemingly unending manner.
- **Class four.** Patterns display both microcosmic order and macrocosmic randomness.

The case of B34/S34 (3-4 Life), first written about in Volume 4 of the Lifeline newsletter, published in December 1971, is a rare example of a rule seemingly considered to be stable before recent advents in computing power challenged these beliefs.

2.14 Software and computation

2.14.1 Implementation of rulespaces

2.14.2 Pattern identification

2.14.3 Pattern searching

2.14.4 Rulegolfing

Bibliography

Articles

Books

Index

A

alternating rule 7, 21

C

classes 25

D

Dergachev

Artem 5

E

Eppstein

David 7

explosive 25

O

one-dimensional

cellular automata 3

R

replicator

self-replicating machine 5

S

stable 25

U

universal construction 5

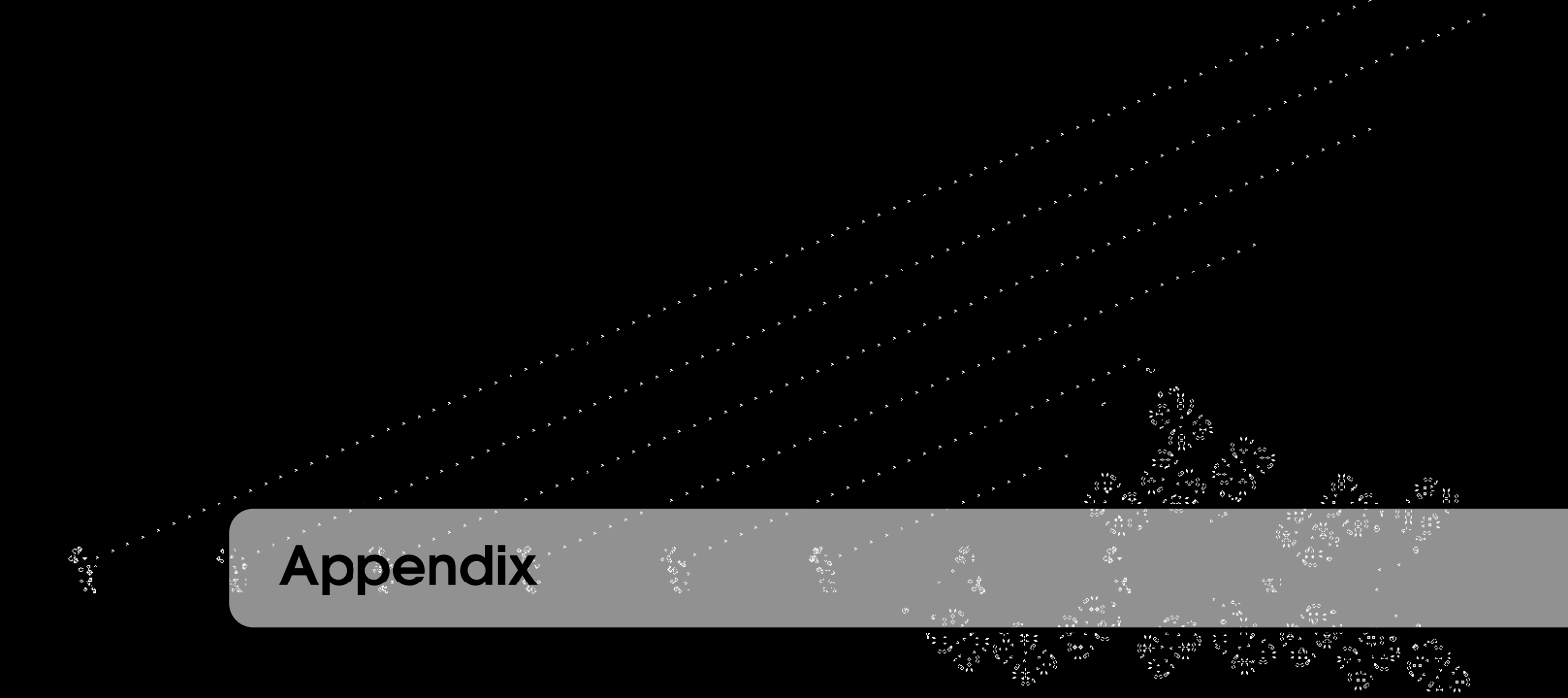
W

Wolfram

classification 25

code 3

Stephen 3



Appendix

Pattern identification

2.15 Object formats

A distinction between pattern and object formats must be made: pattern formats are made to store any configuration of cells, whilst object formats are made to store any identifiable object. "o" and "p" enclosed in parentheses denote the format's intended use.

RLE (p): Short for *Run Length Encoded*. Patterns encoded as such customarily start with a header line, denoting the bounding box of the pattern and the rule it runs in: "x = m, y = n, rule = abc", where m, n, and the placeholder rule are substituted for the total number of columns, rows, and the rule respectively. Runs in the pattern are written as numbers followed by "b" for dead cells and "o" for alive cells (if they exist; the numbers are omitted otherwise.) Newlines are marked with the "\$" character, and a number before said character can be used to encode repeated blank lines.

Comments can be defined with lines beginning with the "#" character and are followed by a letter indicating the purpose of the comment: "C" (or "c") precede a generic string; "N" is the name of the pattern where such exists; "O" lists metadata about the pattern such as its discoverer and the time the pattern was saved; "R" lists the coordinates of the top-left corner of the pattern; and "r" lists the rule of the pattern in "S/B" format without the opening letters.

apgcodes (o): Also known as Extended Wechsler format. Objects are split into strips of five rows that encompass the whole of the pattern. From left-to-right, each column of the strip is read top-to-bottom and the five cells in this order, in binary, are written as the corresponding character in a character set starting with the character "0", proceeding with the numerals before beginning with lowercase letters, ending with the character "w". The characters "x" and "y" compress two and three empty columns ("00" and "000") respectively - "y" followed by any letter in an extended version of the character set ending with the lowercase letter "z" denotes any run of four empty columns or longer, with the letter after it describing its length, and multiple of these can be used to compress empty columns of length longer than 40. "z" characters are used to end the current strip and begin the next one. Once the final strip has been encoded, the pattern is prepended with an "x" character and, depending on the type of pattern, can be followed by one of two strings: "s" and the object's population if it is a still life, and "p" and the object's period if it is an oscillator.

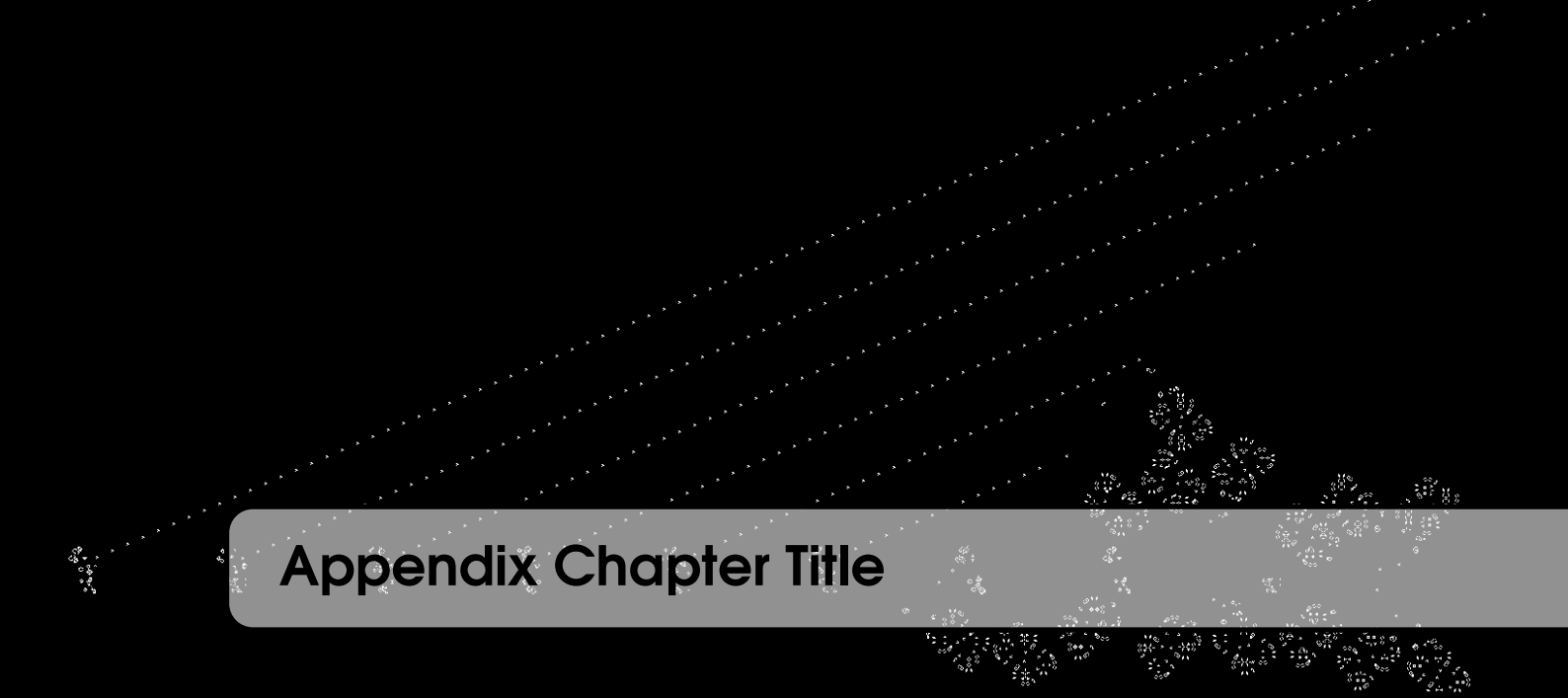
Linear growth patterns are a special case: their apgcodes start with "yl" and are immediately followed by the period at which the growth of the total population repeats, then by the period at which the population of the products repeats, the value of population growth after the number of generations specified in the first value have passed, and the MD5 digest of the total sum of the square of the population in a specific phase and the population n generations after, where n is the

period of the population of the products, and the cube of the aforementioned population, with a "#" character interspersed.

To determine the canonical apgcode for an object, the apgcodes for all orientations and phases of the object are generated, and the lexicographically earliest encoding is considered the object's canonical apgcode. For objects in multi-state rules, the above procedures are repeated for each state the cells have in the object: the pattern is prefixed with identifying information as above, and each state is encoded and separated with underscores. During this encoding, only the relevant cells are encoded - no cells of other states are present.

SOF (o): Also known as Small Object Format. Objects are written as runs of cells alternating between each state for each row. "-" and "." characters separate each encoded row of the pattern and conclude the SOF encoding itself, respectively. If the row does not start with an ON cell, the run of OFF cells is written first; if the row is empty, it is skipped. The characters in the ASCII set between ":" and " " inclusive are used to extend the numeral character set for cell runs greater or equal to 10 cells. If a run is still equal to or longer than 79 cells, the character "]" corresponding to a run of 78 cells is written, followed by a "0" and the number that remains after subtraction of the original run with 78, repeated as necessary. (Repeated empty lines can be truncated to a "+" character and the total number of repetitions, subject to the same rules above, before a non-empty line or the end of the pattern, but this went unimplemented.) Comments can be defined after the "." character; comments can start either with "!" on the very next line representing a generic string, or begin with the name of the pattern enclosed in parentheses that may appear before or after any generic comments.

To compute the canonical SOF encoding of an object, or the SOF of its **Standard Form**, all phases of the object and the eight orientations of these phases are compared with one another. The preference for encoding is any phase with a distinct smallest population; if no such phase exists, the composite bounding boxes of each phase and the orientation being considered (the union of the bounding boxes) are used to equalise each pattern's bounding box. From left-to-right, top-to-bottom, each phase is checked with padding (on the east and south sides) applied; the first instance where an ON cell is present in one pattern where an OFF cell is present in the other is considered the canonical version. All eight orientations (and phases) are sorted by these rules. The Standard Form of an object, and the phase of the SOF encoding, is the first form of this sorting. If both patterns are iterated through and no inequalities are found, the patterns are considered identical for the purposes of sorting. For spaceships, only one orientation is considered: for an orthogonal spaceship, the object is oriented north; for a diagonal spaceship, it is oriented northwest; for an oblique spaceship, it is oriented north-northwest. For linear growth patterns such as guns and puffers, their products are ignored during canonisation. Primarily used by www.pentadecathlon.com from early 2005 to late 2022, upon the website's shutdown.



Appendix Chapter Title

2.16 Appendix Section Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam auctor mi risus, quis tempor libero hendrerit at. Duis hendrerit placerat quam et semper. Nam ultricies metus vehicula arcu viverra, vel ullamcorper justo elementum. Pellentesque vel mi ac lectus cursus posuere et nec ex. Fusce quis mauris egestas lacus commodo venenatis. Ut at arcu lectus. Donec et urna nunc. Morbi eu nisl cursus sapien eleifend tincidunt quis quis est. Donec ut orci ex. Praesent ligula enim, ullamcorper non lorem a, ultrices volutpat dolor. Nullam at imperdiet urna. Pellentesque nec velit eget est euismod pretium.