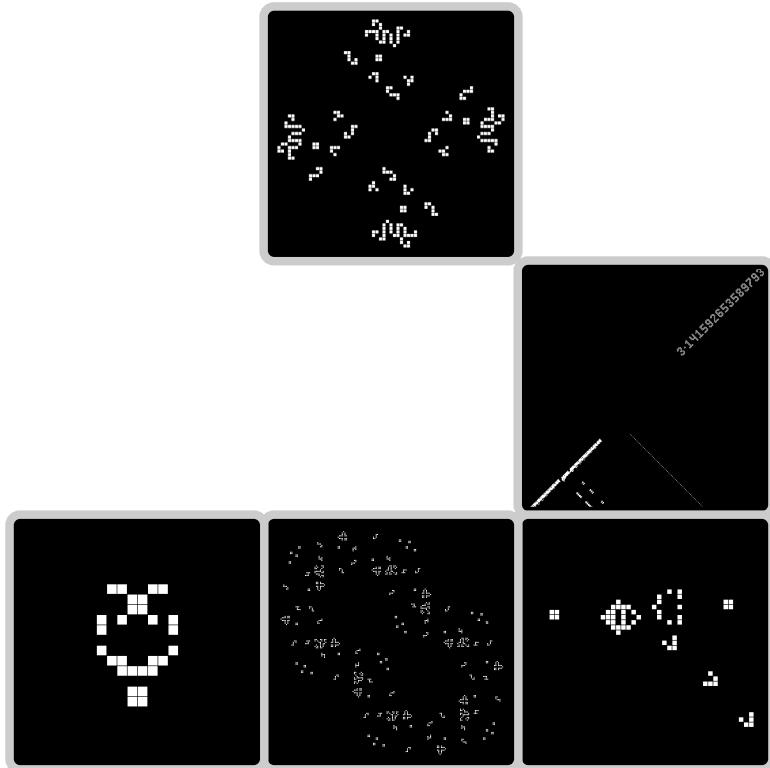


# Conway's Game of Life

## Mathematics and Construction



Nathaniel Johnston and Dave Greene

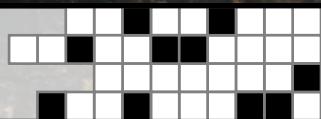


**To John Horton Conway**  
For giving us 50 years of Life.





# Contents



<b>Preface .....</b>	<b>vii</b>
The Goal	vii
Intended Audience	vii
How to Use	viii
Acknowledgments	ix

I

## Classical Topics

II

## Circuitry and Logic

<b>1 Glider Synthesis .....</b>	<b>5</b>
1.1 Two-Glider Syntheses	6
1.2 Syntheses Involving Three or More Gliders	8
1.3 Incremental Syntheses	10
1.4 Synthesis of Moving Objects	13
1.5 Developing New Syntheses	15
1.6 A Gosper Glider Gun Breeder	17
1.7 Slow Salvo Synthesis	19
1.8 Notes and Historical Remarks	27
Exercises	30

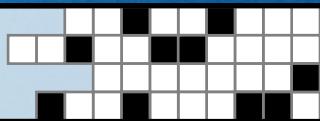
III

## Constructions

## Appendices and Supplements

Bibliography .....	39
Index .....	41

# Preface



## The Goal

In this book, we provide an introduction to Conway’s Game of Life, the mathematics behind it, and the methods used to construct many of its most interesting patterns. We generally try to avoid presenting patterns in isolation or as historical notes, but instead we try to guide the reader through the thought processes that went into creating them in the first place.

While we largely follow the history of the Game of Life as we go through the book, we emphasize that this is *not* its primary purpose. Rather, it is a by-product of the fact that most recently-discovered patterns build upon patterns and techniques that were developed earlier. The goal of this book is to demystify the Game of Life by breaking down the complex patterns that have been developed in it into bite-size chunks that can be understood individually.

## Intended Audience

While this book does not have any formal mathematical or computer science prerequisites, it is written at a level aimed at students of at least a first-year undergraduate university level. Some high-school-level topics like logarithms, the “floor” function  $f(x) = \lfloor x \rfloor$  for rounding numbers down, the binary representation of a positive integer, and summation notation like  $\sum_{k=1}^n k^2$  for adding numbers, are used frequently and without much explanation. A basic understanding of how mathematical proofs and computer programming work is also expected. That is, we expect a certain level of mathematical and computer science maturity from the reader, but no specialized knowledge of either topic.

More specifically, we prove some simple theorems about the Game of Life in Chapters 1 through 5. These proofs do not use any specialized proof techniques or expect the reader to have any specific university-level mathematical knowledge, but rather just expect the reader to be able to follow a logical argument. Similarly, we introduce a programming language for building computer programs out of Life circuits in Chapter ??, so that material will be easier to master if the reader has had prior exposure to computer programming.

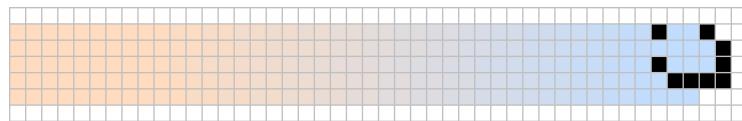
Somewhat more advanced mathematical topics that we make use of are summarized in Appendix ??, though they are typically introduced very gently in the main text as well, and we only require a very surface-level understanding of them. We make use of the greatest common divisor, the least common multiple, and Bézout’s identity when discussing oscillator periods in Chapter ??, so we

introduce these tools in Appendix ???. Infinite series make brief appearances at the end of Chapter ?? and in Section ??, though the reader is not expected to really have any familiarity with them or to understand convergence issues. Finally, big-O notation is used to discuss the growth rate of patterns in Sections ?? and ??, so we introduce this concept in Appendix ??.

## How to Use

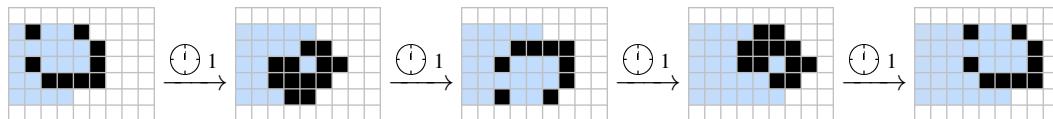
Conway's Game of Life is an extremely visual game, so this book makes very liberal use of figures throughout, particularly when new patterns or techniques for creating patterns are introduced. However, the Game of Life is best observed in motion, which makes static images in a textbook less than ideal as learning tools. In order to help present the motion of patterns a bit better, we do five things:

- Figures are presented with alive cells in black and dead cells in white, but furthermore we use a gradient from blue to orange to denote cells that were alive in past generations of the pattern. Bright blue cells were just alive, whereas cells that are orange were alive in the more distant past (roughly 75 generations or more—see Figure 1).



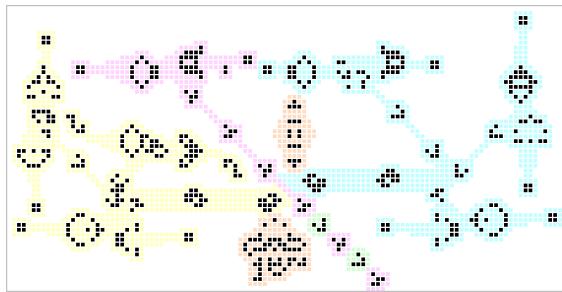
**Figure 1:** An object moving to the right with a gradient behind it indicating how long ago it was in each location. White cells were never alive, black cells are currently alive, blue cells were alive recently, and orange cells were alive long ago (roughly 75 generations or more).

- If we really wish to emphasize what a pattern looks like in different generations, all generations of interest will be displayed, along with arrows that specify how many generations have passed. For example, if we want to clarify exactly what the pattern in Figure 1 does as it moves from left to right, we might display it as in Figure 2.

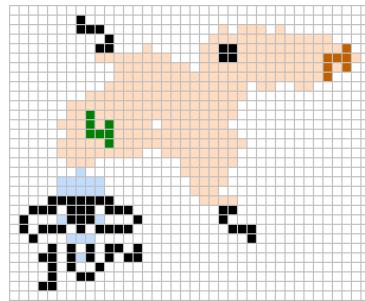


**Figure 2:** The same object as in Figure 1, but displayed in a more explicit fashion. This image shows how the pattern changes every time 1 generation elapses.

- We use colors to highlight various pieces of patterns, and we maintain a consistent coloring scheme throughout the book. Light pastel colors like aqua, magenta, light green, yellow, and light orange are used to highlight *around* objects—cells in these colors are dead, and they indicate a region in the Life plane consisting of cells that all serve some common purpose or logically make up one “object” (see Figure 3). On the other hand, darker colors like dark green, dark orange, and dark red are used to highlight certain live cells. Dark green is typically used to highlight the input to some reaction, dark orange is typically used for the output of the reaction, and dark red is used otherwise (see Figure 4).



**Figure 3:** A glider gun made up of several different component reactions that are highlighted in different colors. In particular, gliders are created by a gun highlighted in magenta, lightweight spaceships are created by guns highlighted in aqua and yellow, and those lightweight spaceships are merged into the original glider stream by oscillators highlighted in light orange.



**Figure 4:** A dark green “Herschel” comes in from the left and creates the dark orange Herschel on the right. Cells highlighted in blue are constantly changing due to being part of an oscillator, whereas cells that are light orange are usually dead, except when the Herschel passes through.

- In the electronic version of this book, almost every figure is actually a clickable link that will open a text file containing RLE or Macrocell code for the displayed pattern (go ahead, click on any of the figures above, or even one of the five images on the cover page). This code can be copy and pasted into Life simulation software like Golly ([golly.sourceforge.net](http://golly.sourceforge.net)) so that it can be explored and manipulated.<sup>1</sup> Note that clicking on figures may not work in certain PDF viewers (such as the viewers built into web browsers), so we recommend using Adobe Acrobat Reader ([get.adobe.com/reader](http://get.adobe.com/reader)) to read this book digitally.
- RLE, LifeHistory, or Macrocell codes for all of the patterns displayed in the book are also available at the book’s website ([conwaylife.com/book](http://conwaylife.com/book)). Furthermore, the patterns can be viewed and manipulated right on that website as well via any modern web browser, without downloading any additional software.

## Exercises

We strongly encourage the reader to work through this book’s many exercises that can be found at the end of each chapter. For this reason, it is extremely important to either download Life software or use the book’s website to view and edit patterns while making your way through the book—Life is meant to be played, not just watched, and many of the exercises simply cannot be solved without the assistance of Life simulation software.

Roughly half of the exercises are marked with an asterisk (\*), which means that they have a solution provided in Appendix ??.

## Acknowledgments

The authors are indebted to dozens of people who opened the world of Conway’s Game of Life to them. Rather than acknowledging the people who discovered the reactions and patterns that we discuss here, they are credited in footnotes and figures throughout the book.

We extend thanks to Adam P. Goucher for numerous contributions to this book, including much of Chapter ???. Thanks to Nicolay Beluchenko, Steven Eker, Tanner Jacobi, Mark Niemiec, and Michael Simkin for various helpful conversations and corrections. Thanks to Andrew Trevorrow and Tomas Rokicki for creating the open-source cross-platform CA editor and simulator Golly ([golly.sourceforge.net](http://golly.sourceforge.net)), without which many of the patterns discussed in this book would not have been discovered. Thanks to Chris Rowett for creating LifeViewer ([lazyslug.com/lifeviewer](http://lazyslug.com/lifeviewer)), which is

<sup>1</sup>For an explanation of how RLE or Macrocell code works, see [conwaylife.com/wiki/Run\\_Length\\_Encoded](http://conwaylife.com/wiki/Run_Length_Encoded) or [conwaylife.com/wiki/Macrocell](http://conwaylife.com/wiki/Macrocell).

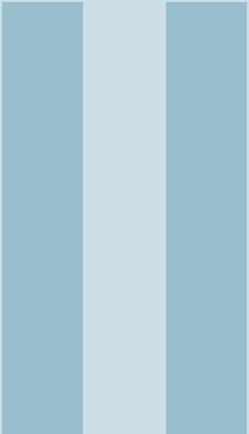
used on this book's website ([conwaylife.com/book](http://conwaylife.com/book)) to display patterns and make them interactive. Thanks to Velimir Gayevskiy and Mathias Legrand for the *Legrand Orange Book* LaTeX template from [LaTeXTemplates.com](http://LaTeXTemplates.com).

Finally, the authors would like to thank their wives Kathryn and Melanie for tolerating them during their years of mental absence glued to this book, and their parents for encouraging them to care about both learning and teaching. Many thanks also to Mount Allison University for giving the first author the academic freedom to pursue a project like this one.



# Classical Topics



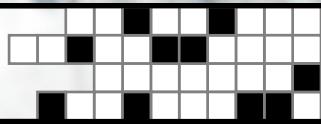


# Circuitry and Logic

<b>1</b>	<b>Glider Synthesis</b>	.....	5
1.1	Two-Glider Syntheses		
1.2	Syntheses Involving Three or More Gliders		
1.3	Incremental Syntheses		
1.4	Synthesis of Moving Objects		
1.5	Developing New Syntheses		
1.6	A Gosper Glider Gun Breeder		
1.7	Slow Salvo Synthesis		
1.8	Notes and Historical Remarks		
	Exercises		



# 1. Glider Synthesis



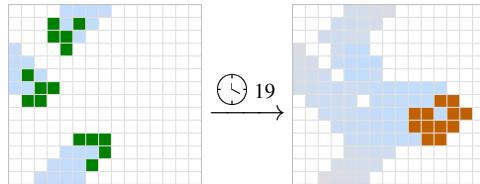
Life isn't about finding yourself. Life is about creating yourself.

---

George Bernard Shaw

In previous chapters, we saw several different patterns that were capable of generating an endless supply of gliders: glider guns like the Gosper glider gun create streams of gliders coming from a fixed location, and rakes like the space rake create waves of gliders that come from a moving source. We have also seen a few patterns (mostly based on the switch engine) capable of creating other objects like blocks or other small still lifes.

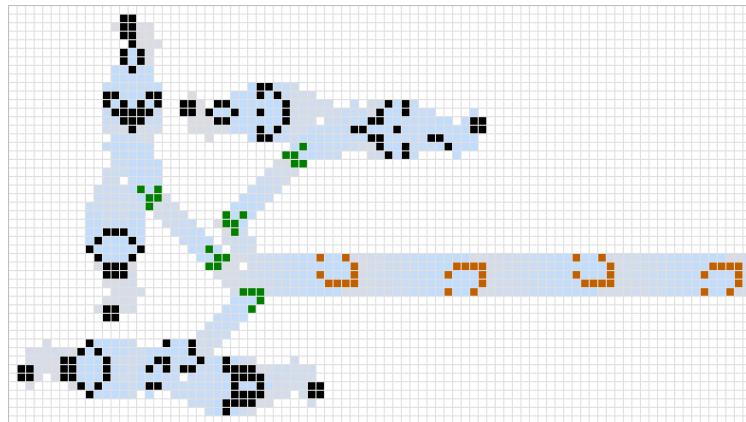
In this chapter, we develop a systematic method of constructing patterns that turn these simple objects like gliders and blocks into more complicated objects. In particular, we look at how to collide gliders with each other and with other objects so as to create (or “synthesize”) new ones. For example, Figure 1.1 illustrates a method of colliding three gliders so as to produce a lightweight spaceship.



**Figure 1.1:** Three gliders colliding in such a way as to create a lightweight spaceship.

We call this a *3-glider synthesis* of the lightweight spaceship, and it is useful because we already know of some patterns (i.e., glider guns) that generate gliders, so we can now create patterns that generate lightweight spaceships: we can just aim the output of three glider guns so as to collide with each other in the right way. For example, Figure 1.2 uses three Gosper glider guns to create a period 30 lightweight spaceship gun.

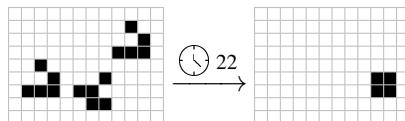
There is nothing particularly special about the lightweight spaceship in this example—we will see throughout this chapter that we can collide gliders from glider guns to create a wide variety of moving objects. We can also use gliders to synthesize stationary objects, but for this the glider source has to be moving in order to prevent subsequent gliders from colliding with the synthesized object. Indeed, once we have a large catalogue of glider syntheses, the world of Life will open up considerably for



**Figure 1.2:** A lightweight spaceship gun constructed by using three Gosper glider guns (top-right, top-left, and bottom-left) to shoot three gliders at each other, which collide in the orientation depicted in Figure 1.1 in order to create a lightweight spaceship.

us—we will be able to create patterns that construct almost any object in almost any location on the Life plane that we like.

As one somewhat technical note before we proceed, we require that the gliders in a synthesis could arrive at their positions from arbitrarily far away, since our ultimate goal is to make use of these syntheses via guns and rakes. An example of a glider collision that is *not* considered a true glider synthesis, since there is no way to get the gliders in the indicated positions, is provided in Figure 1.3.



**Figure 1.3:** A collision of three gliders that produces a block. However, if the two rightmost gliders came from farther away, they would collide with each other before reaching the displayed configuration, so this is not a valid 3-glider synthesis.

## 1.1 Two-Glider Syntheses

Once we have a glider synthesis in hand, it is typically straightforward to make use of it, since glider streams can typically be created just by lining up glider guns or rakes in the right spots. But how can we come up with glider syntheses in the first place? For example, how was the three-glider synthesis of a lightweight spaceship in Figure 1.1 found? The simplest answer, as unsatisfying as it might seem, is to just collide a bunch of gliders together in different ways and catalog which collisions lead to which objects. Once we have a good selection of “simple” objects that we know how to synthesize, we can then try colliding gliders with those objects to create more complicated patterns, and so on.

To begin down this road, we look at the simplest type of glider synthesis we can perform: synthesis using a collision of only two gliders. We already saw a few two-glider collisions in Section ?? when we aimed two glider waves at each other so as to destroy some gliders and reflect others, and again in Section ?? when we aimed two glider waves at each other to create a bi-block wick. Fortunately, cataloging all two-glider syntheses is not too difficult, since there are only 71 different ways that two gliders can collide.<sup>1</sup> A full summary of exactly which two-glider collisions lead to which objects being created is provided by Table 1.1.

While the objects that we can create with just two gliders are not particularly exciting, many of them are essential building blocks of complex patterns, such as eater 1 and the B-heptomino (which

<sup>1</sup>There does not seem to be a nice “mathematical” way to arrive at the number 71 here: it’s just a matter of arranging gliders in all possible different orientations and phases and seeing which ones result in collisions.

Result	2-Glider Collisions						Result	Collision
block								
honey farm								
blinker								
traffic light								
pi-hept.								
B-hept.								
loaf + blinker								
glider								
pond								
misc.								
nothing								

**Table 1.1:** A summary of the results of all 71 possible 2-glider collisions. The four “misc” collisions yield somewhat messy combinations of common objects like blocks and blinkers. The rightmost of the “misc” collisions (highlighted in red) is sometimes called the *two-glider mess*, as it takes 530 generations to stabilize—more than any of the other collisions. The left glider-producing collision (highlighted in yellow) is called the *kickback reaction*, since it produces an output glider traveling in a different direction than either of the input gliders.

we recall evolves into a Herschel). It is also worth pointing out that the two collisions that result in a single glider can in fact sometimes be useful. In particular, the one of these collisions that is on the left in Table 1.1 is called the *kickback reaction*, and it is useful for that fact that the output glider travels in a different direction than either of the two input gliders. This feature can help us navigate gliders around tight spots and simplify many complicated glider syntheses (see Exercise 1.13, for example).

## 1.2 Syntheses Involving Three or More Gliders

When moving from two-glider syntheses to three-glider syntheses, things become much more complicated—it is no longer possible to list all collisions and catalog their output, since there are far too many possibilities. To see why this is, recall that the two-glider mess takes 530 generations to stabilize, and we could fire a third glider from dozens of different positions and hundreds of different timings to collide with that chaotic mess. Even worse, that two-glider mess produces multiple gliders, which we could hit with the third glider after an arbitrarily long period of time so as to synthesize a new object as far away as we like.

On the other hand, being unable to catalog all of these three-glider collisions is perhaps not too much of a loss, since we expect that the majority of them would lead to uninteresting combinations of blocks, blinkers, and other common objects that we already know how to synthesize with just two gliders. We thus only present, in Table 1.2, some of the particularly useful three-glider syntheses of more exotic objects. However, we stress that this table is not complete: there are known three-glider syntheses not presented in this table, and it is entirely possible that there are simple objects with three-glider syntheses that have not yet been found.<sup>2</sup>

Although the 3-glider syntheses of a single glider from Table 1.2 (which are called *tees*) might seem somewhat silly at first glance, they have the useful property that the synthesized glider travels perpendicular to each of the input gliders<sup>3</sup>—a property that is not shared by any of its 2-glider syntheses. For this reason, these syntheses are actually quite useful when trying to manipulate glider positions or reduce the number of directions used in glider syntheses of more complicated objects (see Exercise 1.15, for example).

The three-glider collision that creates a glider-producing switch engine is quite exciting, as it is our first example of a synthesis of an infinitely-growing pattern. Being able to synthesize queen bees is similarly exciting, since a Gosper glider gun is made up of nothing more than two blocks and two queen bees, all of which we now know how to synthesize. By simply synthesizing all of these objects in the correct positions and phases, we are now able to use gliders to synthesize a pattern that creates additional gliders (see Figure 1.4),<sup>4</sup> and we could conceivably use rakes to create Gosper glider guns that in turn create other patterns.

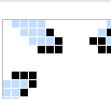
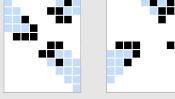
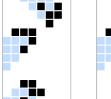
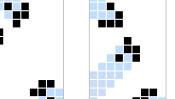
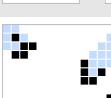
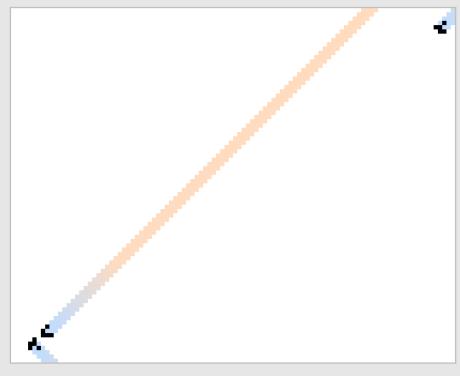
We have a fairly wide variety of objects that we can now synthesize with gliders, but there are still some rather fundamental objects that are missing from our lists of 2- and 3-glider syntheses. For example, if we wanted to synthesize a twin bees gun, we would first have to know how to synthesize twin bees, which (as far as we know) requires at least 4 gliders. It is also sometimes beneficial to be aware of glider syntheses that make use of more gliders, even when syntheses involving fewer gliders exist. For example, the 3-glider synthesis of the heavyweight spaceship given in Table 1.2 is actually quite difficult to make use of, since the two gliders coming in from the top-right are so close together that they cannot be generated by many of the glider sources that we have seen (like adjacent Gosper glider guns).

---

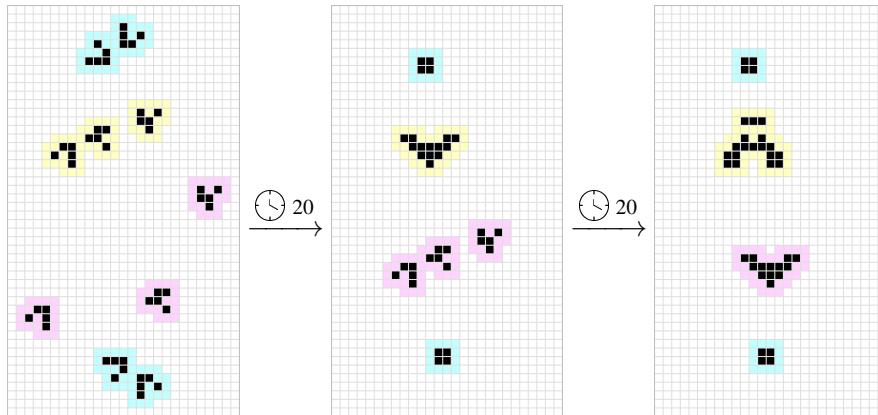
<sup>2</sup>For example, the pentadecathlon and switch engine were not known to be synthesizable via only three gliders until the syntheses given in Table 1.2 were found by Heinrich Koenig and Luka Okanishi in April 1997 and March 2017, respectively. Similarly, it was not known how to create *any* infinitely-growing object using just 3 gliders until Michael Simkin found a 3-glider collision that produces a glider-producing switch engine in October 2014.

<sup>3</sup>In fact, both of these syntheses work by colliding two of the gliders so as to create a banana spark, which then reflects the third glider as in Figure ??.

<sup>4</sup>Synthesizing the Gosper glider gun “piece-by-piece” like this is the quite straightforward, and requires 10 gliders. However, slightly smaller syntheses of the Gosper glider gun are known, requiring as few as 8 gliders.

Result	3-Glider Collisions	
 glider		
 lightweight spaceship		
 middleweight spaceship		
 heavyweight spaceship		
 pentadecathlon		
 pulsar		
 queen bee		
 R-pentomino		
 switch engine		
glider-producing switch engine (plus junk)		

**Table 1.2:** A selection of useful 3-glider syntheses. The 3-glider collision that creates a glider-producing switch engine is not a true glider synthesis due to the fact that it also creates a wide assortment of other debris, but it is nonetheless noteworthy for being the only known way of generating infinite growth with just 3 gliders.



**Figure 1.4:** A ten-glider synthesis of a Gosper glider gun, which is composed of two syntheses of blocks (highlighted in aqua) and two syntheses of queen bees. The queen bee synthesis highlighted in magenta is the same as the one highlighted in yellow, but delayed by 20 generations in order to make the queen bees collide in their proper phases.

For this reason, we often prefer glider syntheses that use widely-spaced gliders, even if that comes at the expense of a larger number of gliders. In particular, there are 4-glider syntheses of a heavyweight spaceship that consist of one glider coming from each direction, and thus are often much easier to make use of. We present a summary of useful syntheses that involve 4 or more gliders in Table 1.3.

Result	Glider Syntheses	Result	Synthesis
clock		twin bees	
HWSS		eater 2	

**Table 1.3:** Some useful glider syntheses involving 4 or more gliders. Most of these syntheses have been known for a long time, but the eater 2 synthesis was found by Tanner Jacobi in November 2014 (previously no synthesis with fewer than 17 gliders was known, though there were syntheses of slight variants of eater 2 with as few as 8 gliders).

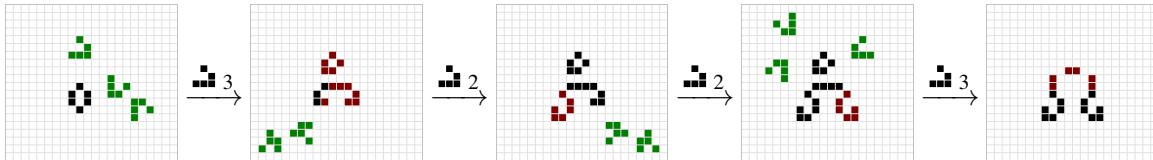
### 1.3 Incremental Syntheses

Glider syntheses can quickly become cumbersome to use as the number of gliders involved increases. While the guns and rakes that we have already developed let us fire two streams of gliders that collide in any of the 71 possible ways illustrated in Table 1.1, as soon as three or more gliders are involved, it can be quite difficult to coordinate them so that they all arise at the desired time. For example, it would be challenging to implement the ten-glider synthesis of a Gosper glider gun from Figure 1.4 “directly”, as it would require to coordinate the precise positioning and timing of all ten gliders.

To get around this problem, we make use of *incremental synthesis*: we build up complicated patterns by first synthesizing a small stable pattern (like a block or a pond), and then colliding only a few gliders with that stable pattern to create a slightly more complicated stable pattern, and so on until we arrive at the pattern we actually want. The main benefit of this type of synthesis is that we do not

need to be particularly precise with the timing of the gliders—the different stages of synthesis can take place as many generations apart from each other as we like, so we only need to coordinate a few gliders at a time.

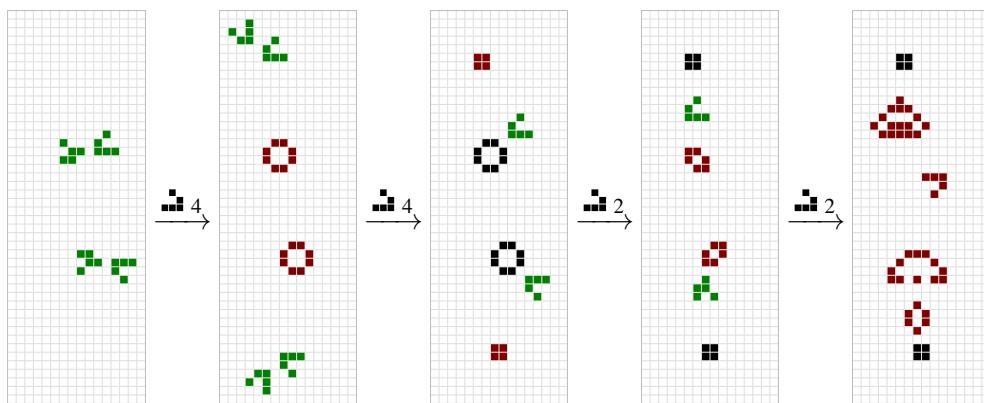
As an example, consider the incremental synthesis of the fumarole that is presented in Figure 1.5. While there is a glider synthesis for the fumarole that involves only 7 gliders (see Exercise 1.1), this 12-glider synthesis is somewhat easier to make use of, since each stage of its synthesis requires no more than 3 synchronized gliders.



**Figure 1.5:** A 12-glider incremental synthesis of a fumarole (2 gliders that are not shown are required to synthesize the initial beehive). The cells that were created or modified in the previous step of the synthesis are shown in red, and the gliders that will be involved in the next collision are shown in green.

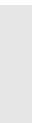
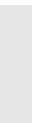
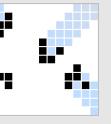
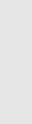
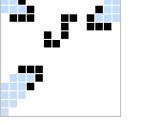
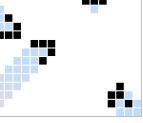
Incremental syntheses are also advantageous for the fact that we do not have to find ways of synthesizing complicated patterns from scratch, but rather we can keep a catalog of syntheses of various objects, and then to synthesize a new object we can use the most similar still life that we already know how to synthesize as our starting point. For example, the incremental synthesis of the fumarole in Figure 1.5 works by synthesizing a still life that looks similar to the stator of the fumarole (i.e., its stable bottom half), and then the last step of the synthesis creates the fumarole’s rotor (i.e., its oscillating top half). This is the most commonly-used technique for synthesizing oscillators, and generally billiard table oscillators are much more difficult to synthesize than other oscillators precisely because their rotors are contained within their stators and thus this synthesis technique does not work for them.

By using the reactions listed in Table 1.4, it is straightforward to come up with an incremental synthesis of more complicated objects as well. For example, we can synthesize a Gosper glider gun in a way that never requires more than two synchronized gliders at a time by using incremental syntheses to turn a pond into a ship into a queen bee, as in Figure 1.6. We note that this synthesis uses a total of 12 gliders instead of the 10 gliders used in Figure 1.4, but we consider this a small price to pay for the benefit of not having to synchronize as many gliders in any of the steps of synthesis.



**Figure 1.6:** An incremental synthesis of the Gosper glider gun. Each stage in the synthesis works by either using two gliders to construct a simple still life or using just one glider to transform one object into another one. The objects that were created or modified in the previous step of the synthesis are shown in red, and the gliders that will be involved in the next collision are shown in green.

Because incremental synthesis typically works by first constructing a still life or a collection of still lifes, and then transforming those still lifes into the desired object, the Life community has gone to great lengths to synthesize still lifes. In fact, syntheses are known for all 78,297 strict still lifes with

Result	Last Step of Incremental Synthesis		
			
tub			
			
honey farm			
			
queen bee			
			
hat			
			
ship			
			
tub with tail			
			
snake			
			
T-tetromino / traffic light			
			
pentadecathlon			
			
LWSS			
			
MWSS			
			
HWSS			

**Table 1.4:** A selection of useful incremental syntheses that can be used to construct many of the simple Life objects that we have seen. Note that the arrangement of a loaf and a blinker in the HWSS incremental synthesis is the exact arrangement produced by the 2-glider syntheses of a loaf and blinker in Table 1.1.

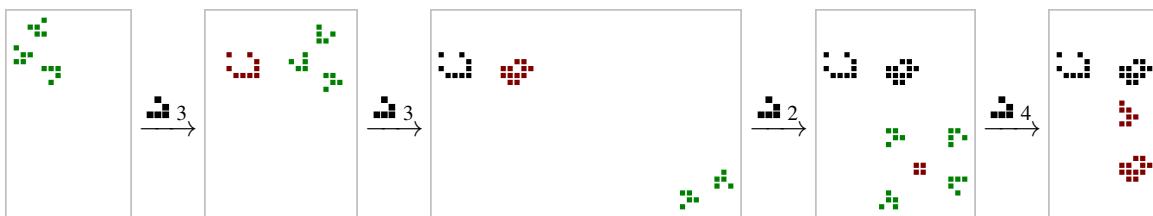
19 or fewer live cells.<sup>5</sup>

## 1.4 Synthesis of Moving Objects

Incremental synthesis is most useful when trying to construct objects that are made up of many simpler objects. We already demonstrated this fact when constructing glider syntheses of the Gosper glider gun, and some other perfect examples include many of the spaceships, puffers, and rakes that we introduced in Chapter ???. These objects are mostly composed of many copies of small objects like lightweight spaceships, B-heptominoes, and switch engines that we already know how to synthesize, so we can just synthesize each component individually. However, it can sometimes be tricky to construct the various bits and pieces of these spaceships and puffers with timing that works, so we make these syntheses explicit.

### 1.4.1 The Ecologist and Space Rake

Recall that the ecologist of Figure ?? is made up of three lightweight spaceships and a B-heptomino, so we can synthesize it just by synthesizing those four individual pieces. The only part of the synthesis that is somewhat complicated is the construction of the B-heptomino, since it is not stable without a lightweight spaceship on both sides of it (so we cannot place it first), but there is not enough room for gliders to synthesize it between already-placed lightweight spaceships (so we cannot place it last). One solution is to synthesize the B-heptomino at the same time as the final lightweight spaceship, as in Figure 1.7.



**Figure 1.7:** An incremental synthesis of the ecologist. The first two steps simply use one of the three-glider syntheses of lightweight spaceships, and the third step uses a two-glider synthesis of a block. The fourth step is slightly more complicated, since we have to construct the B-heptomino at the same time that we construct the final lightweight spaceship below it. The top two gliders form the B-heptomino, while the bottom two gliders and the block construct a lightweight spaceship via one of the incremental syntheses that we saw in Table 1.4.

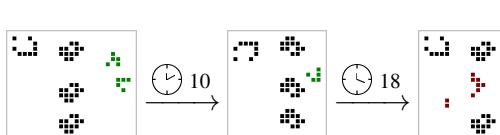
There is another way of synthesizing the ecologist that has the advantages of never using more than two synchronized gliders at a time, and only ever using gliders that come from two of the four possible directions. The way that this synthesis works is to first create four lightweight spaceships (three in the positions that we want and one where the B-heptomino should be), each using a two-glider synthesis of a block followed by the two-glider block-to-LWSS synthesis from Table 1.4. Then we can fire one additional glider at the central lightweight spaceship to transform it into the desired B-heptomino (to actually get this final glider in the proper orientation though, we make use of the 2-glider kickback reaction from Table 1.1). The tricky final stage of this synthesis is made explicit in Figure 1.8.

Once we have constructed the ecologist, it is straightforward to turn it into the space rake or its backward variant just by synthesizing the extra lightweight spaceship in the correct position, as in Figure 1.9 (also see Exercise 1.19).

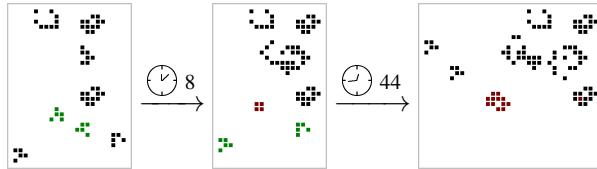
### 1.4.2 The Schick Engine and Coe Ship

To create a glider synthesis of the Schick engine from Section ??, we recall that it is made up of two lightweight spaceships, followed by a spark that is essentially just a T-tetromino (see Exercise ??).

<sup>5</sup>A database containing all of these syntheses of small still lifes, as well as incremental syntheses of many other objects, as available at [catagolue.appspot.com/syntheses](http://catagolue.appspot.com/syntheses).

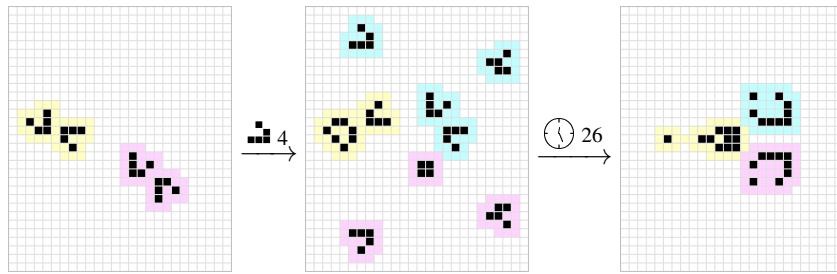


**Figure 1.8:** Another incremental synthesis of the ecologist. This synthesis has the advantage of never using more than two gliders at a time, and they can all arrive from the lower-left and lower-right (see Exercise 1.17).



**Figure 1.9:** An incremental glider synthesis of a (forward) space rake. The same technique can be used to synthesize the backward space rake (see Exercise 1.19).

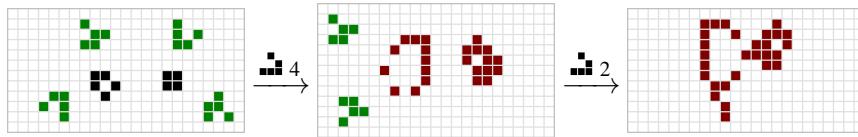
The various syntheses of lightweight spaceships and T-tetrominoes that we have seen can thus be used together in a straightforward manner to synthesize the Schick engine, as in Figure 1.10. The only difficulty comes from having to carefully choose ways of synthesizing the individual components so that they do not interfere with each other.



**Figure 1.10:** A glider synthesis of a Schick engine. Lightweight spaceships are created by the magenta and aqua glider collisions, and a T-tetromino (which becomes the Schick engine's trailing spark) is created by the gliders outlined in yellow.

With this synthesis of the Schick engine in hand, we are also now able to synthesize the period 60 variants of the space rake that we saw in Section ??—we just synthesize the space rake and then synthesize a Schick engine next to it in the appropriate position, or vice-versa (see Exercise 1.18(b)).

Synthesizing the Coe ship is slightly less straightforward, since we have not yet seen how to synthesize the deformed heavyweight spaceship that makes up half of it. Perhaps the simplest way to synthesize this object is to first synthesize a lightweight and middleweight spaceship next to each other, and then use two additional gliders to transform the MWSS into the deformed HWSS, as illustrated in Figure 1.11.



**Figure 1.11:** An incremental glider synthesis of a Coe ship that works by synthesizing a lightweight spaceship next to a middleweight spaceship, and then transforming that middleweight spaceship into the necessary deformed heavyweight spaceship.

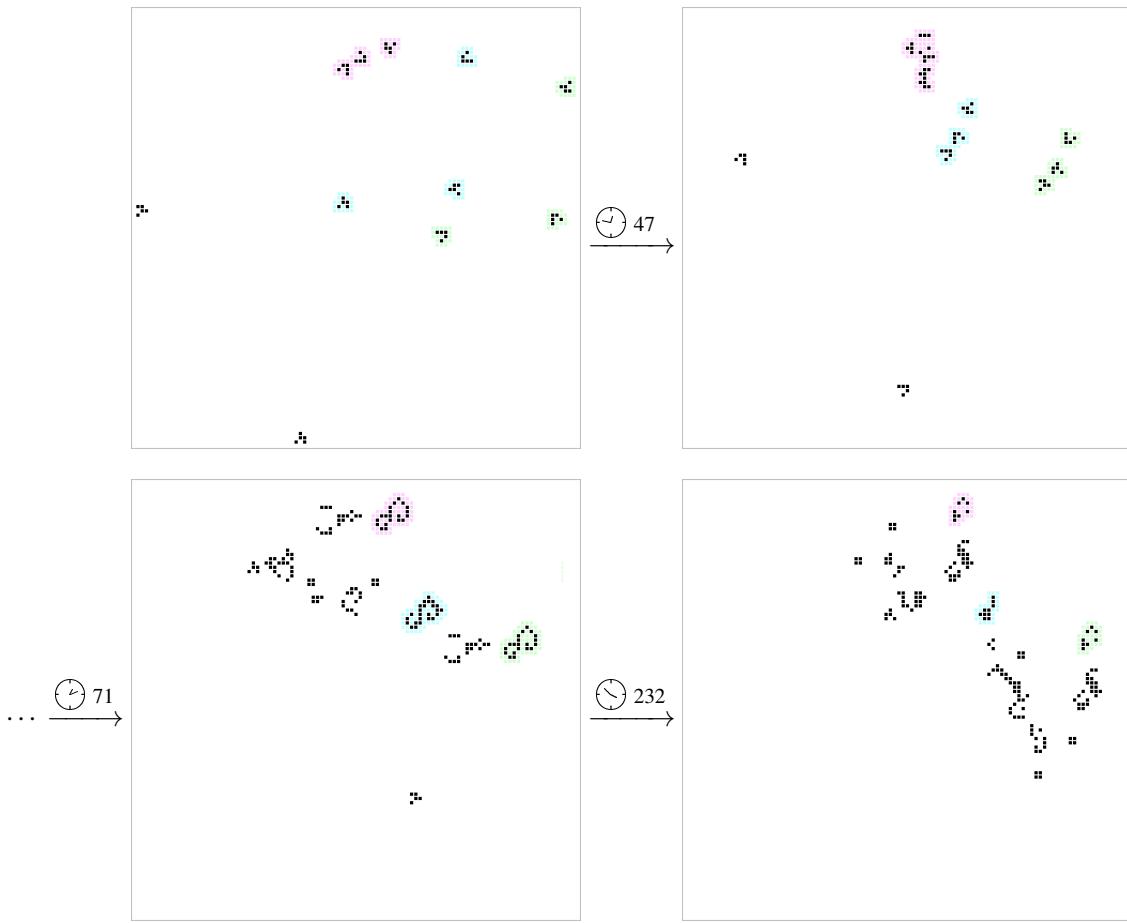
### 1.4.3 Corderships

Corderships make up another family of objects that are prime candidates for glider synthesis, since they are also made up of several easily-synthesized components—switch engines. The difficult part of synthesizing a Cordership is that the switch engines have to be synchronized with each other, making an incremental synthesis difficult. Since each switch engine requires at least three gliders to synthesize, we expect the 10-engine Cordership from Figure ?? to require at least 30 gliders to synthesize. Furthermore, the 6 front and middle switch engines will have to be constructed at almost

the exact same time as each other.<sup>6</sup>

With these difficulties in mind, it seems desirable to try synthesizing a Cordership that uses as few switch engines as possible—the 2-engine Cordership from Exercise ?? or the 3-engine Cordership from Figure ?? . Here we focus on the 3-engine Cordership, and we leave the task of synthesizing its 2-engine counterpart to Exercise 1.14.

As expected, it is not too difficult to use  $3 \times 3 = 9$  gliders to synthesize the three switch engines that make up this Cordership. However, one small problem arises after we use these 9 gliders: while the Cordership is indeed synthesized correctly, the switch engines take a few generations to sync up with each other, so some unwanted debris is left behind. To prevent this debris from forming, we use 2 extra gliders—one for each of the outermost switch engines—to suppress some chaotic reactions that are normally suppressed by the central switch engine. Altogether, this results in the 11-glider synthesis of the 3-engine Cordership displayed in Figure 1.12.



**Figure 1.12:** An 11-glider synthesis of the 3-engine Cordership. The 3 switch engines are synthesized by the 3-glider syntheses highlighted in aqua, magenta, and green. The remaining 2 gliders clean up some extra junk that would otherwise be left behind by the outermost switch engines.

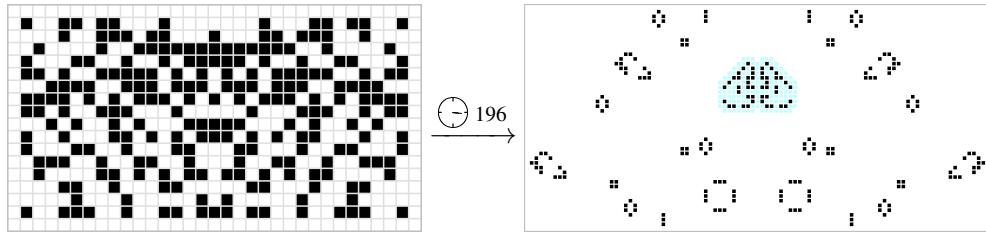
## 1.5 Developing New Syntheses

So far the glider syntheses that we have seen have been restricted to two basic types: (a) syntheses that were found just by crashing gliders together and cataloguing what objects were created, and (b) syntheses that can be made simply by piecing together the syntheses of type (a). We now branch

<sup>6</sup>Despite these obstacles, Stephen Silver created a 60-glider synthesis of the 10-engine Cordership in 1998 (using a 4-glider synthesis of a switch engine, since the 3-glider synthesis was not yet known).

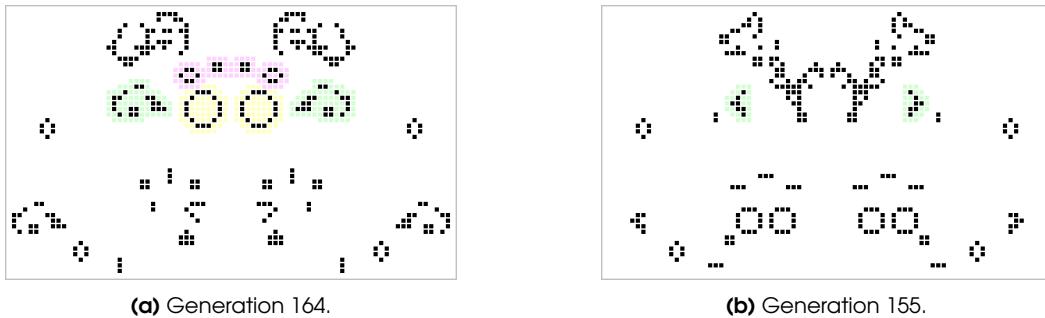
out and demonstrate how to construct glider syntheses of more exotic objects that do not decompose in any natural way into simpler objects that we already know how to synthesize.

Perhaps the most common way to develop new glider syntheses is to find a soup that leaves behind the pattern of interest in its ash, and try to reverse-engineer the part of the soup's evolution that led to the object's formation. To illustrate this method, consider Rich's p16 (the period 16 oscillator that we introduced in Figure ??), which was found by evolving the soup displayed in Figure 1.13.



**Figure 1.13:** The soup that led to the discovery of Rich's p16 (highlighted on the right in aqua).

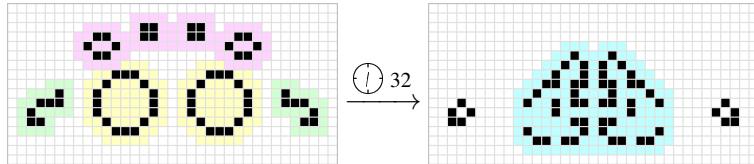
To develop a glider synthesis for this oscillator, we evolve the soup to just before its formation (in this case, slightly earlier than generation 196) and try to isolate the reaction that creates the oscillator. In this step, it is important to keep in mind that the goal is to go back far enough that all (or at least most) of the reactions are ones that we are familiar with, such as a T-tetromino or a queen bee colliding with some still lifes. In this particular case, if we go to generation 164 of the soup's evolution then the reaction has decomposed enough that we can identify all of the important pieces of the oscillator's creation: Rich's p16 (plus a lot of extra debris) is formed when a particular configuration of two honey farm predecessors and two B-heptominoes collide with two blocks and two beehives (see Figure 1.14).



**Figure 1.14:** Rich's p16 is created (a) when two honey farm predecessors (highlighted in yellow) and two B-heptominoes (highlighted in green) collide with two blocks and two beehives (highlighted in magenta). The honey farm predecessors can be identified by comparing them with Figure ??, and the B-heptomino can be identified by (b) going backward to generation 155 of the soup.

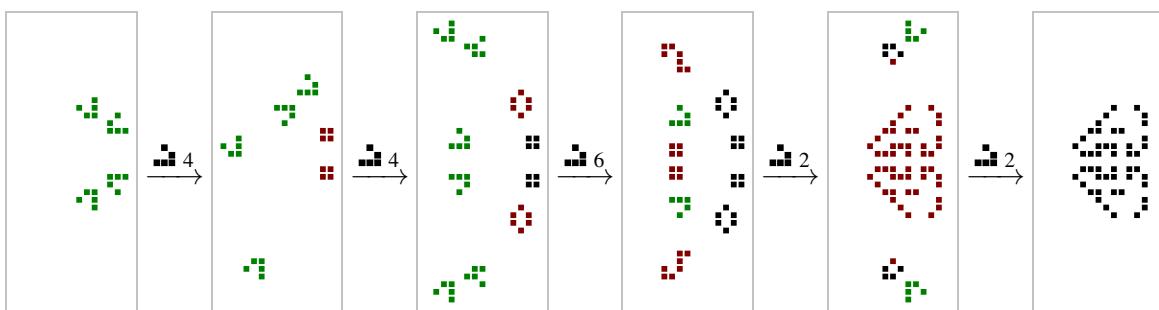
We now have enough tools at our disposal to synthesize this oscillator, since we know how to synthesize blocks, beehives, honey farms, and B-heptominoes. However, the B-heptominoes in this reaction are quite messy, and create a lot of leftover debris that we would then have to clean up with several additional gliders (see Exercise 1.2). To reduce the number of required gliders somewhat, we notice that only three cells on one side of the B-heptominoes interact with the rest of the reaction, and only for two generations. Another easy-to-synthesize object with the same configuration of three cells is the eater 1, which can thus replace the B-heptominoes as in Figure 1.15. Fortunately, using eater 1s in this way does result in a much smaller amount of debris being created around the Rich's p16.

The only somewhat tricky part of synthesizing this collection of objects is creating the two honey farm predecessors, since they are unstable and we thus have to create them after creating the still lifes. For this reason, we opt to use the incremental honey farm synthesis from Table 1.4, so that instead of having to use 4 coordinated gliders in the final stage of synthesis (2 for each honey farm), we only need to use 2 coordinated gliders (plus 2 blocks that we can place much earlier). A complete



**Figure 1.15:** A reaction of common objects that results in Rich’s p16 (and two unwanted boats). This is the same reaction as in Figure 1.14(a), but with the B-heptominoes replaced by eater 1s (highlighted in green).

incremental synthesis of this oscillator is presented in Figure 1.16.<sup>7</sup> We stress that there is nothing particularly clever about this synthesis; it just makes use of the 2-glider syntheses from Table 1.1, the incremental honey farm synthesis we already mentioned, and one final glider collision (which is easily found by hand) to destroy the left-over boats.



**Figure 1.16:** An 18-glider incremental synthesis of Rich’s p16. The first three stages of synthesis just involve placing still lifes that can each be synthesized by 2 gliders. The next stage synthesizes the honey farm predecessors, and the final stage uses 2 additional gliders to destroy the 2 leftover boats. Objects that were created or modified in the previous step of the synthesis are shown in red, and the gliders that will be involved in the next collision are shown in green.

## 1.6 A Gosper Glider Gun Breeder

We have now learned quite a few recipes for synthesizing different objects with gliders. To demonstrate how useful these glider syntheses can be, we now switch focus a bit and start *using* these glider syntheses to create new types of patterns. In particular, we now construct our first *breeder*: an object that grows quadratically (as opposed to objects like guns or rakes, which grow linearly). In other words, we will construct an object that does not just fill up some *strip* of the Life plane, but instead fills an ever-expanding *region* of the Life plane.

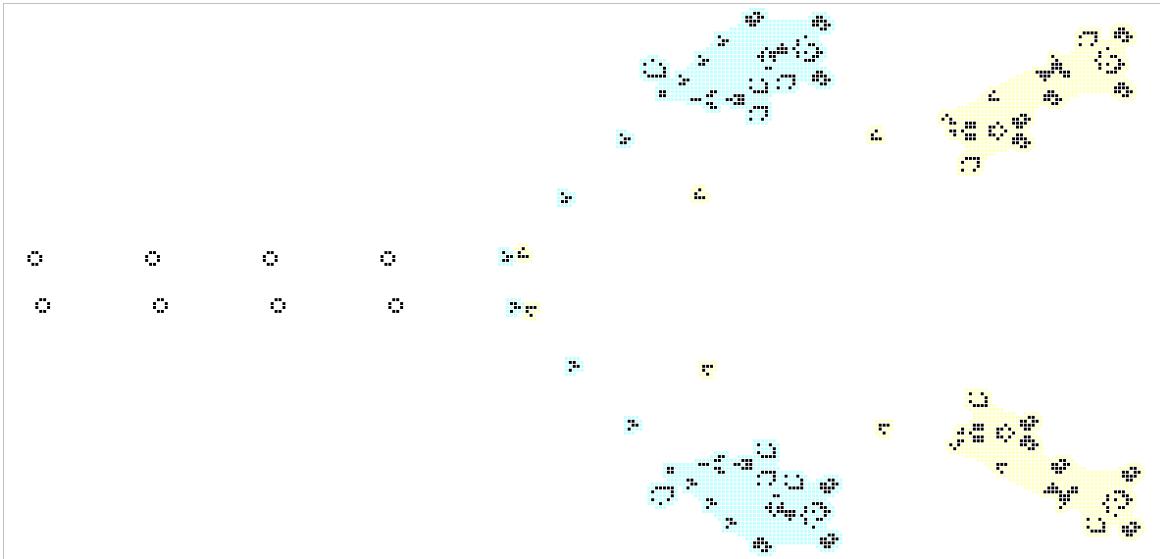
The idea behind this breeder is simple enough: we use the rakes that we introduced in Section ?? to fire gliders in such a way that those gliders synthesize Gosper glider guns (using the incremental synthesis from Figure 1.6). Since the rakes will create Gosper glider guns at a linear rate, and the Gosper glider guns will then each create gliders at a linear rate, the overall growth of our pattern will be quadratic.

To make this construction explicit, we first note that we cannot possibly use the period 20 space rakes directly, since they can only be used to construct objects that are 10 cells apart. Since Gosper glider guns are wider than that, they would collide and destroy each other. We thus make use of the slightly more complicated period 60 variants of the space rake that we introduced in Section ??, which will produce glider guns that are 30 cells apart from each other.

Now we just go through the Gosper glider gun’s incremental synthesis step by step, lining up rakes so that the gliders they produce collide in the desired way. For example, for the first step of

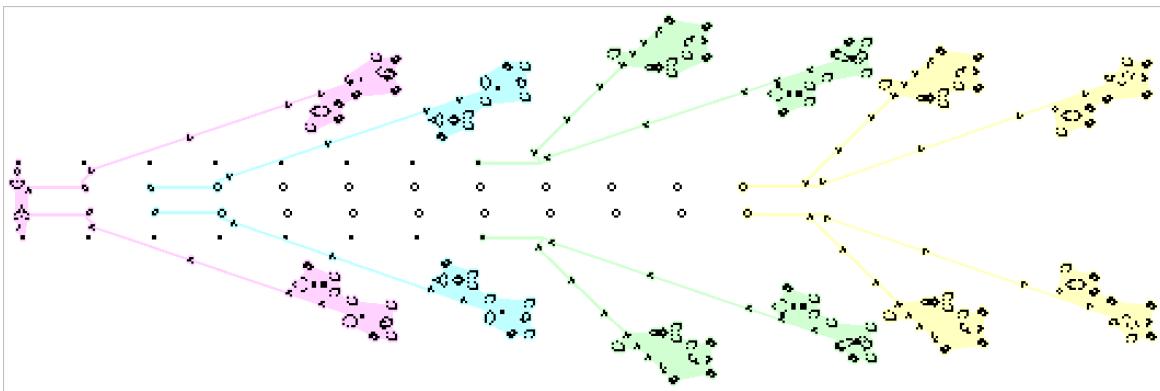
<sup>7</sup>This synthesis was originally found, using the method we outlined in this section, by user “BlinkerSpawn” on the ConwayLife.com forums less than a day after the oscillator’s discovery.

the synthesis (i.e., creating the two initial ponds), we need four rakes, which we can position as in Figure 1.17.



**Figure 1.17:** The front end of our breeder features two copies of the period 60 backward space rake (outlined in yellow), followed by two copies of the period 60 forward space rake (outlined in aqua), all traveling to the right. The gliders from these rakes collide in such a way as to create ponds, which is the first step of the incremental synthesis of a Gosper glider gun presented in Figure 1.6.

Once we have positioned those four rakes, they create two endless lines of ponds with a distance of 30 cells between each consecutive pair. At this point, we simply move on to the next portion of the glider synthesis: we construct two blocks near each pair of ponds, which we can do with four more rakes. We then need to hit the ponds with a single glider each in order to turn them into ships, which we do with two more rakes. Lastly, we use two final rakes to fire gliders at the pair of ships, turning each of them into queen bees and thus completing the Gosper glider gun synthesis. The resulting breeder is displayed in Figure 1.18.<sup>8</sup>

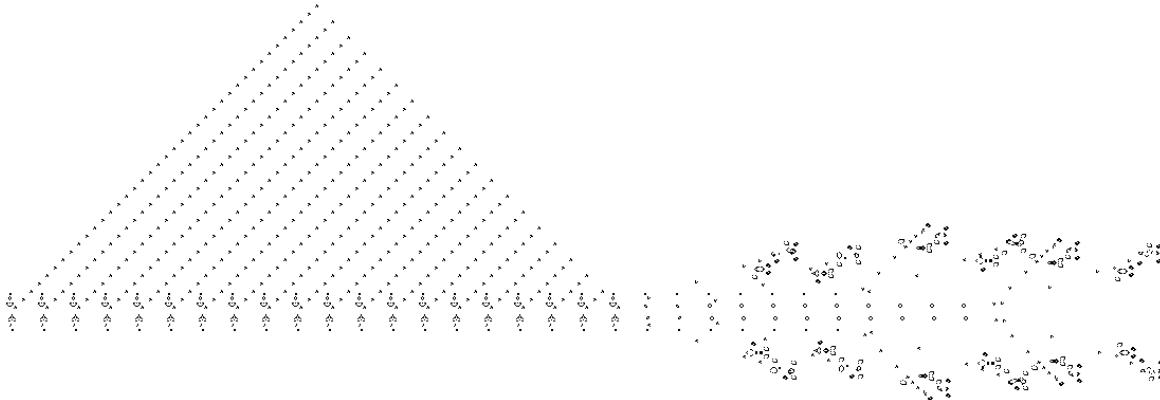


**Figure 1.18:** A breeder constructed using an incremental glider synthesis of a Gosper glider gun. First, a forward and backward p60 space rake create a pond as in Figure 1.17 (outlined on the right in yellow), followed by another forward and backward space rake creating a block (outlined in green). Next, a backward space rake converts the pond into a ship (outlined in aqua), and finally another backward space rake converts the ship into a queen bee (outlined in magenta), thus completing the synthesis of the Gosper glider gun.

In generation 0 (i.e., its starting phase), this breeder has a population of 2038 cells. The quadratic growth of this breeder can be quantified by noting that every 60 generations, an additional 44-cell

<sup>8</sup>The first breeder was found by Bill Gosper in the early 1970s. It used the exact same ideas that we used in the construction of our breeder, but used a different puffer in place of the space rakes that we used.

Gosper glider gun is created, and each of those guns creates two additional 5-cell gliders, for a total population in generation  $60n$  of  $5n^2 + 44n + 2038$ . This quadratic growth is illustrated in Figure 1.19, which shows the expanding triangular region of gliders behind generation 1200 of the breeder.



**Figure 1.19:** After running the breeder for 1200 generations, we see a triangle of gliders form at the west, above the line of Gosper glider guns that has been synthesized. As the breeder moves farther to the east, the triangle continues to expand to the northeast.

While this breeder is quite large, we have purposely not optimized it all, in order to make the mechanisms that make it work more apparent. The space rakes can be moved much closer to each other without compromising its functionality (see Exercise 1.23), and by cleverly manipulating the space rake debris it is actually possible to place more than one of the still lifes at the same time, thus halving the total number of space rakes from 12 to 6. We present and make use of a much more compact Gosper glider gun breeder that comes from these ideas a bit later, in Section ??.

## 1.7 Slow Salvo Synthesis

In Section 1.3, we saw that we can often break glider syntheses down into bits and pieces that are individually easy to understand and make use of. In this section, we take this idea to its most extreme by considering *slow salvos*, which are collections of gliders<sup>9</sup> that (a) are *slow*: only one glider interacts in the synthesis at a time, and (b) form a *salvo*: all of the gliders come from the same direction.

In order for the gliders in a slow salvo to be able to actually synthesize anything, we need another object (called a *seed*) for it to crash into. While we could in principle use any object, it is typical to use a block as a starting point, since it is the simplest and most symmetric stationary object. Furthermore, it is typical to only consider either *p1 slow salvos* or *p2 slow salvos*, which are slow salvos in which the intermediate objects that are created after every glider collision are still lifes or some combination of still lifes and period 2 oscillators, respectively.<sup>10</sup> This perhaps seems like quite an extensive list of restrictions, but we have in fact already seen a few objects that can be created via *p1 slow salvos*. For example, we saw in Table 1.4 that we could use a pond as a seed in a *p1 slow salvo* to produce a ship, and then a queen bee.

Despite how restrictive slow salvos seem at first, it is a remarkable fact that they are exactly as general as standard glider syntheses. That is, if we can synthesize an object with gliders at all then we can synthesize it with a *p2 slow salvo*.<sup>11</sup> In order to pin down exactly why this is the case, we now introduce several new reactions that allow us to systematically create slow salvo syntheses from standard glider syntheses.

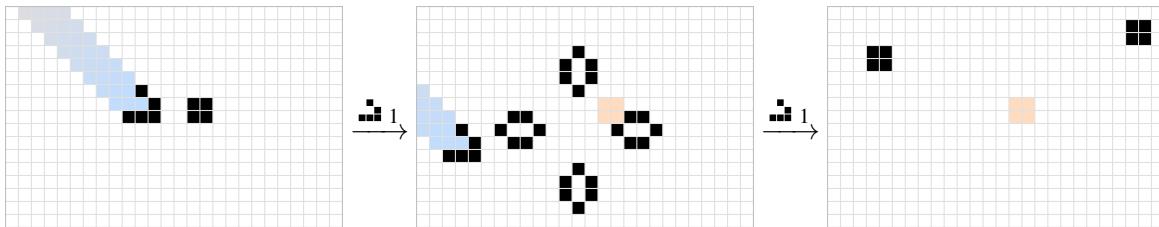
<sup>9</sup>Technically, a salvo can be made up of any spaceships, but the term almost always refers to gliders.

<sup>10</sup>P2 salvos are advantageous because blinkers are so easy to synthesize, and we will see shortly that the clock is also a useful tool when working with slow salvos. We could analogously consider *pn slow salvos* for some  $n \geq 3$ , but in practice not much is gained by doing so, since objects with period 3 or higher typically aren't any easier to create in intermediate stages of synthesis than objects period 1 or period 2.

<sup>11</sup>However, the slow salvo might contain considerably more gliders than other glider syntheses.

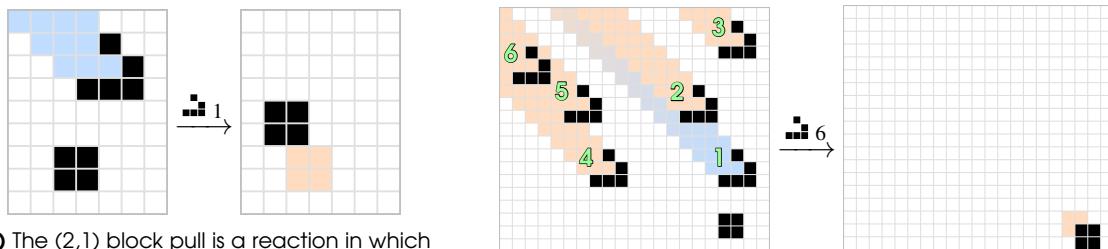
### 1.7.1 Creating and Moving Blocks

Our first step toward showing that slow salvos can construct anything that regular glider syntheses can is to demonstrate how we can create almost any arrangement of any number of blocks in the Life plane that we like. The first reaction that we will need is the one displayed in Figure 1.20, which uses a slow salvo of 2 gliders to turn a single block into two blocks.



**Figure 1.20:** A 2-glider slow salvo can turn one block into two blocks.

While the two resulting blocks are in very different positions than the initial block, it turns out that this does not particularly matter, as we can also use p1 slow salvos to move a block from one position on the Life plane to any other. To show that such a movement is possible, we use the two block-moving reactions displayed in Figure 1.21. The first of these reactions uses a single glider to move a block up 2 cells and left 1 cell (in fact, this is exactly the (2,1) block pull that we saw way back in Figure ??), and the second reaction uses six gliders (in a p1 slow salvo) to move a block down and to the right 1 cell.



(a) The (2,1) block pull is a reaction in which a single glider pulls a block by 2 cells in one direction and 1 cell in the other.

(b) A 6-glider slow salvo moving a block down and right by 1 cell.

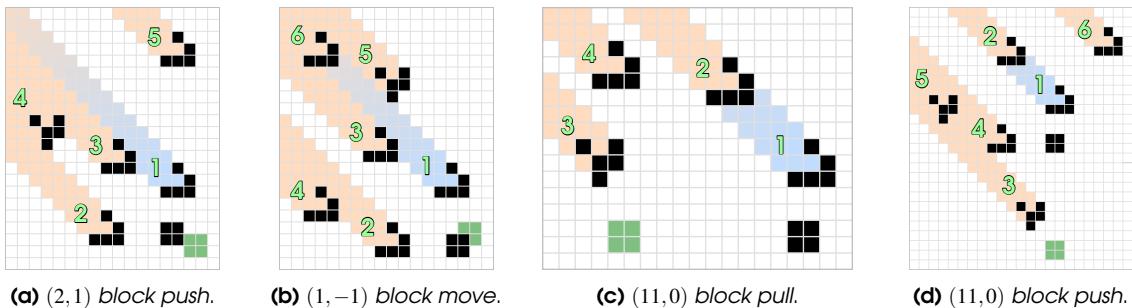
**Figure 1.21:** Two methods of moving a block around with p1 slow salvos. The order in which the gliders should come in is indicated by the green numbers—their exact timing does not matter, as long as they are far enough apart that each collision settles down before the next glider arrives. Repeating these slow salvos allows us to move a block to any position on the Life plane.

If we repeat these two reactions then we can move a block a single cell up, down, left, or right. For example, to move a block up a single cell, we could perform the movement in Figure 1.21(a), followed by the movement in Figure 1.21(b) (for a total cost of 7 gliders). Similarly:

- To move a block right by a single cell, we could perform the movement in Figure 1.21(a), followed by the movement in Figure 1.21(b) twice.<sup>12</sup>
- To move a block left by a single cell, we could perform the movement in Figure 1.21(a), but reflected along the diagonal from the top-left to the bottom-right. Then perform the movement in Figure 1.21(b).
- To move a block down by a single cell, we could first move it left by a single cell and then perform the movement in Figure 1.21(b).

<sup>12</sup>This requires a total of 13 gliders, which is far from optimal—another p1 slow salvo is known that accomplishes the same movement via only 7 total gliders. However, we are just interested in presenting the conceptually simplest method of moving blocks, not necessarily the most efficient one. Paul Chapman and Dave Greene put together an extensive table of the cheapest known block-moving slow salvos, which is available at [b3s23life.blogspot.com/2004/09/glues-slow-salvo-block-move-table.html](http://b3s23life.blogspot.com/2004/09/glues-slow-salvo-block-move-table.html) (note that the values in their table assume that the gliders come from the bottom-right, rather than the top-left).

Now that we know how to use p1 slow salvos to move a block by a single cell in any direction, we can easily repeat these reactions over and over until the block is moved to whatever position we desire. However, there are a few other block-moving slow salvos that are worth making note of, since they help us move blocks around the plane a bit more quickly. In particular, if we are willing to make use of p2 salvos instead of just p1 ones, then we gain access to the salvos displayed in Figure 1.22. There are hundreds of salvos of this type known.<sup>13</sup> It is worth noting that the (2, 1) *block push* of Figure 1.22(a) *pushes* the block by the same amount that the (2, 1) block pull of Figure 1.21(a) pulls it.

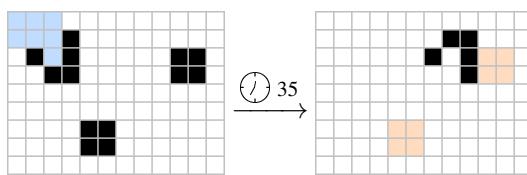


**Figure 1.22:** Some p2 slow salvos that move a block by various amounts. The output location of the block is highlighted in green.

### 1.7.2 One-Time Turners

Just like we can use gliders to create and move blocks around the Life plane, we can also use blocks to move gliders around the Life plane. We can thus use a p1 slow salvo to create arrangements of blocks that then change the direction of other gliders in the salvo, thus creating salvos that come from multiple directions.

To be a bit more explicit, consider the arrangement of blocks presented in Figure 1.23, which can be used to rotate a glider by 90 degrees, but is destroyed in the process (contrast this with reflectors like the Snark, which rotate gliders but remain unchanged). Reflectors like this are called *one-time turners*, and they are one of the key ingredients that let us emulate arbitrary glider syntheses via slow salvos. There are also numerous other 90-degree one-time turners involving various numbers of blocks (and sometimes other small still lifes—see Exercise 1.31), but this turner involving just two blocks is perhaps the simplest, and for now it is enough for our purposes.

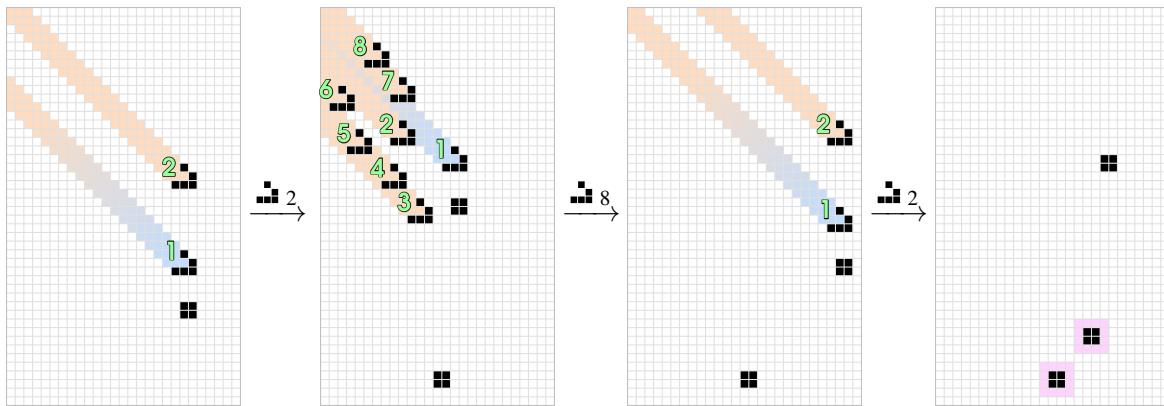


**Figure 1.23:** A one-time turner composed of two blocks that rotates a glider by 90 degrees and destroys the blocks at the same time.

Importantly, because this one-time turner is made up of nothing other than blocks, we can construct it using a p1 slow salvo via the techniques that we developed in the previous section. An explicit 12-glider p1 slow salvo that does the job (and also leaves us with an extra block that we can then use to construct additional turners) is displayed in Figure 1.24. This salvo works by using 2 gliders to duplicate the initial block (as in Figure 1.20), 8 gliders to move one of these blocks to a new position, and then 2 more gliders to duplicate this block again.

We can thus use a total of 13 gliders in a p1 slow salvo (12 to create the one-time turner and 1 to use and destroy it) to create 1 glider in a perpendicular direction, while still having a block left

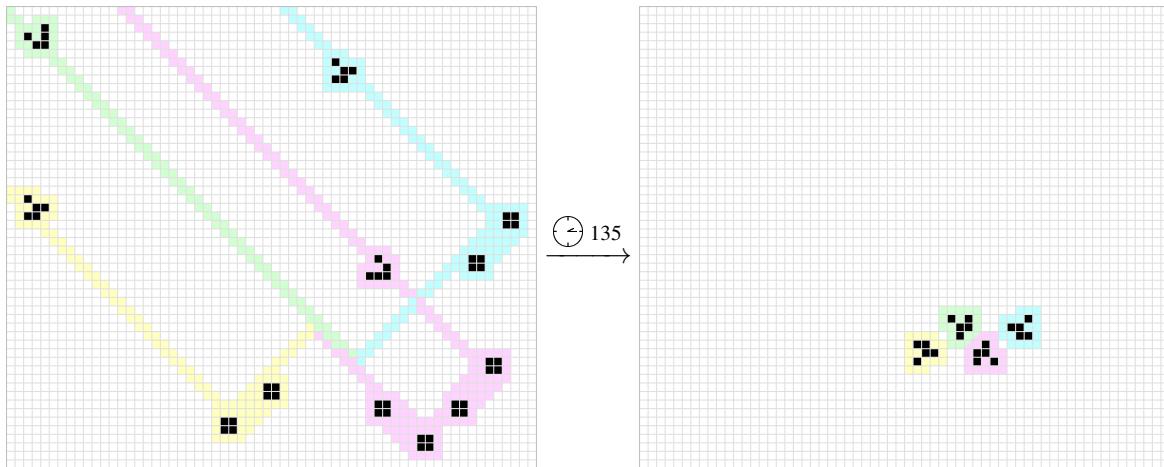
<sup>13</sup>A collection of them is available at [conwaylife.com/forums/viewtopic.php?f=7&t=1072#p8182](http://conwaylife.com/forums/viewtopic.php?f=7&t=1072#p8182).



**Figure 1.24:** A 12-glider p1 slow salvo creating the one-time turner from Figure 1.23 (highlighted in magenta) and leaving behind another block to create additional turners or other objects with. The first 2 gliders duplicate the initial block via the reaction of Figure 1.20, the next 8 gliders move one of the blocks down and to the right, and then the final 2 gliders duplicate that moved block.

over to perform additional constructions with. Using this technique, we are now able to emulate multi-directional glider syntheses via unidirectional syntheses.

For example, consider the four-glider syntheses of the clock that we saw in Table 1.3, which each use one glider coming from all four possible directions. A unidirectional synthesis of the clock can be constructed by using a p1 slow salvo to construct four one-time turners—one of the gliders does not need to be turned at all, two gliders need to be turned once each, and one glider needs to be turned twice—and then firing four gliders at the one-time turners with the proper timing. An explicit configuration of one-time turners that works is displayed in Figure 1.25.<sup>14</sup>



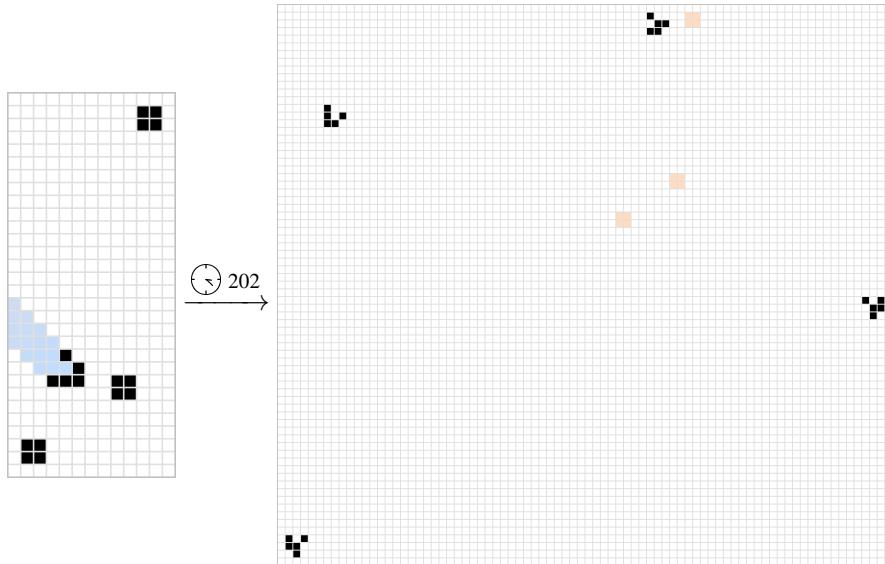
**Figure 1.25:** One-time turners (which can be synthesized with a p1 slow salvo) can be used to allow unidirectional glider waves (like the one on the left) to emulate multi-directional glider syntheses (like the one on the right). The gliders on the right are in exactly the position of the clock synthesis that we saw in Table 1.3.

Unfortunately, this construction does not give us a true p1 slow salvo for constructing a clock, since the final four gliders in the synthesis (i.e., the ones that hit the one-time turners) must be synchronized with each other—we are not free to space them arbitrarily far from one another. In order to fix this problem, we need to introduce some methods for using blocks to duplicate gliders and adjust their timing.

<sup>14</sup>It is possible to create a p1 slow salvo that builds this arrangement of blocks by hand, but there are also computer scripts that automate this process—see Exercise 1.38.

### 1.7.3 Splitters and Timing

The first ingredient that we need in order avoid having to send synchronized gliders at our one-time turners is a method of turning one glider into many gliders. Just like our one-time turner, we would like this pattern to be composed entirely of blocks, and we would like it to be cleanly destroyed during the glider-duplicating reaction. Patterns with these properties are called *blockic splitters*,<sup>15</sup> and one example is provided in Figure 1.26.



**Figure 1.26:** A *blockic splitter*, which uses three blocks to turn one glider into four gliders (while destroying the blocks in the process).

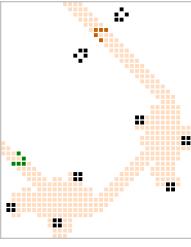
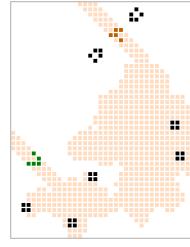
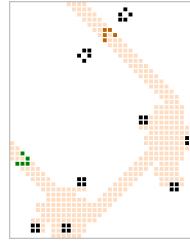
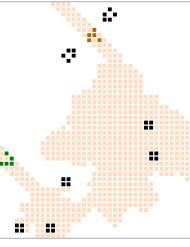
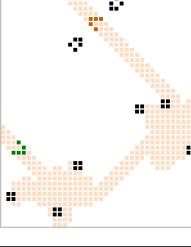
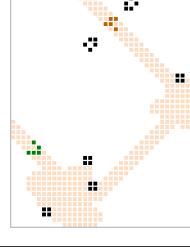
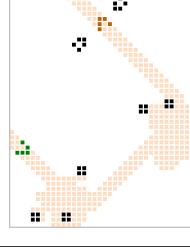
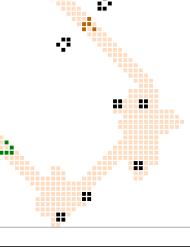
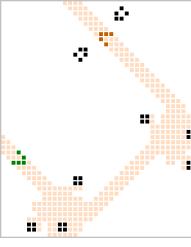
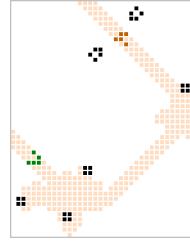
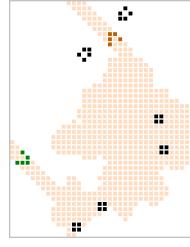
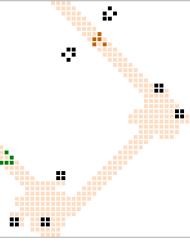
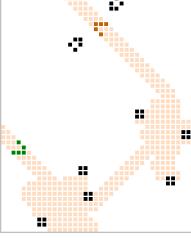
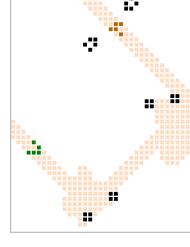
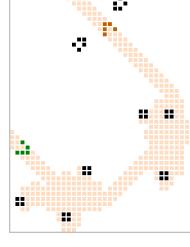
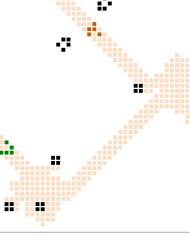
This splitter *almost* gets us what we need—it lets us turn one glider into four, and we could chain multiple splitters together to turn one glider into as many as we desire. We can use one-time turners to move these gliders around the Life plane, with the intention of emulating multi-direction glider syntheses, but unfortunately we are faced with two brand new problems:

- The one-time turner that we introduced in Figure 1.23 preserves the glider’s color, so we have no way of obtaining glider color combinations other than those provided to us by the splitter in Figure 1.26 (in which the bottom-left glider has the same color as the input glider, whereas the other 3 gliders have the opposite color). Since the 4-glider synthesis of a clock uses 2 gliders of each color, in order to make a slow salvo synthesis of the clock we would have to use this splitter multiple times and then use blocks to delete the excess gliders, which is quite wasteful.
- The one-time turner from Figure 1.23 gives us no control over timing, and we need some way of making sure that all of the desired gliders in a given synthesis not only arrive at the right place, but also at the right time.

Fortunately, there is a common solution to both of the above problems, and it is simply to introduce several new one-time turners. In particular, the numerous 180-degree one-time turners presented in Table 1.5 are capable of either preserving or changing the color of a glider, and also offsetting its timing by any amount that we desire.

In order to make use of these 180-degree one-time turners to put gliders in (almost) any *position* that we want, we first simply use copies of the 90-degree one-time turner from Figure 1.23 and the 180-degree one-time turners with offset 0 from Table 1.5. Importantly, we always place at least one 180-degree turner in the path of each glider, even if it is not required to get its positioning right (we will need that 180-degree turner to get the timing right momentarily).

<sup>15</sup>The term *blockic* refers to the fact that it is composed entirely of blocks. There are also splitters (and one-time turners) made up of other still lifes (or even blinkers), but we do not consider them here since blocks are enough for our purposes.

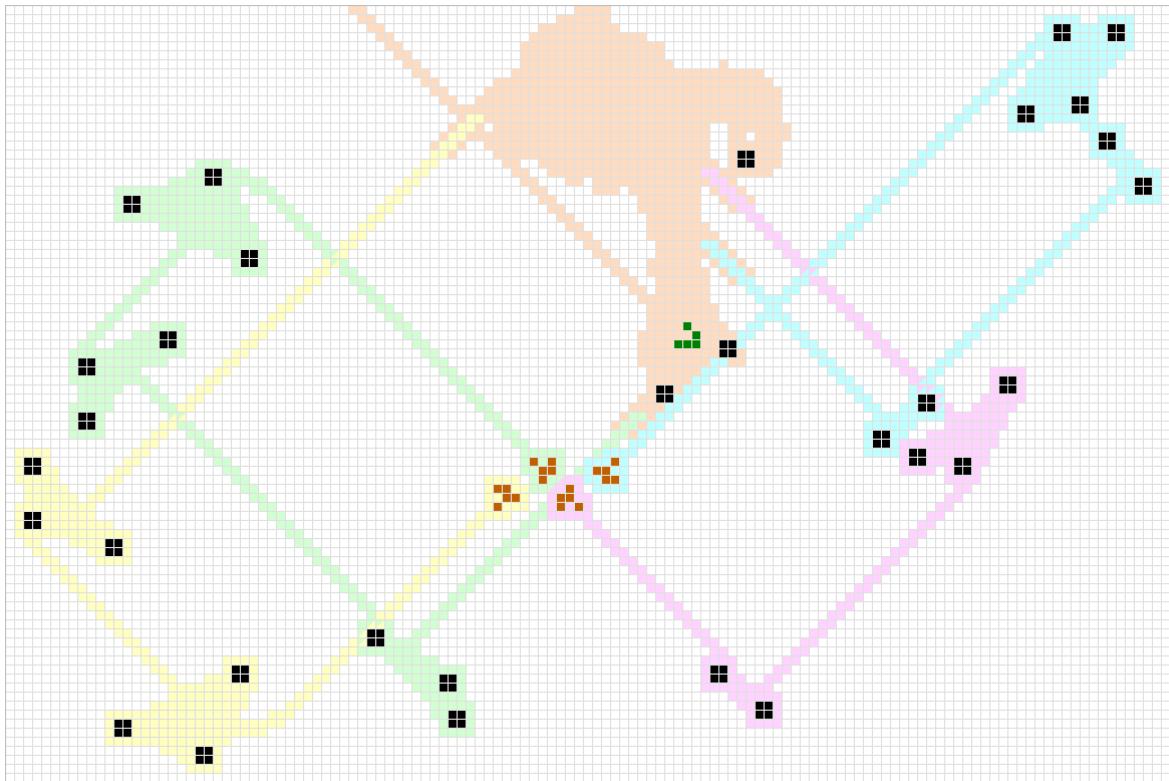
	Offset			
	0	1	2	3
Color-Preserving				
				
Color-Changing				
				

**Table 1.5:** By using different 90-degree one-time turners together, we can create this collection of 180-degree one-time turners that lets us set a glider to any timing and color. In all cases, the input glider is highlighted in green and comes in from the top-left, while the location of the output glider exactly 200 generations later is highlighted in orange. The boats are not required for the turners to function, but just serve to make it easier to line up multiple different turners.

Even though the positioning of the gliders is now correct, their timing will likely be horribly wrong. To fix this problem, we now focus on the glider that gets to its destination last—we will synchronize all of the other gliders with this one. To slow down those other gliders, we note that moving a 180-degree turner 1 cell farther away causes the glider to reach its destination 8 generations later (it now has to travel 1 cell, and thus 4 generations, farther in both directions). To delay a glider by  $n$  generations, we can thus move its 180-degree turner farther away by  $\lfloor n/8 \rfloor$  generations and then replace it by the turner from Table 1.5 that has the same effect on its color but has offset  $n \bmod 8$ .

We now have a complete set of glider-preserving or glider-changing turners capable of delaying a glider by any of  $0, 1, 2, \dots, 7$  generations. So we have all the tools we need to position *and* time gliders as we see fit. An example of how we can use the 4 gliders produced by the splitter in Figure 1.26

to synthesize a clock is presented in Figure 1.27. Note that we placed a 180-degree one-time turner in the path of each of the four gliders so that we could use the technique for correcting their timing described in the previous paragraph, and since each of those 180-degree turners is simply made up of two 90-degree one-time turners, we are able to separate their two halves in order to more easily control the glider's output position.



**Figure 1.27:** A *blockic seed* for a clock: The blockic splitter at the top-center (highlighted in orange) turns the single incoming glider into four gliders whose paths are outlined in yellow, green, aqua, and magenta. Those gliders are repeatedly reflected via one-time turners in such a way that they end up in the indicated positions required to synthesize a clock after 537 generations. We could probably get away with using fewer one-time turners, but by placing a 180-degree turner in the path of every glider it is much easier to get their timing right.

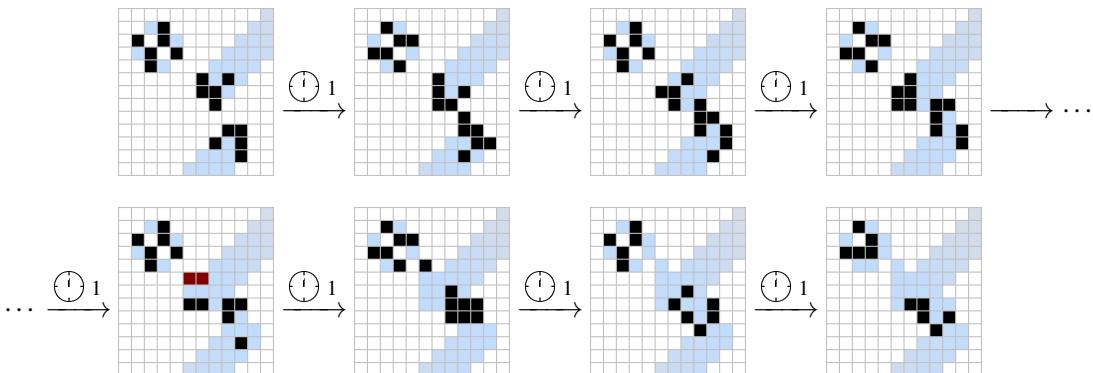
A configuration of simple still lifes, like the one in Figure 1.27, that synthesizes a particular object when hit by a glider (or gliders) is called a *seed* (and a seed made up entirely of blocks is called a *blockic seed*). Seeds are particularly useful since they immediately give us a p1 slow salvo for constructing an object—a clock in this case—since we “just” have to use a p1 slow salvo to construct the still lifes in the seed and then fire one additional glider to trigger the synthesis.

However, actually building a slow salvo synthesis for a seed is a tedious process. The 31-block seed for a clock from Figure 1.27 requires hundreds of slow gliders to construct (see Exercise 1.39)—recall that we already needed 12 gliders just to create a single 2-block one-time turner in Figure 1.24. For this reason, slow salvo syntheses of seeds are typically built by computer scripts, with *slsparse* (see [conwaylife.com/wiki/Slsparse](http://conwaylife.com/wiki/Slsparse) for tutorials and a download link) being the most widely used.

#### 1.7.4 Tight Packings of Gliders

There is still one final problem that might arise when trying to emulate an arbitrary glider synthesis via a slow salvo synthesis, and that is the fact that some syntheses involve packings of gliders coming from the same direction that are too close together for us to place with our one-time turners. This was not a problem for the clock synthesis in Figure 1.27 since each of the gliders in the synthesis comes from a different direction. However, if we were to try to use a p1 slow salvo to emulate the 3-glider

synthesis of a HWSS displayed in Table 1.2, it is not obvious that we can position the two gliders that are heading southwest close enough to each other (since one of the gliders might interfere with any one-time turner that we try to use to position the other glider). To fix this glider-packing problem, we introduce one final reaction: the *clock inserter*,<sup>16</sup> which uses two opposing gliders to transform a clock into a perpendicular glider, as in Figure 1.28.



**Figure 1.28:** The *clock inserter* is a reaction that very cleanly collides two gliders in such a way as to transform a clock into a perpendicular glider. The two-glider collision produces a domino spark in generation 5 (shown in red) and then dies off, while the domino spark initiates the clock-to-glider transformation. The debris at the bottom-right in generation 8 dies off in 5 more generations.

The key facts that make this reaction so useful for us are (1) that the output glider appears at a small offset and moves through the space just vacated by the colliding input gliders, and (2) the reaction is almost effortless—it does not disrupt any other cells around the clock and output glider, and thus does not disrupt other nearby gliders either. This reaction can thus be used to place a glider very close in front of, or to the side of, another already-placed glider (see Figure 1.29 for an illustration of how closely it can place a new glider to other gliders). Because of this, we can use this reaction to build any arrangement of gliders, no matter how tight, as long as we start by placing the gliders in the back first. Actually *proving* that the clock inserter reaction can be used to build any arrangement of gliders is somewhat technical and messy,<sup>17</sup> since there are many possible ways for gliders to be near each other, so we defer the proof to Appendix ??.

With this reaction in hand, we can now use a p2 slow salvo to construct any arrangement of gliders in the plane that we desire. First, we “rewind” the desired glider synthesis back as far as we like, so that it consists of four salvos of gliders—one coming from each direction.<sup>18</sup> We then construct the four salvos one glider at a time. If the gliders in a salvo are spaced sufficiently far apart, we can insert gliders into it via one-time turners. If the gliders are spaced close together, we instead first synthesize a clock (via p1 slow salvo synthesis, such as in Figure 1.27) and then use the clock inserter to insert the close gliders one at a time. We have thus finally proved the following theorem:

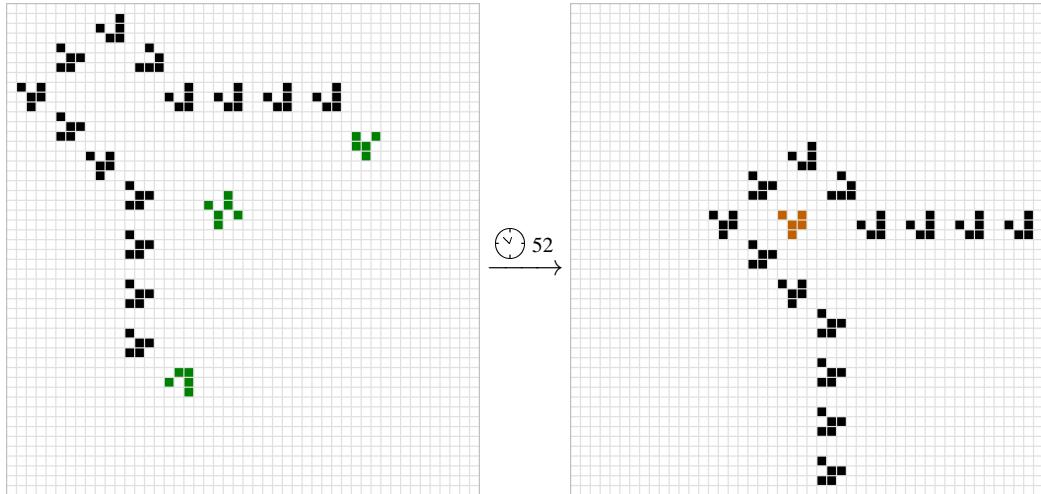
**Theorem 1.1** Every pattern that can be constructed via glider synthesis can be constructed by a p2 slow salvo glider synthesis.

The only p2 object that we needed to prove Theorem 1.1 was the clock, which we used to place clusters of tightly-packed gliders. We can get around having to use clocks (and thus avoid p2 objects altogether) by instead using the blockic seed of a clock from Figure 1.27. More specifically, if we wish to use a clock insertion reaction, we first split a single glider into three via blockic splitters—one

<sup>16</sup>Found by Martin Grant in December 2014.

<sup>17</sup>Chris Cain wrote a script that automatically uses clock inserters to build glider arrangements in 2014, and then he and Dave Greene used the script to build such a wide array of tight glider arrangements that this could probably be considered the first proof that p2 slow salvos can build any configuration of gliders. Cain’s script can be found at [conwaylife.com/forums/viewtopic.php?f=2&t=1512&start=25#p15133](http://conwaylife.com/forums/viewtopic.php?f=2&t=1512&start=25#p15133).

<sup>18</sup>Recall that the gliders must be able to reach their positions from arbitrarily far away in order to be considered a valid glider synthesis.



**Figure 1.29:** A demonstration of how the clock inserter (shown in green on the left) can be used to place a glider (shown in orange on the right) closely beside and in front of other already-placed gliders.

of those gliders then hits the seed, creating a clock, and the other two are reflected so as to trigger the clock insertion reaction as in Figure 1.28. Since this entire process is carried out by a single glider in the slow salvo synthesis, the result is a p1 slow salvo synthesis. We thus have the following strengthening of Theorem 1.1:<sup>19</sup>

**Theorem 1.2** Every pattern that can be constructed via glider synthesis can be constructed by a p1 slow salvo glider synthesis.

## 1.8 Notes and Historical Remarks

The importance of glider synthesis was known essentially as soon as the glider itself was found in 1970, with common folklore being that we could send gliders as signals throughout the Life plane and collide those gliders in different ways to simulate arbitrary computations. This basic idea has been refined and made more precise repeatedly over the past 50 years, to the point that there are now explicit patterns that do exactly this—they collide gliders so as to perform arbitrary computations and build almost any pattern of our choosing (we will delve deeply into the specifics of how these patterns work in Chapters ?? and ??).

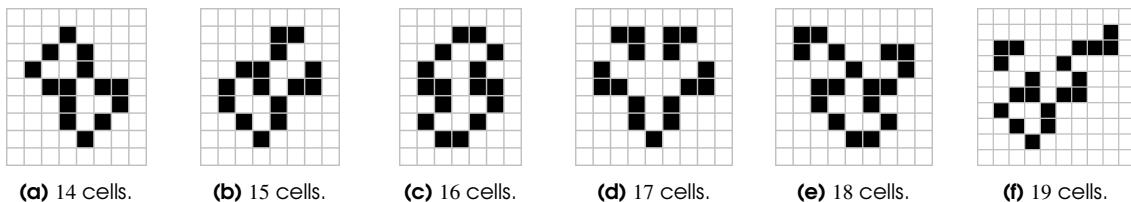
The first specific and explicit uses of glider syntheses were demonstrated in 1971, when Bill Gosper constructed the first breeder (essentially the breeder that we built in Section 1.6) as well as the first lightweight and middleweight spaceship guns. These patterns demonstrated the kind of leap in complexity that is possible when taking advantage of glider synthesis, and it led to a surge in interest in the topic over the following years.

By 1973, the majority of “basic” Life objects were synthesizable, including lightweight, middleweight, and heavyweight spaceships, switch engines (and their block-laying and glider-producing counterparts), commonly-occurring oscillators like the pentadecathlon and pulsar, and all still lifes and oscillators with 7 or fewer cells other than the clock and the long snake. Syntheses of larger composite patterns were even being discovered by this point, with Douglas Petrie constructing the 11-glider synthesis of the Schick engine displayed in Figure 1.10 in 1973. The majority of these early syntheses were developed by David Buckingham, Mark Niemiec, and Douglas Petrie.

Over the following decades, David Buckingham continued to develop syntheses for still lifes and oscillators, and he completed syntheses of all of them with 14 or fewer cells by no later than 1992. His technique was to make heavy use of incremental synthesis, building the objects from the inside

<sup>19</sup>There are some mild technicalities that we have glossed over here—see Exercise 1.42.

out, only using a couple of gliders at a time to tweak the outermost portion of the object that he was synthesizing. This method works very well when synthesizing objects that have “end pieces” like tails that can be removed or altered without affecting the pattern’s stability. However, compact still lifes like the one in Figure 1.30(a) are much more difficult to construct, and this particular still life (which was the last 14-cell still life to be synthesized) was initially synthesized using a massive incremental synthesis involving more than 30 gliders.



**Figure 1.30:** The final still lifes with 14–19 cells to be synthesized by gliders.

Mark Niemiec continued on with this work, creating a large database that he used to automatically generate syntheses of thousands of still lifes by piecing together known reactions [Nie03, Nie10]. This greatly reduced the number of still life syntheses that needed to be found by hand, and in 2013 he completed syntheses of all 15-cell still lifes (as well as syntheses for all except for a few hundred 16-, 17-, and 18-cell still lifes).

Glider syntheses for the remaining 16-cell still lifes were then found via a collaborative effort on the ConwayLife.com forums, mostly led by Martin Grant, Matthias Merzenich, and Mark Niemiec, with the final synthesis (see Figure 1.30(c)) being completed in January 2014. Another five-month collaborative effort, led by the same group of people, completed syntheses of the 17-cell still lifes in May 2014. A comparatively quick two-month effort finished the remaining syntheses of 18-cell still lifes in November of that year, and then a final four-month effort resulted in the synthesis of the last 930 or so 19-cell still lifes in February 2020.

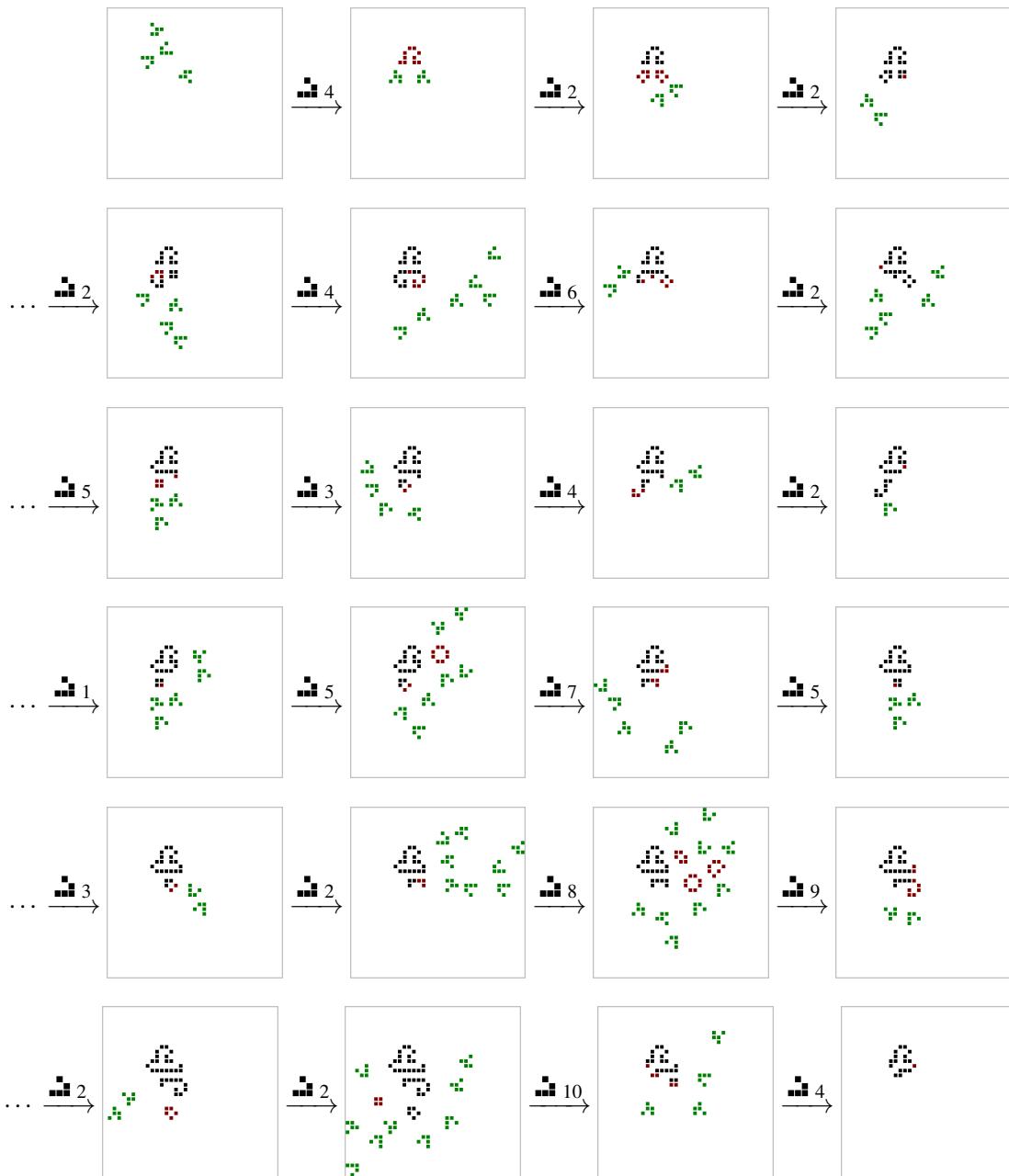
To give an idea of the size of this achievement, recall from Table ?? that there are 78,297 different strict still lifes with 19 or fewer cells. Not only is this a huge number of distinct objects to synthesize, but the syntheses themselves are often monstrously large. For example, Figure 1.31 shows how one of the “problematic” 17-cell still lifes was constructed by using a whopping total of 94 gliders and over 20 stages of incremental synthesis.

Rather than trying to push these techniques to synthesize all still lifes with 20 live cells, focus has shifted somewhat to trying to reduce the number of gliders required to synthesize these objects. It has been known for decades how to synthesize all still lifes with 8 or fewer cells via 4 or fewer gliders. Similarly, glider syntheses are known for constructing every still life with 9, 10, 11, 12, and 13 live cells via 5, 9, 9, 11, and 12 or fewer gliders, respectively.<sup>20</sup> Recent efforts have attempted to continue this pattern and synthesize all small still lifes in fewer than 1 glider per live cell. This project was completed for 14-, 15-, 16-, and 17-cell still lifes in October 2016, November 2016, May 2017, and September 2019, respectively. For example, we now know how to construct the 17-cell still life from Figure 1.31 (which was first synthesized by 94 gliders) via just 10 gliders—see Figure 1.32.

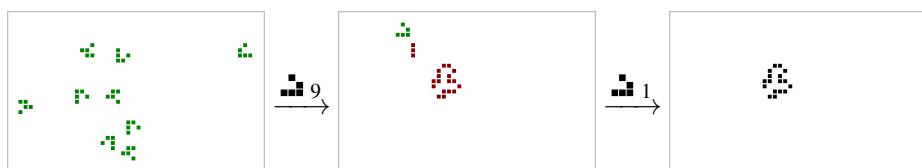
Remarkably, it was even shown in September 2020 that every glider-synthesizable still life can be synthesized via 17 or fewer gliders—see Section ???. However, it remains an open question whether or not every still life (or every oscillator, or even every pattern that has predecessors) is actually glider-synthesizable. Closely related is the (also open) question of whether or not there exists a still life whose only parent is itself.<sup>21</sup> Finding such a pattern would answer both of these questions, since if a still life’s only parent is itself then it can not be constructed via glider synthesis (or any other means).

<sup>20</sup>It was not known how to synthesize the 9-cell long<sup>3</sup> snake with fewer than 6 gliders until February 2015, when Matthias Merzenich derived a 5-glider synthesis from an apgsearch soup.

<sup>21</sup>For the purposes of this problem, a still life together with far away non-interacting dying ash is not considered a different parent than just the still life itself. Conway offered a \$50 prize for a solution to this problem in October 1972, which has gone unclaimed to this day.



**Figure 1.31:** A summary of a massive 94-glider incremental synthesis of a hard-to-construct 17-cell still life (shown at the bottom right). Each stage in the synthesis works by using a glider collision to transform one still life into another. The cells that were created or modified in the previous step of the synthesis are shown in red, and the gliders that will be involved in the next collision are shown in green.

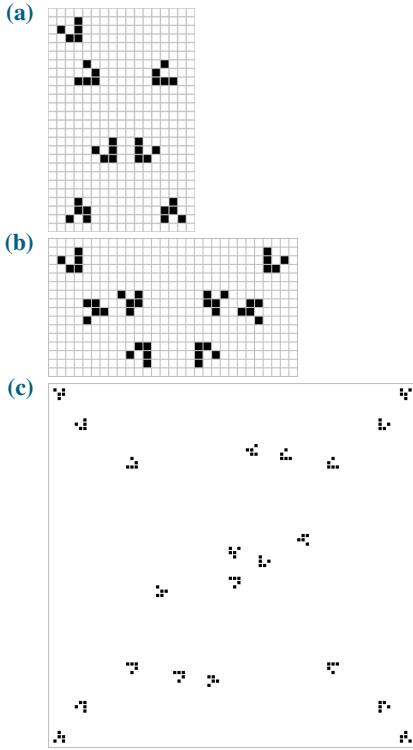


**Figure 1.32:** A 10-glider synthesis of the 17-cell still life from Figure 1.31 that was constructed with the help of a soup that was found by apgsearch.

## Exercises

solutions to starred exercises on page ??

\***1.1** Many glider syntheses work by using a small number of gliders to create the desired object plus some debris, and then additional gliders to clean up the debris. In each of the following syntheses, identify which gliders are used to clean up debris.



**1.2** Gliders can be used to destroy essentially any unwanted debris that is left over after a synthesis. Use a single glider to destroy each of the following objects (and also destroy the glider in the process):

- (a) A block.
- (b) A beehive.
- (c) A blinker.
- (d) A ship.
- (e) An LWSS.

**1.3** Use multiple gliders to completely destroy each of the following objects.

[Hint: Use one or two gliders to break the object down into simple ash objects like those from Exercise 1.2 and then use additional gliders to clean up those simpler objects.]

- (a) A queen bee shuttle.
- (b) A copperhead.
- (c) Rich's p16.

\***1.4** There are exactly 6 distinct ways for a glider to collide with a block. List them all and describe the result of each collision.

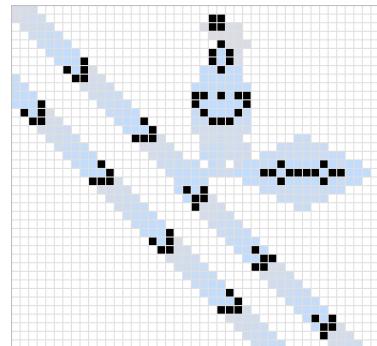
\***1.5** Use the syntheses from this chapter to create a 7-glider synthesis of eater 5.

**1.6** Use a lightweight spaceship synthesis from this chapter to construct a backrake that creates a period 120 stream of lightweight spaceships.

**1.7** Use three Gosper glider guns to create a middleweight spaceship gun.

**1.8** Consider the heavyweight spaceship synthesis in Table 1.2.

- (a) Why can't you use three Gosper glider guns and this synthesis to create a heavyweight spaceship gun?
- (b) One way to overcome this problem is to use *glider pushers*<sup>22</sup> (displayed below) to repeatedly push one glider stream closer to another. How many glider pushers would you need to use to fix the problem from part (a)?



**1.9** Use a two-glider synthesis of a block and a three-glider synthesis of a queen bee to create a 7-glider synthesis of a queen bee shuttle.

**1.10** Consider the four-glider synthesis of the twin bees presented in Figure 1.3.

- (a) Use this synthesis to synthesize a twin bees shuttle.
- (b) Use this synthesis to synthesize the twin bees gun from Figure ??.

**1.11** Create a glider synthesis of the glider pusher from Exercise 1.8.

**1.12** Construct a glider synthesis of the period 36 oscillator from Figure ??.

\***1.13** By using the 2-glider *kickback reaction* from Table 1.1 that changes the direction of a glider, we can decrease the number of directions used in many glider syntheses at the expense of increasing the number of gliders required. Use this technique to create glider syntheses for each of the following patterns with the property that all gliders come from just two different directions.

- (a) A switch engine.
- (b) A clock.
- (c) A 3-engine Cordership.

<sup>22</sup>Found by Dietrich Leithner in December 1993.

**\*1.14** Recall the 2-engine Cordership from Exercise ??.

- (a) Construct a glider synthesis of this Cordership.
- (b) Construct a glider synthesis of this Cordership that uses gliders coming only from two different directions.

[Hint: Refer back to Exercise 1.13.]

**\*1.15** By using the 3-glider *tee* collision from Table 1.2, which produces a glider perpendicular to each of the input gliders, we can modify many glider syntheses so that all input gliders come from two directions that are opposite each other. Use this technique to create glider syntheses of each of the following patterns, with the property that all gliders come from two opposing directions.

- (a) A switch engine.
- (b) A clock.
- (c) Twin bees.

**1.16** Construct a glider synthesis for a Gosper glider gun that is stabilized on one end by an eater 1 instead of a block (as in the buckaroo of Figure ??).

**1.17** For incremental syntheses of moving objects, like this ecologist from Figure 1.8, there tend to be more choices to be made, since individual components may be constructed earlier or later in time, and the location where they must be constructed will change accordingly.

- (a) Complete this glider synthesis (i.e., construct the ecologist its entirety from an arrangement of gliders). This synthesis should only use gliders coming from the bottom-left and bottom-right.

[Hint: You can use two gliders to create a block and then two more gliders to turn the block into a LWSS.]

- (b) Find the four gliders that synthesize the first LWSS in this ecologist. To construct that LWSS four generations sooner without disrupting the final recipe, how far and in what direction should each of these gliders be moved?

[Hint: Divide the gliders into two pairs based on their direction of travel. Each pair will move a different distance in a different direction.]

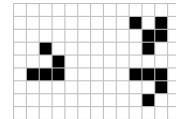
**\*1.18** In Figure 1.9 we showed how to use gliders to turn an ecologist into a forward space rake.

- (a) Complete this incremental glider synthesis of the space rake (i.e., construct it in its entirety from an arrangement of gliders).
- (b) Show how to synthesize a period 60 space rake. [Hint: Recall the synthesis of the Schick engine from Figure 1.10.]

**\*1.19** In Figure 1.9 we showed how to use gliders to turn an ecologist into a forward space rake. Use similar techniques to turn an ecologist into a backward space rake.

**1.20** Create an arrangement of guns that synthesizes the fast forward force field from Figure ???. Use another gun to fire lightweight spaceships that are teleported through this fast forward force field.

**1.21** Use the following 3-glider collision to reduce the 18-glider synthesis of Rich's p16 in Figure 1.16 down to 16 gliders:<sup>23</sup>



**1.22** Create a breeder that uses three rakes to synthesize glider-producing switch engines via the 3-glider collision displayed in Table 1.2.

[Hint: You will need to use rakes with very high period.]

**1.23** The breeder from Figure 1.18 can be made quite a bit smaller by moving its space rakes closer together (and rephasing them as necessary). Use this technique to reduce the breeder's width by at least 50 cells.

**1.24** Modify the breeder from Figure 1.18 so that the space rakes still move to the east, but the Gosper glider guns shoot gliders to the northwest instead of the northeast.

**1.25** Construct a breeder that creates Gosper glider guns that are stabilized on at least one side by eater 1s instead of blocks. [Hint: Use the glider synthesis from Exercise 1.16.]

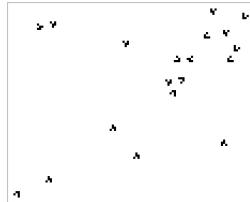
**1.26** Create a breeder that uses several rakes to synthesize block-laying switch engines as they move. [Hint: Use a 3-glider synthesis of a switch engine together with the reaction from Figure ??.]

**1.27** Create a breeder that uses several rakes to synthesize twin bees guns as they move, using the glider synthesis that you constructed in Exercise 1.10.

**1.28** Use a p1 slow salvo to create the arrangement of 8 blocks in Figure 1.25.

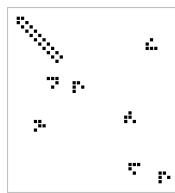
<sup>23</sup>This 16-glider synthesis was found by Martin Grant.

\*1.29 A 19-glider synthesis of a boatstretcher (refer back to Figure ?? and Exercise ??) is displayed below.



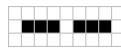
- (a) Remove some gliders so as to produce a glider synthesis of the crab.
- (b) Add some extra gliders to destroy the boatstretcher (but not the boat itself) after it has been synthesized. Use this method to show that a fixed number of gliders (say 30 or so) can be used to synthesize arbitrarily-large strict still lifes.

1.30 The 7-glider reaction below attaches a period-2 oscillator called a *barberpole* to the end of a stretched tub or boat. Show how this reaction can be combined with the method of Exercise 1.29 to synthesize arbitrarily-large oscillators with a fixed number of gliders (say 40 or so).



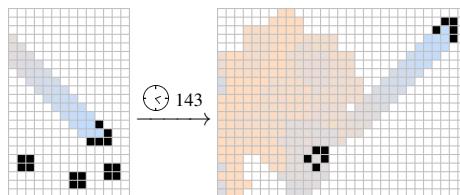
\*1.31 In this exercise, you will construct some limited-use turners that are different from the blockic one-time turners that we saw in this chapter.

- (a) Show that a single boat can be used as a 90-degree one-time turner.
- (b) Show that a single eater 1 can be used as a 90-degree one-time turner.
- (c) Show that a single long boat can be used as either a 90-degree or 180-degree one-time turner.
- (d) Show that the following arrangement of two blinkers can be used as a 90-degree one-time turner.



\*1.32 In this exercise, we practice moving gliders around one-time tracks.

- (a) Use four boats to move a glider around a square track once, and then leave the track in the same direction that it started in (destroying the track in the process).
- (b) The following pattern might be called a “two-time turner”, since it can be used to turn two gliders (by firing the second glider at the boat that the first glider produces). Explain why it is *not* possible to use four copies of this pattern to move a single glider around a square track twice (destroying the track in the process).



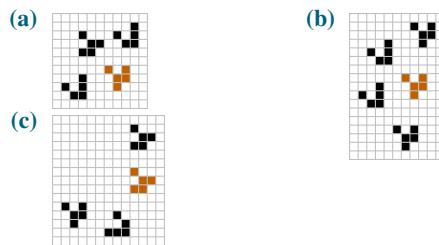
- (c) Use three copies of the two-time turner, together with some additional one-time-turners, to move a glider around a square track twice, and then leave the track in the same direction that it started in (destroying the track in the process).

\*1.33 One-time turners can be used as tracks for gliders, allowing us to create objects that move at a different speed at the front than at the back (such patterns are called *growing spaceships*).

- (a) Use two copies of the blinker puffer from Figure ?? to lay two blinker fuses that a glider bounces back and forth between, using the reaction from Exercise 1.31(d).
- (b) Use rakes of your choosing to synthesize a track of boats that a glider uses as one-time turners and destroys (as in Exercise 1.31(a)).

\*1.34 Find a way of placing a block near a clock so that a single glider (rather than a pair of gliders, as in Figure 1.28) can trigger the clock inserter reaction.

\*1.35 Use a clock inserter to insert the orange glider into each of these glider salvos, similar to how we inserted the orange glider into the salvo in Figure 1.29. In all cases, the input gliders to the clock inserter must come only from the lower-left and the upper-right.



\*1.36 Many reactions can be used as inserters other than the clock inserter from Figure 1.28 (however, none are quite as good as the clock inserter itself).

- (a) Use the “tee” 3-glider collision from Table 1.2 to insert a glider 15 generations in front of another glider.
- (b) Use an eater 1 to insert a glider 15 generations in front of another glider.  
[Hint: Refer to Exercise 1.31(b).]
- (c) Use the clock inserter to insert a glider 14 generations in front of another glider.

1.37 Create blockic seeds for each of the following objects. [Hint: These can all be created using the same techniques that we used to make the blockic seed in Figure 1.27.]

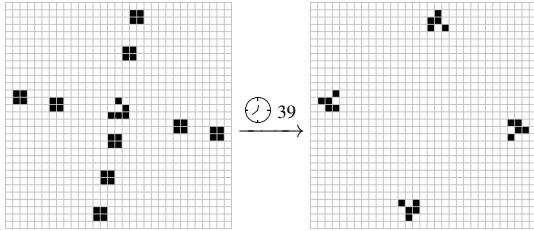
- (a) A lightweight spaceship.
- (b) A pulsar.
- (c) A switch engine.

1.38 Create a slow salvo that turns a single block into the configuration of 8 blocks displayed in Figure 1.25.

[Hint: You can do this by hand via 60 or so gliders by repeating the method of Figure 1.24. However, there is a search program called *s1sparse* that builds slow salvos like this automatically. Tutorials and a download link can be found at [conwaylife.com/wiki/S1sparse](http://conwaylife.com/wiki/S1sparse).]

**1.39** Use slsparse ([conwaylife.com/wiki/Slsparse](http://conwaylife.com/wiki/Slsparse)) to create a slow salvo that constructs the blockic seed for a clock from Figure 1.27, and thus a slow salvo for the clock itself.

**1.40** A commonly-used blockic splitter that turns one glider into four gliders is displayed below. This splitter has the advantage of being much faster and cleaner than the one in Figure 1.26, but the disadvantage of requiring 9 blocks instead of just 3. Use this splitter as part of a blockic seed for a clock.



**1.41** By chaining together multiple blockic splitters, we can turn one glider into any number of gliders. Create a blockic seed that turns one glider into exactly...

- (a) 7 gliders.
- (b) 10 gliders.
- (c) 9 gliders.

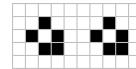
**\*1.42** The argument that we used to improve Theorem 1.1 to Theorem 1.2 relied on us being able to use blockic splitters to create three gliders: one to trigger a clock seed and two to trigger the clock insertion reaction. How can we ensure that the clock is in the correct phase when the latter two gliders arrive to trigger the clock insertion?

**\*1.43** In the proof of universality of the clock inserter in Appendix ??, we defined a glider's “rank” as its lane number plus its timing. Let  $w$  be a real number and suppose that we instead defined a glider's rank as its lane number plus  $w$  times its timing.

- (a) What parts of this proof change if we use  $w = 2$ ? Does the proof still work?
- (b) What is the slope of the lines of constant rank in the Life plane when  $w = 2$  (recall that the lines of constant rank have slope 3 when  $w = 1$ )?
- (c) The largest value of  $w$  for which the proof still works is  $w = 7/3$ . However, there is one extra technicality in this case—what is it, and how can it be overcome?
- (d) What is the smallest value of  $w$  for which the proof still works?  
[Hint: There will be a technicality similar to the one from part (c) that has to be overcome if  $w$  is minimal.]
- (e) What are the possible slopes of lines of constant rank in the Life plane as  $w$  ranges from its minimal to maximal values? These are the slopes that we can use to define the “front” of a glider salvo for clock-insertion purposes.

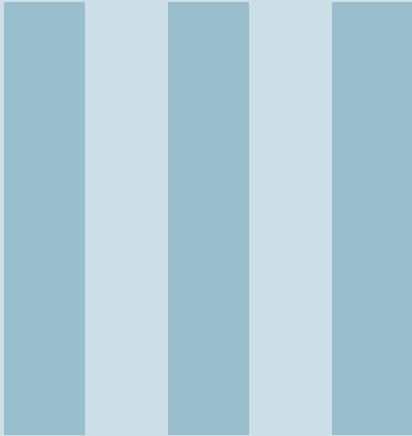
**1.44** In this exercise, we demonstrate how to construct seeds for Corderships.

- (a) Show how a single glider can be fired at the following pair of boats so as to synthesize (generation 2 of) a switch engine.



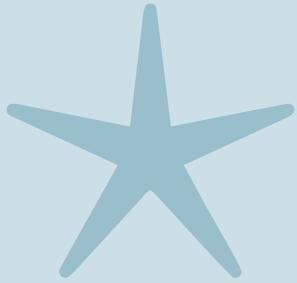
- (b) Create a glider synthesis that first creates the pair of boats displayed in part (a) and then creates a switch engine from them.
- (c) Use your solution to part (a) to construct a seed that, when hit by 2 gliders, synthesizes a 2-engine Cordership (refer back to Exercise ??).
- (d) Create a glider synthesis that first creates your seed from part (c) and then creates a 2-engine Cordership from it.





# Constructions





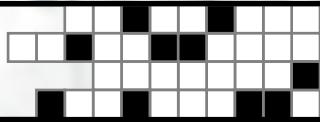
# Appendices and Supplements

Bibliography .....	39
Index .....	41





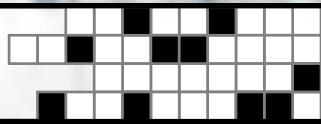
## Bibliography



- [Nie03] Mark D. Niemiec. Synthesis of complex Life objects from gliders. In *New Constructions in Cellular Automata*, pages 55–78. Oxford University Press, London, 2003.
- [Nie10] Mark D. Niemiec. Object synthesis in Conway’s Game of Life and other cellular automata. In *Game of Life Cellular Automata*, chapter 8, pages 115–134. Springer London, 2010.



# Index



## Symbols

- (11,0) block pull ..... 21
- (11,0) block push ..... 21
- (2,1) block pull ..... 20
- (2,1) block push ..... 21

## A

- apgsearch ..... 28

## B

- B-heptomino ..... 7
- banana spark ..... 8
- barperpole ..... 32
- beehive ..... 7
- bi-block ..... 7
- billiard table ..... 11
- blinker ..... 7
- block ..... 7, 19
- blockic seed ..... 25, 32
- blockic splitter ..... 23
- boat ..... 7, 32
- boatstretcher ..... 32
- breeder ..... 17

## C

- clock ..... 10, 19, 22, 26, 30
- inserter ..... 26

- Cordership ..... 14, 30
- crab ..... 32

## D

- domino spark ..... 26

## E

- eater
  - 1 ..... 7, 32
  - 2 ..... 10
  - 5 ..... 30
- ecologist ..... 13

## F

- fast forward force field ..... 31
- fumarole ..... 11

## G

- glider ..... 7, 9
- color ..... 23
- gun ..... 5
- pusher ..... 30
- synthesis ..... 5
- glider-producing switch engine ..... 9
- growing spaceship ..... 32

**H**

- hat ..... 12  
 heavyweight  
     spaceship ..... 9, 10, 12  
 heptomino  
     pi ..... *see* pi-heptomino  
 honey  
     farm ..... 7

**I**

- interchange ..... 7

**K**

- kickback reaction ..... 7, 30

**L**

- lightweight  
     spaceship ..... 5, 9, 12  
 loaf ..... 7  
 long boat ..... 32  
 long<sup>3</sup> snake ..... 28  
 lumps of muck ..... 7

**M**

- middleweight  
     spaceship ..... 9, 12

**O**

- one-time turner ..... 21

**P**

- pentadecathlon ..... 9, 12  
 pi-heptomino ..... 7  
 pond ..... 7

- pulsar ..... 9

**Q**

- queen bee ..... 9, 12  
     shuttle ..... 30

**R**

- R-pentomino ..... 9  
 rake ..... 5  
 Rich's p16 ..... 16, 31

**S**

- salvo ..... 19  
 seed ..... 19, 25, 33  
 ship ..... 12  
 slmake ..... *see* slsparse  
 slsparse ..... 25, 32  
 snake ..... 12  
 Snark ..... 21  
 soup ..... 16  
 space rake ..... 13  
 strict still life ..... 32  
 switch engine ..... 5, 9, 14, 30

**T**

- T-tetromino ..... 12, 13  
 teardrop ..... 7  
 tee ..... 8, 31, 32  
 traffic light ..... 7, 12  
 tub ..... 12  
     with tail ..... 12  
 twin bees ..... 10, 30  
     gun ..... 8  
 two-glider mess ..... 7  
 two-time turner ..... 32