



ЛАРЕЦ, В КОТОРОМ ПРЯЧЕТСЯ ЖИЗНЬ, МОРСКАЯ РАКОВИНА И ДУША

Книгу с таким названием написал профессиональный программист Руди Рукер (Rudy Rucker), ставший писателем-фантастом после многих лет занятий программированием. Книгу можно отнести к разряду «философия информатики». Информатика – сравнительно новая наука, поэтому узнать, как она отражается на мировоззрении людей, чрезвычайно интересно. Несколько глав книги выложены на сайте <http://www.rudyrucker.com/lifebox/>

В первой главе книги Руди Рукер объясняет смысл использованных в названии слов, сочетание которых отражает замысел книги:

«*Lifebox* – слово, которое я изобрел несколько лет назад, чтобы описать гипотетический технологический прибор для сохранения человеческой индивидуальности. В моих научно-фантастических рассказах, *lifebox* – маленькое диалоговое устройство, которому Вы сообщаете Вашу историю жизни. Оно задает Вам вопросы и организует информацию, которую Вы сообщаете. Наряду со словами, Вы можете сообщить информацию в цифровых образах, видео, звуках и т. п. Этот прибор можно уподобить интеллектуальному отпечатку индивидуальности (от ред.: при переводе *lifebox* в заглавии статьи выбрано словосочетание «ла-

рец, в котором прячется жизнь», образ, известный нам из народных сказок).

Как только Вы обеспечиваете Ваш *lifebox* достаточной информацией, он становится чем-то вроде Вашей модели. Ваша аудитория может взаимодействовать с историями в *lifebox*, прерывая и задавая вопросы. *Lifebox* начинался как техническая реализация мечты пожившего человека о том, чтобы оставить о себе память, а вылился в идею моделирования ее владельца.

Что дает человеку *lifebox*?

Бессмертие, вездесущность, всемогущество.

Вы могли бы оставить *lifebox* для Ваших внуков, и прапраправнуки смогут узнать то, какими Вы были. Вы могли бы использовать Ваш *lifebox* как способ предста-

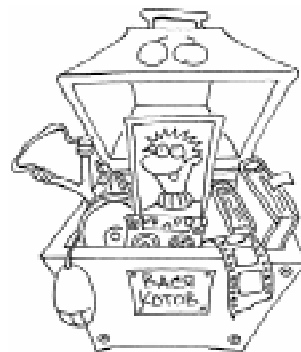
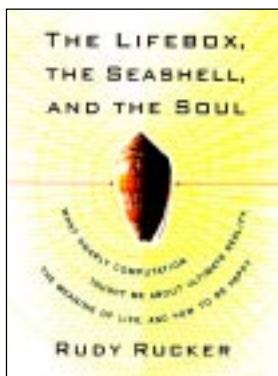


Рисунок 1. Обложка книги и ее автор.

Lifebox ... вылился в идею моделирования ее владельца.



Рисунок 2. Морские раковины, сфотографированные ночью.

виться большим количеством людей. Вы могли бы позволить Вашему *lifebox* выполнять некоторые из наименее интересных обязанностей, типа ответов на обычные обращения по телефону и электронной почте.

Lifebox – человек, уменьшенный до цифровой базы данных с простым программным обеспечением доступа.

В названии моей книги, я использую *Lifebox* как сокращение для универсального тезиса автоматизации, состоящего в том, что все, даже человеческое сознание, является вычислением».

Почему в название входит душа, ответим словами самого автора:

Антитеза (к *lifebox* – прим. ред.) состоит в том, что на самом деле никто не думает, что какая-нибудь сложно себя ведущая телефонная станция в действительности является живой. Все мы чувствуем, что имеем что-то такое, что не описывается никакой механической моделью, – это то, что мы обычно называем *душой*».

Наконец, какое отношение к сказанному имеет морская раковина?

Автор видит в раковине образ того, как можно вдохнуть жизнь в неодухотворенный *lifebox*.

Автор вдохновляется идеями Стефана Вольфрама (Stephen Wolfram известен своей программой *Mathematica*) об удивительных свойствах клеточных автоматов, действующих по очень простым правилам, но порождающих неповторяющиеся картинку, овеянные смыслом. Такие рисунки мы видим и в природе, рассматривая, например, морские раковины, которые демонстрируют бесконечное разнообразие неповторяемых узоров.

Автор прилагает к своей книге программу для экспериментов с клеточными автоматами (рисунок 3). Программа допускает

некоммерческое распространение и находится на диске к журналу.

«Итак, – пишет автор, – моя основная триада и моя диалектическая тактика подразумевают многократное прохождение следующих трех шагов:

(А) *Гипотетический шаг*: моделировать явления реального мира как вычисления.

(В) *Шаг-антитеза*: учитывать, что фактический мир оказывается более сочным и более интересным, чем вычисление.

(С) *Синтетический шаг*: заметить, что, имея достаточно времени и памяти, наше вычисление способно породить чрезвычайно богатые структуры, подобные естественным».

Приведенные цитаты были взяты из раздела «*Все есть вычисление*».

А теперь приведем несколько любопытных классификаций, взятых из следующего раздела первой главы книги, которая называется «*Новый вид науки*».

Он начинается с определения реальной выполнимости программ, которая не обсуждается в работах по теории алгоритмов, а является скорее предметом осмысления роли компьютера в нашей жизни.

«Факт, противоречащий интуиции.

Хотя теоретически некоторое вычисление сделать можно, но практически оно

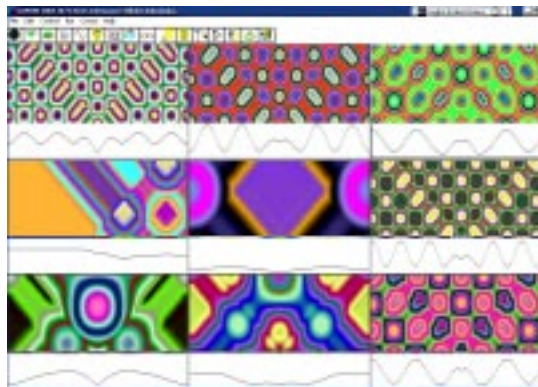


Рисунок 3.

может оказаться невыполнимым. Иногда нас вводит в заблуждение доказательство, что одна система в принципе может моделировать другую, и мы упускаем из виду факт, что это моделирование может оказаться столь медленным и ресурсоемким, что реально является невыполнимым.

Неформальное определение. Вычислительный процесс *выполним*, если он достигает желаемого результата за разумное по человеческим понятиям время.

Вычисление, которое можно сделать за несколько минут «руками», выполнимо, а вычисление, которое потребует десятки лет, – невыполнимо. Другими словами, вычисление *невыполнимо*, если требуется нереальное количество вычислительных ресурсов и/или времени.

Большинство программ искусственного интеллекта (ИИ) попадают в такую категорию *vis-a-vis* человеческого мышления – да, они могут моделировать небольшие части человеческого мышления, но модель работает настолько медленно, что применение таких программ к реальным большим объемам данных является *невыполнимым* процессом. (На самом деле, ситуация еще хуже: не только существующие программы ИИ являются невыполнимыми, но мы, вероятно, не нашли правильного подхода к созданию программ ИИ вообще). Выполнимость вычисления зависит как от вычислительной системы, которую Вы планируете использовать, так и вычислительного метода, который Вы собираетесь применить. Это относится к различию между *аппаратными*



...моделировать явления реального мира как вычисления.

средствами (hardware) и программным обеспечением (software). Если Вы имеете слабые вычислительные средства, то почти никакие вычисления невыполнимы. Но, независимо от того, какими аппаратными средствами Вы располагаете, улучшенное программное обеспечение (типа различных программистских трюков) может расширить область выполнимости.

Положим, мы соглашаемся с принципом всеобщего автоматизма, а именно, с тем, что большинство физических процессов – это вычисления. Вообще говоря, эти физические вычисления недоступны для наших персональных компьютеров. Не только моделирование погоды в мире невыполнимо в форме цифровых вычислений, но даже моделирование завихрений в потоке воды из-под крана недоступно существующим электронным машинам.

Но – и это моя точка зрения – эти процессы могут быть вычислены каким-то способом. Для человека наиболее восхитительные и наиболее знакомые вычисления из всех – это творческие и содержательные процессы человеческой мысли. По причинам, которые будут обсуждаться в одной из следующих глав, я бы не хотел узнать, что человеческое творчество станет вычислимым электронным процессом в следующие сто лет. Но опять же, это не исключает рассмотрение человеческого мозга как некоторого типа компьютера.

Мозг – система, подчиняющаяся конечному набору правил. Человеческая мысль – это выполнимое вычисление *для человеческого мозга*, но в данное время невыполнимое для *электронных компьютеров*.

Второй факт, противоречащий интуиции, который я хочу упомянуть, состоит в том, что вычисления могут породить неожиданные результаты. Можно было бы предположить, что основанный на жестких правилах процесс должен протекать весьма рутинным путем. Да, но это не означает, что долгосрочное поведение вычисления предсказуемо.

Неформальное определение.

Р предсказуемо, если имеется сокращенное вычисление Q , которое дает те же ре-

зультаты, что и P , но намного быстрее. Иначе P называют *непредсказуемым*.

...

Основная идея (непредсказуемого вычисления – *прим. ред.*) состоит в том, что не существует существенно более быстрого способа получить P , иначе как провести вычисление P . В качестве простого примера предсказуемого вычисления рассмотрим определение того, является ли вводимое целое число четным или нечетным. Медленный путь состоит в долгом делении на два для получения частного и выяснения того, будет ли остаток равен нулю или единице. Быстрый или сокращенный путь вычисления того же результата состоит в том, чтобы только рассмотреть последнюю цифру вводимого числа и сказать, что число четное, если эта цифра равна нулю, двум, четырём, шести или восьми. Медленное вычисление четности путем нудного деления предсказуемо в нашем смысле, потому что намного более быстрое вычисление по последней цифре производит тот же результат...

Непредсказуемость вычислений становится значимой, когда мы перекладываем на компьютеры решение реальных задач. Известный пример связан с вычислением цифр числа «пи» – численного отношения длины окружности к диаметру.

Для этого числа, которое начинается с цифр 3.14159..., известно, что его десятичная запись продолжается до бесконечности без появления периода. Тем не менее, существуют простые вычислительные методы для нахождения последовательных цифр числа «пи» посредством арифметических операций. Один из таких методов (не очень эффективный) состоит в последовательном суммировании членов бесконечного знакопередающегося ряда:

$$4 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 + 4/13 - \dots$$

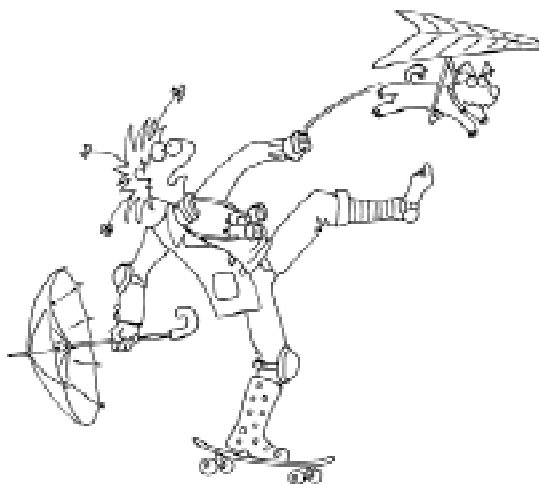
В середине 80-х мой старый друг и компьютерный фанатик Уильям Госпер однажды поставил мировой рекорд в вычислении числа «пи». Он вычислил семнадцать миллионов его цифр. Здесь я привожу первую сотню цифр, начиная с семнадцатимиллионной цифры, взятые из одного из электронных писем Госпера:

6978965266 4312708718
8987022892 7339840950
1815706767 7105940124
6541910101 0611655655
1475202499 7781719847.

...

До того, как вычисления Госпера были сделаны, не было способа узнать, что семнадцатимиллионная цифра – это шесть. И единственный способ получить ее – было дать возможность сверхмощной машине работать над ее нахождением в течение длительного времени. Да, ее значение предопределено законами математики, но реально она непредсказуема. Понятие компьютерных программ с непредсказуемым результатом удивительно, потому что мы полагаем, что все *детерминированное* неинтересно. Заметьте также, что, поскольку мы не относим себя к неинтересным изобретениям природы, мы воображаем, что мы должны быть *недетерминированными* и, таким образом, совсем не похожими на системы, управляемые жесткими правилами.

Но возможно мы неправы. Возможно, мы *детерминированы*, но *непредсказуемы*. Я уже упоминал, что невыполнимость относительна, она зависит от системы, которую вы намереваетесь использовать для данного вычисления. Что-то невыполнимо для данной системы, если оно выполняется дольше, чем вы считаете разумным. Непредсказуемость, напротив, является абсолютным



...мы должны быть *недетерминированными*...

понятием. Вычисление непредсказуемо, если не существует другого вычисления, которое делает то же, но существенно быстрее.

Часто для того, чтобы прояснить вновь введенные определения, полезно изучить возможные связи между ними. Как выполнимость и предсказуемость соотносятся между собой, если временно ограничить наше внимание вычислениями на персональных компьютерах? Как оказывается, все четыре возможных комбинации возможны.

Выполнимые и предсказуемые

Это самый простейший вид вычислений. Я имею здесь в виду тривиальные вычисления типа умножения семи на тысячу. Не дотрагиваясь даже до карандаша и бумаги, мы знаем, что $7 \cdot 1000 = 7000$. Вычисление предсказуемо. Мы знаем, как его сделать, не входя в детали. Но если нас попросят описать все в деталях, мы без труда сделаем это. Итак, вычисление является как выполнимым, так и предсказуемым.

Выполнимые и непредсказуемые

Это вычисления, которые интересуют ученых-компьютерщиков больше всего. Среди них есть вычисления, которые мы реально можем выполнить, но для которых нет быстрого пути угадать результат заранее. В таких случаях компьютерная система делает что-то важное для нас. Вычисление обнаруживает факт, который мы не способны знать заранее.

Невыполнимые, но предсказуемые

Предположим, что вычисление состоит в очень простой задаче, подобной замене каждого символа входной строки нулем. Для каждой данной входной строки выходная строка предсказуема: это строка такой же длины, состоящая из нулей. Но если входную строку взять безумной длины – вообразим сообщение галактических размеров, длинной в газиллион букв – тогда нет выполнимого способа ввести эту строку в настольный компьютер и ожидать ответа в течение любого разумного отрезка времени. Таким образом, в этом смысле вычисление невыполнимо, даже при том, что возможный выход предсказуем.

Невыполнимые и непредсказуемые

Рест уверяет, что все, что происходит внутри наших голов, является и непредсказуемым, и невыполнимым, если иметь в виду существующие электронно-вычислительные машины. Но, так или иначе, головы-то наши работают!

Теперь поговорим о классификации вычислений, которая была предложена Стивеном Вольфрамом в 80-х г. XX в. Вольфрам заметил, что имеются четыре главных типа поведения для произвольных вычислений, которые выполняются продолжительное время.

Класс первый. Вычисление входит в стационарное состояние.

Класс второй. Вычисление производит повторяющийся результат, зацикливается.

Класс третий. Вычисление производит случайную мешанину.

Класс четвертый. Вычисление производит потрясающие, взаимодействующие, неповторяющиеся образцы.

Довольно легко понять, как выглядят первый и второй классы (рисунок 4). Их сущность... состоит в том, что выход вычисления не приносит неожиданностей.

Что касается третьего класса, ... то это сумбурные, случайно выглядящие вычисления без очевидного порядка или структуры на выходе.

Класс четвертый, с другой стороны, можно сравнить с никогда не повторяющимися тканями узорами, полученными переплетениями нитей. Вычисления четвертого класса могли бы быть охарактеризованы как наличие целеустремленного поведения. Я люблю использовать слово «потрясающий» (в оригинале *gnarly*) для процессов четвертого класса – «потрясающий» в смысле замысловато закрученных корней дерева, больших океанских волн, или обветренных человеческих лиц.

Не вдаваясь в детали, о первом и втором классах можно говорить как о *простых*, а о третьем и четвертом – как о *сложных*.

Границы между классами вычислений не являются очень жесткими. Иногда различия между третьим и четвертым классом неочевидны. И не всегда ясно, принадлежит ли система ко второму или четвертому классу: некоторые системы могут казаться интересными в течение очень долгого времени, и только затем процесс устанавливается и становится периодическим. Мы не всегда знаем, можем ли мы действительно найти «хороший» вход, который всегда будет порождать оригинальные выходы, показывая таким образом, что процесс принадлежит четвертому классу.

Критики Вольфрама жалуются, что его классы вычислений определены неформально и формальное определение класса вычислений оказывается неразрешимой проблемой («неразрешимой» в некотором формальном смысле, который определяется в математической теории алгоритмов). Вольфрам мог бы ответить, что использование грубых концепций типично для недавно развивающейся отрасли науки. Я соглашаюсь с ним. Я думаю, что его критики не видят леса за деревьями.

...

Все начальные исследования Вольфрама имели дело с выполнимыми вычислениями – в них он рассматривал реальные программы, которые можно запустить на компьютере. Но его классификация применяется одинаково хорошо к огромным вычислениям, выполняемым физическими и биологическими объектами. Ограничения наших цифровых кремниевых машин таковы, что мы не можем «выполнимо» смоделировать любую достаточно большую часть ре-

ального мира. Даже в этом случае очень полезно классифицировать «невыполнимые для компьютера» вычисления, которые мы видим вокруг нас.

...

Вольфрам выдвинул две гипотезы относительно его классов вычислений. Первая из них – Принцип Вычислительной Эквивалентности (для краткости ПВЭ).

Принцип Вычислительной Эквивалентности (ПВЭ).

Почти все процессы, которые не очевидно просты, могут рассматриваться как вычисления эквивалентной сложности.

...

ПВЭ в некотором смысле разочаровывает, поскольку предполагает, что когда Вы не можете дать простое объяснение естественному явлению, то это означает, что явление не только сложное, но имеет максимальную сложность. Все, что не очевидно просто, фактически относится к «потрясающим» явлениям.

Простой пример. Рассмотрим движение листьев на дереве. Физик мог бы описать это движение как управляемую ветром систему с многоколенным маятником. Но соответствующее вычисление относится к четвертому классу и, конечно, является сложным. Если следовать Принципу Вычислительной Эквивалентности, то «потрясающие» движения листьев должны быть столь же сложны, как и происходящее внутри нашего мозга. Стало быть, я трепещущий лист? Возможно, и так.

Помимо Принципа Вычислительной Эквивалентности, Вольфрам выдвигает вторую гипотезу, которой я называю ПВН или

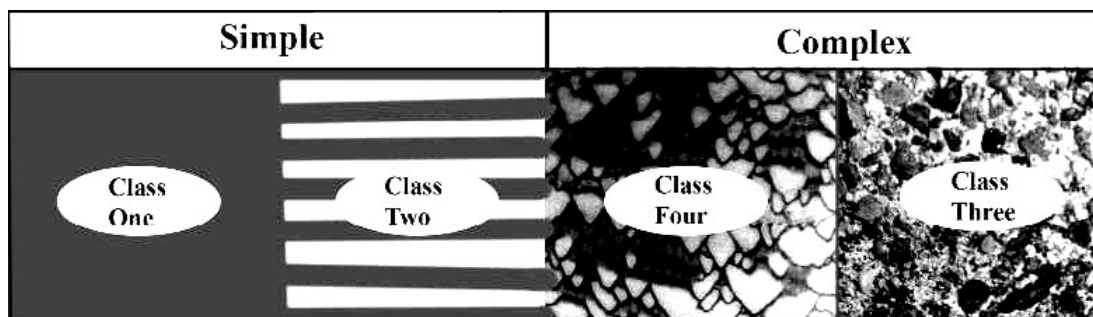
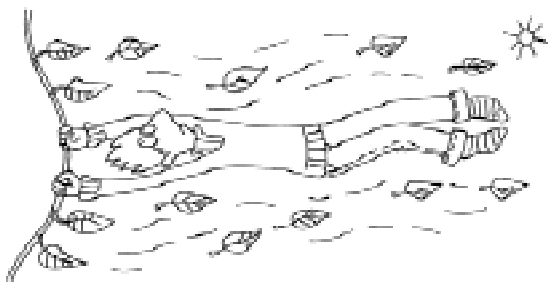


Рисунок 4.



Стало быть, я трепещущий лист?

Принцип Вычислительной Непредсказуемости.

Принцип Вычислительной Непредсказуемости (ПВН).

Большинство естественных сложных вычислений непредсказуемо.

...

Когда мы встречаем некоторый естественный процесс, мы можем часто смоделировать этот процесс как вычисление. И в некоторых редких случаях мы можем также смоделировать процесс несколькими простыми и довольно легко разрешимыми уравнениями. ПВН говорит, что последняя ситуация чрезвычайно редка. Вообще говоря, не имеется никакого быстрого способа предсказать результаты естественно возникающего вычисления.

Вольфрам не считает нужным явно заявить эти принципы, но они неявно содержатся в его труде «Новый вид науки», где формулируется его философия.

Он предпочитает использовать термины *приводимые* и *неприводимые* для того, что я называю *предсказуемым* и *непредсказуемым*.



Рисунок 5. Различные траектории движения небольшой стаи «птушек».

...

Таким образом Принцип Вычислительной Эквивалентности означает, что от систем, использующихся для предсказаний, нельзя ожидать, чтобы они делали вычисления более сложные, чем вычисления, которые происходят в системах, чье поведение мы пробуем предсказать. И отсюда следует, что для многих систем никакое систематическое предсказание не может быть сделано. Таким образом, не существует никакого общего пути к сокращению процесса эволюции, и поведение таких систем в вычислительном отношении должно рассматриваться как неприводимое (непредсказуемое).

Если поведение системы очевидно простое – повторяющееся или зацикливающееся – то в вычислительном отношении оно будет всегда приводимо (предсказуемо). Но из Принципа Вычислительной Эквивалентности следует, что практически во всех других случаях в вычислительном отношении оно будет неприводимо (непредсказуемо).

Я полагаю, что именно это является фундаментальной причиной того, что традиционная теоретическая наука не сумела продвинуться в изучении большинства типов систем, чье поведение не является крайне простым.

Оба принципа ПВЭ и ПВН фактически независимы друг от друга. В то время как последний используется, чтобы вывести непредсказуемость наблюдаемых в природе процессов, первый используется, чтобы вывести неразрешимость некоторых вопросов об этих процессах, где неразрешимость означает, что некоторые виды вопросов не могут быть решены никакими мыслимыми видами вычислений вообще.

Я соглашаюсь с Вольфрамом, что и ПВЭ и ПВН, вероятно, будут справедливы для всех интересных примеров проявления вычислений в природе, включая физические системы, биологическое развитие, человеческое мышление и социальные законы.

В заключение, я хочу представить еще одну идею. Когда вычисление приводит к интересному и неожиданному результату или поведению, это называют *явлением*. Я при-

веду три небольших примера, взятые соответственно из области «искусственной жизни», фракталов и клеточных автоматов.

...

Алгоритм «птишек»

В области информатики, которая изучает искусственную жизнь, на компьютерах пробуют моделировать поведение живых организмов. Классическое открытие в этой области – алгоритм «птишек» Крэйга Рейнольдса (от ред.: в оригинале *boids algorithm*, где *boids* – искусственное слово, возможно полученное объединением слов *birds* и *toys*; мы рискнули перевести его как «птишки», соединив птишек с игрушками; читайте об этих интересных искусственных птичках в ближайших номерах нашего журнала). Крэйг Рейнольдс (*Craig Reynolds*) обнаружил, что если группа моделируемых птиц, или птишек, повинуетя нескольким простым правилам, то птишки будут двигаться, как движется стая реальных птиц (рисунок 5). Это пример явления, так как несколько неожиданное стайное поведение появляется от соединения вычислений отдельных птишек.

...

Человеческий Улей

Фрактал – структура, которая имеет интересные детали на многих различных уровнях. Наиболее известный фрактал – множество Мандельброта (*Mandelbrot*) (об этом писалось в КИО: 1999, № 5, с. 58; 2000, № 3/4, с. 116, 2001, № 5, с. 55, 61, 73, 85). Предположим, что мы думаем о компьютерном экране как об области плоскости, сопоставляя каждому пикселю пару вещественных чисел. Предположим далее, что для каждого пикселя мы используем соответствующую пару чисел как вход для итеративного вычисления, которое заканчивается определением цвета данного пикселя.



...несколько неожиданное стайное поведение появляется от соединения вычислений отдельных птишек.

В 70-х годах прошлого уже века Benoit Mandelbrot исследовал широкий класс таких вычислений, которые производят чудесно сплетенные рекурсивные образцы. Будучи фрактальным (рекурсивным), множество Мандельброта имеет особенность, что можно увеличивать масштаб изображения и обнаруживать все новые и новые детали. Это пример явления, в котором мы имеем изобилие форм, являющихся результатом многократного применения очень простых правил (рисунок 6).

Клеточные автоматы

Клеточные автоматы – очаровательный тип вычислений, которые популяризирует Стивен Вольфрам. Для наших целей можно рассмотреть двумерный клеточный автомат как вычисление, в котором все пиксели экрана компьютера одновременно меняют цвет согласно одинаковым правилам.

Рассмотрим простой пример правил для такого клеточного автомата: положим, что каждый пиксель либо черный, либо белый, и что он меняет цвет, в зависимости от того, является ли большинство его ближайших соседей белыми или черными. Оказывается, что при таких правилах случайная россыпь черных и белых пикселей приходит в

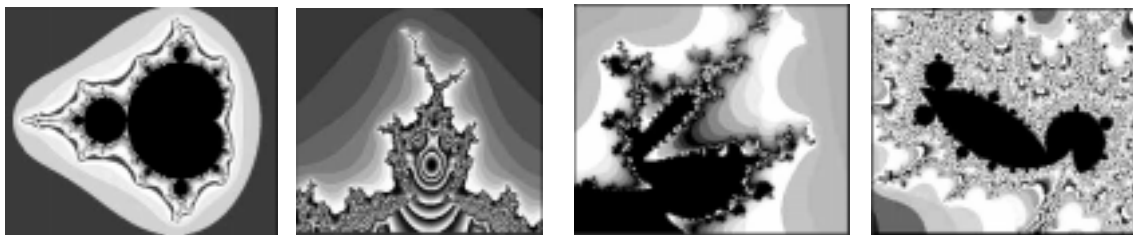


Рисунок 6. Фракталы.

состояние в форме гладких холмистых наплывов, мало чем отличаясь от капелек лавы (эти правила мы будем называть Автоматом Голосования).

Сущность стайного поведения, вычисления Мандельброта и работы Автомата Голосования в том, что нечто интересное возникает из простых правил и заданного состояния.

Понятие явления не совпадает с понятием непредсказуемости: мы можем видеть непредсказуемые вычисления, которые не порождают образцов высокого уровня организации. Унылое вычисление цифр числа «пи» являются тому примером. С другой стороны, мы можем видеть вычисления, которые производят замечательные образцы на стадии становления, но которые, в конечном счете, предсказуемы. Если Вы позволите автомату Голосования работать достаточно долго, один или другой цвет исчезнет совсем. Таким образом, Автомат Голосования, в конечном счете, – предсказуемое вычисление класса два.

А что можно сказать о стайном поведении «птушек» и множестве Мандельброта?

В большинстве ситуаций стайное поведение группы моделируемых птиц будет относиться к классу четыре и быть непредсказуемым. Время от времени будут появляться новые конфигурации стаи, один такой образец я наблюдал – пара птиц кружилась относительно друг друга по траектории сильно скрученной двойной спирали. Если бы мы могли бесконечно долго увеличивать масштаб изображения множества Мандельброта, то мы и в этом примере нашли бы непредсказуемое поведение класса четыре, по крайней мере, около границы множества (хотя проблема, является ли множество Мандельброта действительно непредсказуемым, как я полагаю, еще не решена).

На этом мы заканчиваем цитирование книги Руди Рукера, надеясь, что заинтересовали читателя необычной «машинной» философией Руди Рукера и Стивена Вольфрама.



Рисунок 7. Три изображения показывают случайное начальное расположение черных и белых пикселей для поля из 320×200 пикселей, вид поля после 30 шагов работы автомата Голосования и после 300 шагов.