Лабораторная работа. Клеточные автоматы

Теоретический материал

Клеточный автомат (КлА) – это дискретная модель, изучаемая в математике, теории вычислимости, физике, теоретической биологии и микромеханике. Включает в себя регулярную решётку ячеек, каждая из которых может находится в одном из конечного множества состояний, таких как 1 и 0. Решетка может быть любой размерности. Для каждой ячейки определено множество ячеек, называемых соседством. Например, соседство может быть определено как все ячейки на расстоянии не более 2 от текущей. Для работы клеточного автомата требуется задание начального состояния всех ячеек, и правил перехода ячеек из одного состояния в другое. На каждой итерации, используя правила перехода и состояния соседних ячеек, определяется новое состояние каждой ячейки. Обычно правила перехода одинаковы для всех ячеек и применяются сразу ко всей решётке.

Основное направление исследования клеточных автоматов – алгоритмическая разрешимость тех или иных проблем. Также рассматриваются вопросы построения начальных состояний, при которых клеточный автомат будет решать заданную задачу.

Клеточный автомат можно определить, как множество конечных автоматов, каждый из которых может находиться в одном из состояний:

$$\sigma \in \Sigma \equiv \{0,1,2...k-1,k\}$$

Изменение состояний автоматов происходит согласно правилу перехода:

$$\sigma_{i,j}(t+1) = \phi(\sigma_{k,l}(t) \mid \sigma_{k,l}(t) \in N)_{\sigma_{i,j}}$$

где N – множество автоматов, составляющих соседство. К примеру, соседство фон Неймана определяется как:

$$N_N^1(i,j) = \{\sigma_{k,l} \mid |i-k| + |j-l| \le 1\} \tag{1}$$

а соседство Мура:

$$N_M^1(i,j) = \{\sigma_{k,l} \mid |i-k| \le 1, |j-l| \le 1\}$$

Число всех возможных правил перехода определяется числом состояний – σ и количеством соседей – n и составляет:

$$N_r = \sigma^{\sigma^n}$$

Наиболее простые клеточные автоматы состоят из клеток, расположенных вдоль одной линии, каждая клетка может находиться в одном из k состояний. Правила эволюции могут быть записаны в виде:

$$a_i^{(t)} = f[\sum_{j=-r}^{j=r} a_j \cdot a_{i+j}^{(t-1)}],$$

где $a_i^{(t)}$ – состояние i-й клетки в момент t.

Определение. Автомат называется правильным, если $f(0)=0, f(a_{i-r,\dots\dots,}a_{i+r}) = f(a_{i+r},\dots.,a_{i-r})$. Если к тому же $\forall i : a_i = 1$, то автомат называется суммирующим.

Клеточные автоматы представляют собой такой частный случай больших сетевых структур который идеально соответствует целям проверки и развития ряда идей аналитико-имитационного моделирования, связанных с анализом асимптотических структурных свойств больших сетевых структур широкого класса.

Примеры моделирования клеточных автоматов

Существуют некоторые классические и базовые установленные правила клеточных автоматов [Wolfram (1994), Hegselmann and Flache (1998), Casti (1992), Toffoli & Margolus (1987)]:

- автомат состоит из постоянной N-мерной решётки ячеек.

- каждая ячейка или клетка в автомате, может находится в одном из конечных определённых состояний.

- автомат развивается с течением определённого количества временных этапов.

- состояние всех клеток в автомате обновляется одновременно на каждом шаге.

- клетки изменяют своё состояние в соответствии с локальными правилами, другими словами состояния перехода каждой ячейки определяется из состояния самой ячейки и её соседей.

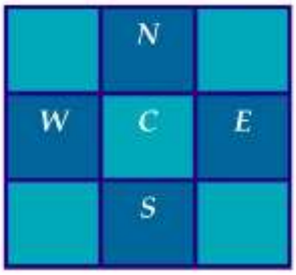


Рисунок 1. Правило четырёх соседей фон Неймана, здесь клетка C имеет четыре соседа (E, N, W и S), состояние которых и будет определять её дальнейшую судьбу.

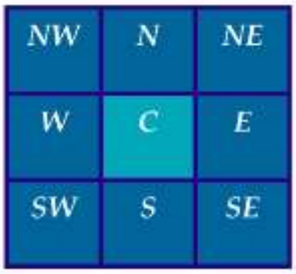


Рисунок 2. Правило восьми соседей Мура, в данном случае клетка C имеет уже восемь соседей (N, NE, E, SE, S, SW, W, NW).

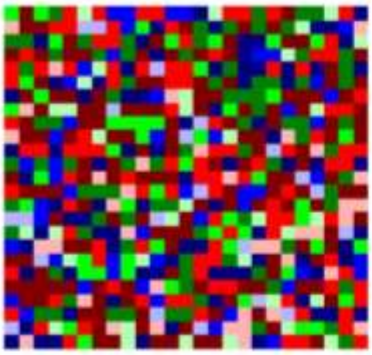


Рисунок 3. Двумерный клеточный автомат, в котором клетки ограничены в квадратной сетке размером 25x25, различное состояние каждой клетки определяется её цветом, который закодирован определённым тоном и оттенком.

В данной двумерной плоской структуре решетки, имеется одна проблема, связана она с тем, что на левом крае решётки, клетки не имеют западных соседей, из-за этого на восточной стороне так же уменьшается количество соседей у граничных клеток. Подобное положение ячеек у северной и южной стороны.

Исправляется этот недочёт «приклеиванием» левого края решётки к правому, а верхний край приклеиваются к нижней части. Это преобразует плоскую структуру, к структуре решётки формы тора. В таком виде у каждой ячейки имеется полноценное количество соседей.

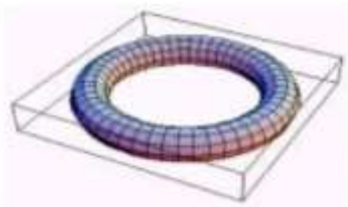


Рисунок 4. Двумерный клеточный автомат, в котором левая часть "приклеена" к правой, а верхняя к нижней.

Если представить, что τ определяет правило перехода для каждой ячейки, а $C(t)$ представляет собой состояние ячейки C на шаге времени t, с аналогичными обозначениями для других клеток, то состояние ячейки *C* при следующем шаге *t+1*, для правил окрестности фон Неймана, будет определяться выражением:

$$c(t+1) = \tau[c(\tau), e(\tau), n(\tau), W(t), S(t)].$$

Это правило применяется одновременно и несколько раз для каждой ячейки в автомате, на каждом этапе эволюции. Это правило может быть очень простым или сложным.

Самовоспроизведения в тривиальном смысле – без использования конфигураций, включающих в себя машину Тьюринга, – добиться легко. В этой системе ячейки могут находиться лишь в двух состояниях, каждая из них, имеет четырёх соседей, а правила перехода сводятся к следующим. Каждая клетка, имеющая в момент времени *t* четное число (0, 2, 4, ...) живых соседей, в момент времени *t+1* становится пустой (то есть переходит в нулевое состояние или, если она уже находилась в нулевом состоянии, остаётся в нём). Каждая клетка, имеющая в момент времени *t* нечетное число (1, 3, ...) соседей, в момент времени *t+1* становится живой (то есть переходит в ненулевое состояние или сохраняет его, если она уже в нём находилась). Через 2^n ходов (число *n* зависит от выбора конфигурации) любая конфигурация живых клеток в таком пространстве воспроизведет себя четыре раза: одна копия расположится справа, другая – слева, третья – сверху, четвёртая – снизу оттого (уже пустого) места, где находилась начальная конфигурация. Новая конфигурация через 2^nшагов снова размножится (с коэффициентом воспроизводства, равным 4) и т.д. Число копий увеличивается в геометрической прогрессии 1, 4, 16, 64... На рисунке 5 показаны полтора цикла размножения тримино в форме прямого угла.

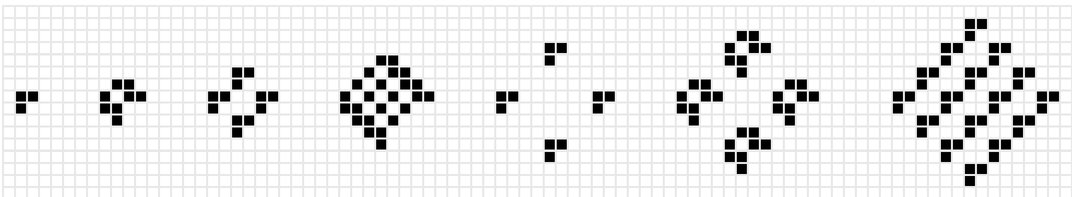


Рисунок 5. Эволюция тримино в форме прямого угла в шести шагах.

На рисунке 6 показано сорок пятое поколение организма, родившегося из одной-единственной фишки, стоявшей в центральной клетке.

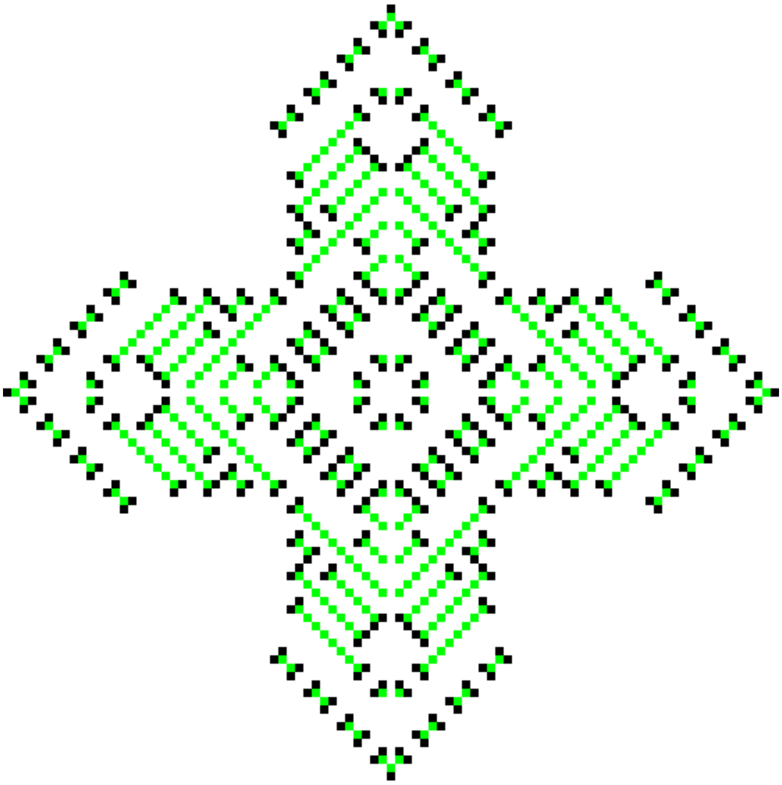


Рисунок 6. Сорок пятое поколение клеточного автомата С. Улама, произошедшего из единственной фишки в центре.

Клетки в игре Улама могут находиться в двух состояниях, но соседними считаются клетки, примыкающие к данной лишь по вертикали, но не по диагонали («соседи» в смысле фон Неймана). Рождение фишки происходит на клетке, имеющей одного и только одного соседа, а все клетки поколения n погибают после рождения поколения $n+2$. Иначе говоря, на любом этапе эволюции выживают лишь два последних поколения. На рисунке черными изображены новорожденные клетки (их 444). Зеленые клетки предыдущего поколения (их 404) исчезнут на следующем ходу. Обратите внимание на характерную деталь, которую Улам назвал «обглоданной костью». Улам проводил эксперименты и с такими играми, в которых две конфигурация могли расти до тех пор, пока они не сталкивались. В следующей за столкновением «битве» одной стороне иногда удавалось одержать верх над другой, а иногда обе армии исчезали. Улам рассмотрел также игры на трёхмерных досках – кубических «паркетах», заполняющих всё пространство.

Аналогичные игры можно вести и на бесконечных досках, клетки которых имеют форму равносторонних треугольников и правильных шестиугольников.

Клеточный автомат «Перемешивающая машина» был предложен М. Герхардом и Х. Шустером. Правила взаимодействия ячеек автомата Герхарда и Шустера удобно излагать, используя эпидемическую терминологию. Отдельная клетка может находиться в некотором множестве состояний. Обозначим их целыми числами от 0 до n. Клетки, находящиеся в состоянии 0, будем называть здоровыми, а клетки, находящиеся в состоянии *n* – больными. Все остальные промежуточные состояния интерпретируются как зараженность различной степени. Иными словами, клетка в рассматриваемой модели может быть либо здорова, либо заражена, либо больна. В зависимости от того, к какому из этих трех типов клетка относится в данный момент времени *t*, ее судьба в момент времени *t+1* складывается по-разному. Если клетка здорова (то есть ее состояние *x(i,j)*=0), то ее состояние в момент времени *t+1* будет определяться формулой:

$$Y(i,j) = Round(\frac{A}{k1,0}) + Round(\frac{B}{k2,0})$$

Здесь *A* – это количество зараженных, а *B* – количество больных клеток в окрестности. Константы *k1* и *k2* представляют собой некие изначально задаваемые управляющие параметры. Функция Round возвращает округленное до указанного количества знаков значение выражения.

Если в момент времени *t* клетка заражена (то есть *x(i,j)* имеет любое значение, большее 0, но меньшее n), то в момент времени *t+1* ее состояние будет определяться формулой:

$$Y(i,j) = Round(\frac{S}{A,0}) + G$$

Здесь, как и прежде, *A* – количество зараженных соседей в окрестности клетки, а *S* – это сумма состояний всех соседних клеток, включая саму *X(i,j)*. Константа *G* также представляет собой некий изначально задаваемый управляющий параметр. Следует помнить, что *Y(i,j)* ни при каких условиях не может превышать максимального значения *n*. Поэтому, если в процессе вычислений значение выходит за эти рамки, его следует приравнять к *n*.

Наконец, если клетка в момент времени t больна (то есть, ее состояние равно *n*), то в следующий момент времени *t+1* она всегда переходит в состояние 0, то есть выздоравливает.

Программный код, моделирующий поведение перемешивающей машины:

```
Dim x(210, 210) As Integer
Dim y(210, 210) As Integer
```

```
Dim c As Byte
Dim n As Byte
Dim m As Byte
Dim k1 As Single
Dim k2 As Single
Dim g As Byte
Private Sub Form_Click()
For f = 1 To 150
 x(0, 0) = x(n, n): x(0, n + 1) = x(n, 1): x(n + 1, 0) = x(1, n): x(n + 1, n + 1) = x(1, 1)
 For k = 1 To n
  x(0, k) = x(n, k): x(k, 0) = x(k, n): x(n + 1, k) = x(1, k): x(k, n + 1) = x(k, 1)
 Next k
 For i = 1 To n
  For j = 1 To n
   Select Case x(i, j)
    Case Is = 0
    a = 0: b = 0
    For k = i - 1 To i + 1
     For l = j - 1 To j + 1
      Select Case x(k, l)
       Case 1 To m - 1
       a = a + 1
       Case Is = m
       b = b + 1
      End Select
     Next l
    Next k
    y(i, j) = Round((a / k1), 0) + Round((b / k2), 0)
    blue = 0: red = 0: green = 0
    Case 1 To m - 1
    a = 0: s = 0
    For k = i - 1 To i + 1
     For l = j - 1 To j + 1
      If x(k, l) > 0 And x(k, l) < m Then a = a + 1
      s = s + x(k, l)
     Next l
    Next k
    y(i, j) = Round((s / a), 0) + g
    If y(i, j) > m Then y(i, j) = m
    Case Is = m
    y(i, j) = 0
   End Select
   c = (x(i, j) / m) * 255
   Line (4 * (i - 1), 4 * (j - 1))-(4 * i - 1, 4 * j - 1), RGB(c, c, c), BF
  Next j
```

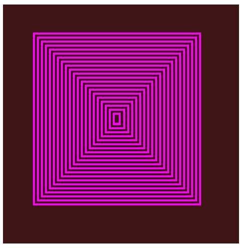
```
 Next i
 For i = 1 To n
  For j = 1 To n
   x(i, j) = y(i, j)
  Next j
 Next i
Next f
End Sub
Private Sub Form_Load()
Randomize
n = 25
m = 100
k1 = 5
k2 = 4
g = 28
For i = 1 To n
 For j = 1 To n
  x(i, j) = m * Rnd
 Next j
Next i
End Sub
```

Для того, чтобы не изменять общие правила на границах поля, клеточная матрица «сворачивается» в тор. Процедура сворачивания лишает поверхность ее границ, а значит, и «пограничных» проблем. Для хранения состояний ячеек в предыдущем и последующем поколении используются два двухмерных массива – *X(i,j) и Y(i,j)*. Результат выводится на форму в виде мозаики квадратиков, раскрашенных в градации серого цвета в соответствии с состояниями ячеек в момент времени *t*. При моделировании перемешивающей машины рекомендуется использовать анимационные технологии, поскольку динамика образования и движения волновых фронтов производит совершенно неповторимое впечатление именно в движении.

Задание на лабораторную работу. Программно реализовать клеточный автомат согласно варианту

Вариант 1. Лабиринт

Состояние автомата



Набор правил

A 0 1 -1 A

A 1 2 -1 A

A 2 3 -1 A

A 3 4 -1 A

A 4 5 -1 A

A 5 6 1 B

B 0 1 1 A

B 5 6 -1 B

B 6 7 -1 B

B 7 8 -1 B

B 8 9 -1 B

B 9 10 -1 B

B 10 11 -1 B

B 11 12 -1 B

B 12 13 -1 B

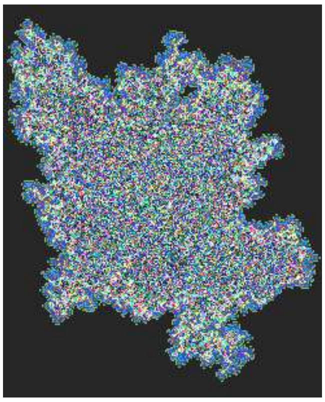
B 13 14 -1 B

B 14 15 -1 B

B 15 0 -1 B

Вариант 2. Остров невезения

Состояние автомата



Набор правил

A 0 1 -1 B

A 1 2 -1 B

A 2 3 -1 B

A 3 4 -1 B

A 4 5 1 B

A 5 6 1 B

A 6 7 1 B

A 7 8 1 B

A 8 9 -1 B

A 9 10 -1 B

A 10 11 -1 B

A 11 12 -1 B

A 12 13 1 B

A 13 14 1 B

A 14 15 1 B

A 15 0 1 A

B 0 1 1 B

B 1 2 1 A

B 2 3 1 A

B 3 4 1 A

B 4 5 -1 A

B 5 6 -1 A

B 6 7 -1 A

B 7 8 -1 A

B 8 9 1 A

B 9 10 1 A

B 10 11 1 A

B 11 12 1 A

B 12 13 -1 A

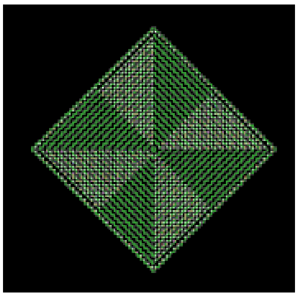
B 13 14 -1 A

B 14 15 -1 A

B 15 0 -1 A

Вариант 3. Вертушка

Состояние автомата



Набор правил

A 0 2 0 C

A 2 0 0 B

B 2 2 1 A

B 15 2 1 A

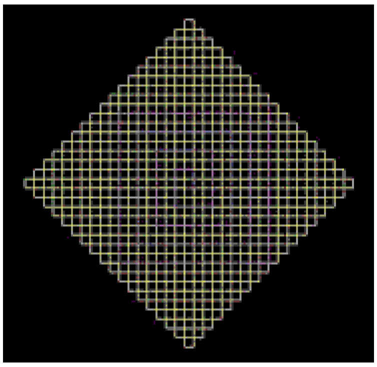
C 2 0 -1 A

C 0 15 -1 A

C 15 2 -1 A

Вариант 4. Желтый квадрат

Состояние автомата



Набор правил

A 0 14 0 B

B 0 14 0 C

C 0 14 0 E

E 0 14 0 F

F 0 14 0 J

J 0 13 -1 A

A 14 13 0 A

A 13 14 0 A

J 14 13 0 A

J 13 14 1 A