

# Линейное по времени конструирование изображений клеточными автоматами

Е. Е. Титова

В работе рассматривается задача конструирования изображений клеточным автоматом на прямоугольном экране. Для случаев, когда элементарный автомат имеет 3, 4 и 5 состояний, и когда число состояний неограниченно растет, получены оценки времени конструирования изображений, линейно зависящие от размеров экрана. Найден алгоритм построения изображений для экрана, имеющего один свободный вход.

**Ключевые слова:** клеточный автомат, генератор, универсальный экран, конструирование изображений, оценка времени конструирования изображения.

## 1. Введение

Клеточный автомат — это математический объект с дискретными пространством и временем. Каждое положение в пространстве представлено отдельной клеткой, а каждый момент времени — дискретным временным шагом или поколением. Состояние каждой пространственной клетки определяется очень простыми правилами взаимодействия. Эти правила предписывают изменения состояния каждой клетки в следующем такте времени в ответ на текущее состояние соседних клеток. При этом для разных клеток правила изменения состояний могут быть разными.

Частным случаем клеточных автоматов являются однородные структуры, которые представляют собой дискретную математическую модель широкого класса реальных систем вместе с протека-

ющими в них процессами, таких как физические среды, в которых реализуются тепловые и волновые явления, химические растворы с реакциями в них, биологические ткани, в которых происходит обмен веществ, технические схемы управления, производящие переработку механических и электрических сигналов, вычислительные схемы и т. п.

Понятие однородной структуры возникает при выборе в качестве преобразователя информации, стоящего в клетке пространства, конечного автомата — простейшей управляющей системы. Если задать начальные состояния автоматов, то в схеме начнется изменение состояний автоматов, определяемое законами функционирования автоматов и связями между ними. Явление глобального изменения этих состояний и является главным объектом изучения в теории однородных структур.

Впервые идея клеточных автоматов отмечена в работах Дж. фон Неймана в 1940-х годах. Вплоть до конца 60-х идея клеточных автоматов была забыта и лишь в 1970 году Джон Конвей, математик Кембриджского университета, попытался упростить идеи предложенные Нейманом, и в конце концов описал ныне широко известный двумерный клеточный автомат, названный игра «Жизнь» («Life»). Описанный выше вариант однородной структуры использовался А. Берксом, Э. Муром, Майхиллом и другими. На механико-математическом факультете МГУ им. Ломоносова исследованием свойств однородных структур занимались В. Б. Кудрявцев, А. С. Подколзин, А. А. Болотов; результатом их работы стала монография [1].

В настоящей работе рассматривается клеточный автомат следующего вида: он представляет собой однородную структуру, заданную на конечном прямоугольнике, причем состояние структуры управляется входами, расположенными по верхней и левой границам прямоугольника (либо одним входом, расположенным в углу прямоугольника). На двух других границах входы изначально доопределены нулями и не меняются. Эта конструкция называется экраном. Последовательности для управляющих входов генерируются другим, внешним автоматом. Задача состоит в построении такой однородной структуры и такого внешнего автомата, чтобы через какое-то время после подачи его выходов на управляющие входы структуры на

экране сформировалась заранее заданная конфигурация состояний элементарных автоматов, причем она сохраняется сколь угодно долго при условии что на управляющие входы поступают только последовательности из нулей. Данная работа является продолжением [3], в которой была получена нижняя оценка числа состояний элементарного автомата (3 состояния) и приведена нелинейная оценка времени построения изображения при конечном числе состояний элементарного автомата и нижняя оценка времени построения изображений при неограниченном числе состояний клеточного автомата.

Работа выполнена на кафедре Математической теории интеллектуальных систем. Автор выражает искреннюю благодарность заведующему кафедрой д.ф.-м.н., профессору, академику В. Б. Кудрявцеву и всему коллективу кафедры за опыт и знания, полученные во время обучения. Особую признательность хочется высказать научному руководителю д.ф.-м.н., профессору Э. Э. Гасанову за постановку задачи и научное руководство.

## 2. Основные понятия и формулировка результатов

Пусть имеется конечный автомат  $\mathcal{A}$  с четырьмя входами.  $E_q = \{0, 1, \dots, q - 1\}$  — множество состояний автомата,  $\varphi(q, l, r, u, d)$  — функция переходов состояний автомата, где  $q$  — текущее состояние,  $l, r, u, d$  принимают значения из  $E_q$ , причем имеет место свойство  $\varphi(0, 0, 0, 0, 0) = 0$ . Функция выхода автомата совпадает с функцией переходов состояний. Описанный автомат будем называть *элементарным* автоматом. Пусть имеется прямоугольник размера  $n \times m$ ,  $m, n \in \mathbb{N}$ . В каждую клетку прямоугольника поместим по одному экземпляру одного и того же элементарного автомата  $\mathcal{A}$ . К входам этого автомата присоединим выходы автоматов, стоящих в четырех соседних с ним клетках, то есть у него имеется *левый* вход  $l$ , *правый* вход  $r$ , *верхний* вход  $u$  и *нижний* вход  $d$  соответственно и  $q$  — текущее состояние автомата. Выходом автомата является его состояние в этот момент времени. Автоматы, стоящие в первой и  $n$ -й строках и в первом и  $m$ -м столбцах будем называть *граничными авто-*

*матами.* Для этих автоматов определены не все входы. Будем считать, что у автоматов, стоящих в  $m$ -м столбце правый вход всегда нулевой, и у автоматов, стоящих в  $n$ -й строке нижний вход всегда нулевой. Неопределенные входы автоматов первой строки и первого столбца будем называть *свободными входами*. Описанную конструкцию назовем  $(n, m)$ -экраном, то есть  $(n, m)$ -экраном  $S$  назовем тройку  $S = \langle \mathcal{A}, n, m \rangle$ , где  $\mathcal{A}$  — элементарный автомат,  $n$  и  $m$  — соответственно количество строк и столбцов прямоугольника.

Состояния 0 и 1 элементарного автомата будем называть *метками*. Если каждый элементарный автомат экрана находится в нулевом или единичном состоянии, то такую конфигурацию состояний экрана будем называть *черно-белой конфигурацией*. Черно-белую конфигурацию назовем *изображением*, если эту конфигурацию можно удерживать сколь угодно долго, подавая на свободные входы автоматов нулевые значения. *Кодом  $K$*  изображения назовем матрицу  $n \times m$ , состоящую из нулей и единиц. Скажем, что изображение  $\mathfrak{Z}_K$  *соответствует данному коду  $K$* , если положение нулей и единиц в изображении и в коде совпадают.

Пусть также имеется *внешний* автономный автомат  $\mathcal{A}_e$  с множеством состояний  $E_{q'} = \{0, 1, \dots, q' - 1\}$ , который генерирует последовательности входных символов из множества  $E_q$  для свободных входов элементарных автоматов. То есть выход внешнего автомата — два вектора:  $(a_1, \dots, a_n)$  и  $(b_1, \dots, b_m)$ , где  $a_i, b_j \in E_q$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, m$ . В начальный момент времени внешний автомат находится в начальном, нулевом состоянии, а на экране имеется черно-белое изображение (либо экран из одних нулей). Задача состоит в построении такого элементарного автомата, что для любого кода  $K$  можно построить такой внешний автомат, что при подаче выходов внешнего автомата на входы экрана, на экране генерируется изображение  $\mathfrak{Z}_K$ , соответствующее коду  $K$ . Обозначим  $\mathfrak{Z}(n, m)$  — множество изображений размера  $(n \times m)$ , соответствующих всем возможным кодам.

*Генератором  $G$*  назовем пару  $G = \langle \mathcal{A}_e, S \rangle$ , где  $\mathcal{A}_e$  — внешний автомат,  $S$  — экран. Если  $\mathcal{A}_e$  — автономный автомат с  $n + m$  выходами,  $S$  —  $(n, m)$ -экран с черно-белым начальным состоянием, то скажем, что генератор  $\langle \mathcal{A}_e, S \rangle$  формирует изображение с кодом  $K$ , если через некоторое время на экране появляется изображение, соот-

ветствующее коду  $K$ . Скажем, что экран  $S = \langle \mathcal{A}, n, m \rangle$  — универсальный, если для любого кода  $K$  существует внешний автомат  $\mathcal{A}_e^K$ , такой что генератор  $G = \langle \mathcal{A}_e^K, S \rangle$  формирует изображение  $\mathfrak{F}_K$ , соответствующее коду  $K$ . Множество всех универсальных  $(n, m)$ -экранов обозначим через  $\mathcal{U}(n, m)$ . Далее будет показано, что  $\mathcal{U}(n, m) \neq \emptyset$  при любых  $n, m \geq 3$ ,  $n, m \in \mathbb{N}$ . Если  $S = \langle \mathcal{A}, n, m \rangle$  — экран, то через  $Q(S)$  обозначим число состояний элементарного автомата  $\mathcal{A}$ . Обозначим

$$Q(n, m) = \min_{S \in \mathcal{U}(n, m)} Q(S).$$

Для генератора  $\langle \mathcal{A}_e, S \rangle$  через  $T(\mathcal{A}_e, S)$  обозначим момент времени, после которого конфигурация экрана не изменяется. Считаем, что  $T(\mathcal{A}_e, S) = \infty$ , если стабилизация не наступает. Если генератор  $\langle \mathcal{A}_e, S \rangle$  формирует изображение  $\mathfrak{F}_K$ , соответствующее коду  $K$ , то  $T(\mathcal{A}_e, S)$  — время формирования изображения  $\mathfrak{F}_K$ .

Пусть  $S$  — универсальный экран,  $\mathfrak{F}$  — изображение. Через  $\mathcal{G}(S, \mathfrak{F})$  обозначим множество генераторов  $\langle \mathcal{A}_e, S \rangle$ , формирующих изображение  $\mathfrak{F}$ . Обозначим

$$\begin{aligned} T(S, \mathfrak{F}) &= \min_{\langle \mathcal{A}_e, S \rangle \in \mathcal{G}(S, \mathfrak{F})} T(\mathcal{A}_e, S), \\ T(S, n, m) &= \max_{\mathfrak{F} \in \mathfrak{F}(n, m)} T(S, \mathfrak{F}), \\ T(n, m) &= \min_{S \in \mathcal{U}(n, m)} T(S, n, m), \\ T(n, m, q) &= \min_{S \in \mathcal{U}(n, m), Q(S) \leq q} T(S, n, m). \end{aligned}$$

**Теорема 1.** Если  $n, m \in \mathbb{N}$ ,  $m, n \geq 2$ , то

$$\begin{aligned} T(n, m, 3) &\leq n + m + \min(n, m) + 1, \\ T(n, m, 4) &\leq n + m + \min(n, m) - 1, \\ T(n, m, 5) &\leq 2 \min(n, m) + 2. \end{aligned}$$

Доопределим теперь нулями все свободные входы экрана кроме одного, первого в верхней строке. Соответственно внешний автомат имеет только один выход. Таким образом, теперь задача состоит в построении генератора, который строит любое наперед заданное изображение, используя только один свободный вход. Показано, что для

этого достаточно чтобы элементарный автомат имел конечное число состояний. Обозначим  $T^1(n, m, 8)$  — время построения изображения на указанном экране с одним входом. Тогда имеет место

**Теорема 2.** Если  $n, m \in \mathbb{N}$ ,  $m, n \geq 2$ , то

$$T^1(n, m, 8) \leq 2mn + 4.$$

### 3. Линейная оценка времени построения изображения при ограниченном числе состояний элементарного автомата

В этом разделе приведем алгоритмы, которые при ограниченном числе состояний элементарных автоматов работают линейное время от  $n$  и  $m$ .

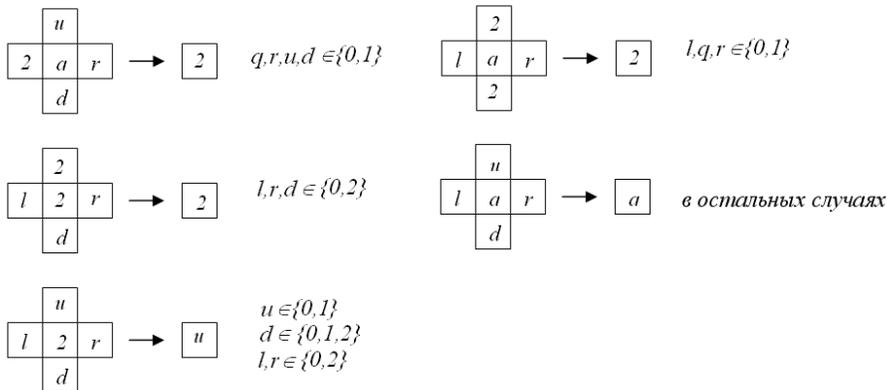


Рис. 1. Переходы состояний элементарного автомата с тремя состояниями.

**Лемма 1.** Для любых  $n, m \in \mathbb{N}$

$$T(n, m, 3) \leq n + m + \min(n, m) + 1.$$

**Доказательство.** Приведем универсальный  $(n, m)$ -экран и внешний автомат, для которых  $T(n, m, 3) \leq n + m + \min(n, m) + 1$ .

Без ограничения общности будем считать, что  $m \geq n$ .

Опишем элементарный автомат. По условию он имеет 3 состояния, то есть множество состояний —  $E_q = \{0, 1, 2\}$ . Функцию переходов состояний зададим следующим образом:

$$\varphi(q, 2, r, u, d) = 2 \text{ для любых } q, r, u, d \in \{0, 1\};$$

$$\varphi(2, l, r, 2, d) = 2 \text{ для любых } l, r, d \in \{0, 2\};$$

$$\varphi(2, l, r, u, d) = u \text{ для любых } u \in \{0, 1\}, d \in \{0, 1, 2\}, l, r \in \{0, 2\};$$

$$\varphi(q, l, r, 2, 2) = 2 \text{ для любых } l, q, r \in \{0, 1\} d \in \{0, 1\};$$

$$\varphi(q, l, r, u, d) = q \text{ в остальных случаях.}$$

На рисунке 1 изображены переходы состояний элементарного автомата.

Опишем внешний автомат, который соответствует изображению с кодом  $K$ . Первый выходной вектор (длины  $n$ ) в первый такт будет равен  $(2, 2, \dots, 2)_n$ , во второй и все последующие такты первый выходной вектор будет нулевым, то есть  $(0, 0, \dots, 0)_n$ .

Вектор  $(b_1, \dots, b_m)$ , подаваемый на верхнюю границу экрана будем строить по следующим правилам:

в первый такт это нулевой вектор, то есть  $(0, 0, \dots, 0)_m$ ;

далее при  $2 \leq i \leq m + 1$  в  $i$ -й такт в первых  $(i - 1)$  битах стоят 2, в остальных битах вектора стоят нули, то есть  $(2, \dots, 2, 0, \dots, 0)_m$ ;

при  $m + 2 \leq i \leq m + 2n$

если  $i = (m + 2) + 2s$ ,  $0 \leq s \leq n - 1$ , то в  $i$ -й такт выходной вектор равен  $(m - s)$ -й строке кода  $K$ ;

если  $i = (m + 2) + 2s + 1$ ,  $0 \leq s \leq n - 1$ , то в  $i$ -й такт в каждом бите выходного вектора стоит 2, то есть он равен  $(2, 2, \dots, 2)_m$ ;

при  $i \geq m + 2n + 1$  выходной вектор нулевой, то есть  $(0, 0, \dots, 0)_m$ .

Поясним, как работает алгоритм. При подаче выходов описанного внешнего автомата на свободные входы экрана состояние 2 распространяется по горизонтали слева направо, пока не заполнит весь экран. Далее на верхнюю границу с задержкой в один такт подаются друг за другом строки кода изображения начиная с нижней и заканчивая верхней строкой. Такая строка кода продвигается по экрану вниз до тех пор, пока и сверху и снизу от нее находятся строки из двоек. Если соседняя снизу строка состоит из нулей и единиц, то

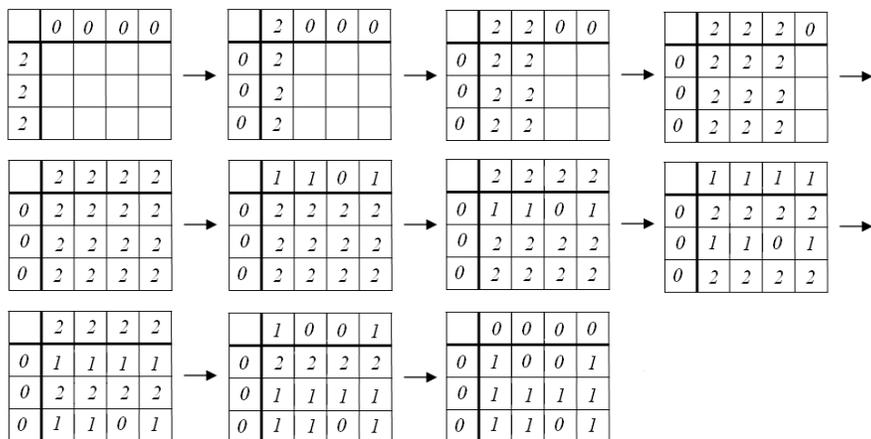


Рис. 2. Пример построения изображения по Алгоритму 3.

строка кода останавливается и дальше не двигается. Таким образом на экране снизу вверх по строкам восстанавливается изображение, соответствующее заданному коду. Оно появляется на экране в следующий такт после подачи на свободные входы экрана последнего ненулевого вектора, то есть на  $(m + 2n + 1)$ -й такт. Таким образом получена оценка  $T(n, m, 3) \leq m + n + \min(n, m) + 1$ , что и требовалось доказать.

Этот алгоритм назовем *Алгоритм 3* (*Алгоритм 1* и *Алгоритм 2* описаны в [3]). На рисунке 2 приведен пример построения изображения по этому алгоритму.

**Лемма 2.** Для любых  $n, m \in \mathbb{N}$

$$T(n, m, 4) \leq n + m + \min(n, m) - 1.$$

**Доказательство.** Приведем универсальный  $(n, m)$ -экран и внешний автомат, для которых  $T(n, m, 4) \leq n + m + \min(n, m) - 1$ .

Без ограничения общности будем считать, что  $m \geq n$ .

Опишем элементарный автомат. По условию он имеет 4 состояния, то есть множество состояний —  $E_q = \{0, 1, 2, 3\}$ . Функцию переходов состояний зададим следующим образом:

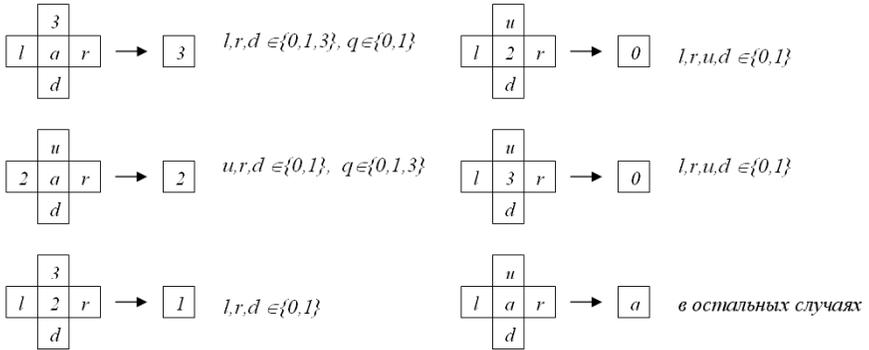


Рис. 3. Переходы состояний элементарного автомата с четырьмя состояниями.

- $\varphi(q, l, r, 3, d) = 3$  для любых  $l, r, d \in \{0, 1, 3\}, q \in \{0, 1\}$ ;
- $\varphi(q, 2, r, u, d) = 2$  для любых  $u, r, d \in \{0, 1\}, q \in \{0, 1, 3\}$ ;
- $\varphi(2, l, r, 3, d) = 1$  для любых  $l, r, d \in \{0, 1\}$ ;
- $\varphi(2, l, r, u, d) = 0$  для любых  $l, r, u, d \in \{0, 1\}$ ;
- $\varphi(3, l, r, u, d) = 0$  для любых  $l, r, u, d \in \{0, 1\}$ ;
- $\varphi(q, l, r, u, d) = q$  в остальных случаях.

На рисунке 3 изображены переходы состояний элементарного автомата.

Опишем теперь внешний автомат, соответствующий изображению, в котором все клетки находятся в состоянии 1, назовем этот автомат  $A_e^1$ . Из него затем получим внешние автоматы для всех остальных изображений.

Мы определили внешний автомат так, что его выход — два вектора  $(a_1, \dots, a_n)$  и  $(b_1, \dots, b_m)$ , где  $a_i, b_j \in E_q, i = 1, \dots, n, j = 1, \dots, m$ . Опишем отдельно как строить каждый из них.

Пока  $i < n - 1$  в  $i$ -й такт первый вектор имеет вид  $(0, \dots, 0)_n$ , то есть нулевой. Следующие  $n$  тактов с  $(n - 1)$ -го по  $(2n - 2)$ -й такты включительно первый выходной вектор строится так: в такт с номером  $j, n - 1 \leq j \leq 2n - 2$  в векторе  $(0, \dots, 0, 2, 0, \dots, 0)_n$  на  $(2n - j - 1)$ -м

месте стоит 2, на остальных местах — нули. Начиная с  $(2n - 1)$ -го такта первый вектор всегда нулевой.

При построении второго вектора различаются два случая:  $m \leq 2n - 1$  и  $m > 2n - 1$ .

Если  $m \leq 2n - 1$ , то вектор строится так: в  $i$ -й такт при  $i < m$  он имеет вид  $(\dots, 3, 0, 3, 0, 0, \dots, 0)_m$ , где до  $i$ -го бита включительно чередуются 0 и 3, причем так что на  $i$ -м месте стоит 3, с  $(i + 1)$ -го по  $m$ -й бит стоят нули; при  $m \leq i \leq 2n - 1$  вектор состоит из чередующихся 0 и 3, причем в  $m$ -й такт на последнем месте стоит 3, то есть  $(\dots, 3, 0, 3)_m$ , в  $(m + 1)$ -й такт на последнем месте стоит 0, то есть  $(\dots, 0, 3, 0)_m$  и т. д. выходные векторы чередуются. Далее если  $2n - 1 < i \leq 2n + m - 2$ , то вектор имеет вид  $(0, \dots, 0, 3, 0, \dots)_m$ , нули стоят на первых  $i - 2n + 1$  местах, на  $(i - 2n + 2)$ -м месте 3 и далее 0 и 3 чередуются. Начиная с  $(2n + m - 1)$ -го такта вектор нулевой.

Если  $m > 2n - 1$ , то вектор строится так: в  $i$ -й такт при  $i < 2n - 1$  он имеет вид  $(\dots, 3, 0, 3, 0, 0, \dots, 0)_m$ , где до  $i$ -го бита включительно чередуются 0 и 3, причем так что на  $i$ -м месте стоит 3, с  $i$ -го по  $m$ -й бит стоят нули; при  $2n - 1 \leq i \leq m$  вектор состоит из  $(i - 2n + 1)$  нулей, затем отрезка длины  $2n - 1$  из чередующихся 0 и 3, начинающегося с 3 (и заканчивающегося на 3) и дальше снова нули, то есть  $(0, \dots, 0, 0, 3, \dots, 0, 3, 0, \dots, 0)_m$ . При  $i \geq (m + 1)$  вектор состоит из  $j - (2n - 1)$  нулей, а затем чередующихся 3 и 0, причем на  $(j - 2n + 2)$ -м месте стоит 3. Начиная с  $(m + 2n - 1)$ -го такта вектор нулевой.

Итак, подавая выходы этого автомата на входы  $(n, m)$ -экрана с описанным выше элементарным автоматом получим изображение, в каждой клетке которого стоит единица, причем оно появится на экране в следующий такт после подачи последнего ненулевого входа, то есть в  $(m + 2n - 1)$ -й такт. В этом алгоритме состояние 2 передается автоматами слева направо, состояние 3 — сверху вниз. Автомат переходит в состояние 1 если он на такт раньше был в состоянии 2, а верхний автомат был в состоянии 3. При этом состояние 2 передается правому автомату. Другими словами единица появляется на пересечении двойки и тройки; кроме того если клетка находится в состоянии 2, а клетка над ней — в состоянии 0 или 1, то она перехо-

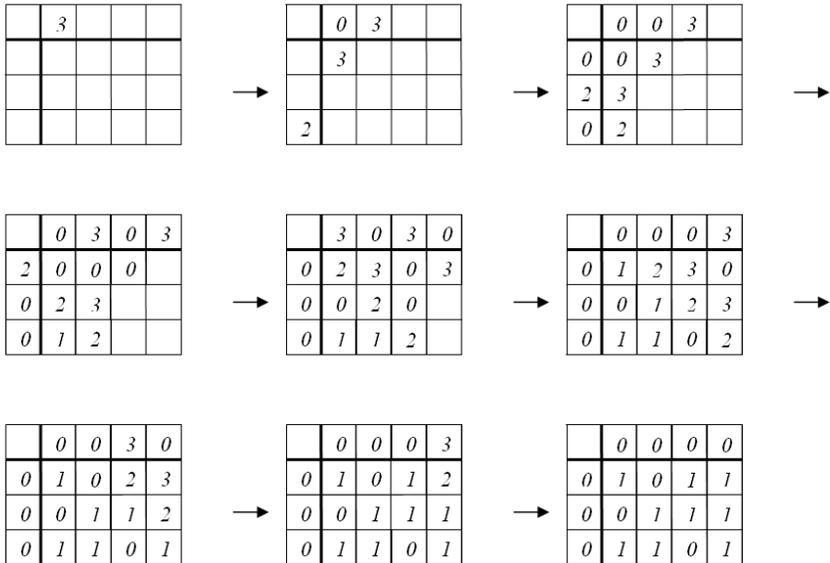


Рис. 4. Пример построения изображения по Алгоритму 4.

дит в состояние 0, тем самым мы можем очищать экран, если точку рисовать не нужно (например нужна строка из нулей).

Теперь легко понять, как из внешнего автомата для изображения из всех единиц получить внешний автомат для любого другого изображения. Для этого необходимо в выходных векторах поставить нули в тех битах, которые отвечают за построение «лишних точек».

Итак, пусть нам нужно построить изображение, в котором в клетке с координатами  $(i, j)$ ,  $1 \leq i \leq n, 1 \leq j \leq m$ , стоит 0. Преобразуем выходы автомата  $A_e^1$ , чтобы получить это изображение. Для этого изменим векторы, которые автомат выдает в  $((j - 1) + 2(m - i) + 1) = j + 2(m - i)$ -й такт: первый вектор оставим без изменения, а во втором в  $j$ -м бите 3 заменим на 0. Прделаем то же самое для каждой клетки изображения, в которой должен стоять 0. Если при этом получится, что один или несколько первых тактов оба выходных вектора нулевые, уберем их (и сделаем первым тактом тот такт, когда впервые на

выходе не оба вектора нулевые). Так мы получим внешний автомат для любого изображения.

Описанный алгоритм будем называть *Алгоритм 4*. На рисунке 4 приведен пример построения изображения.

Согласно *Алгоритму 4* любое изображение строится не дольше чем изображение, в котором каждая клетка находится в состоянии 1. Таким образом время построения любого изображения не превосходит  $(m + 2n - 1)$  или, в общем случае  $T(n, m, 4) \leq n + m + \min(n, m) - 1$ , что и требовалось доказать.

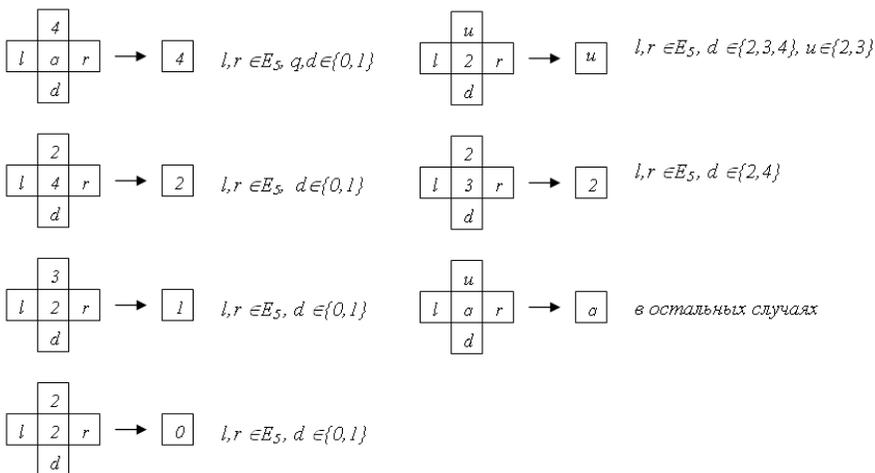


Рис. 5. Переходы состояний элементарного автомата с пятью состояниями.

**Лемма 3.** Для любых  $n, m \in \mathbb{N}$

$$T(n, m, 5) \leq 2 \min(n, m) + 2.$$

**Доказательство.** Приведем универсальный  $(n, m)$ -экран и внешний автомат, для которых  $T(n, m, 5) \leq 2 \min(n, m) + 2$ .

Без ограничения общности будем считать, что  $m \geq n$ .

Опишем элементарный автомат. По условию он имеет 5 состояний, то есть множество состояний —  $E_q = \{0, 1, 2, 3, 4\}$ . Левый и правый

входы элементарного автомата будут несущественными. Функцию переходов состояний зададим следующим образом:

$$\varphi(q, l, r, 4, d) = 4 \text{ для любых } l, r \in E_5, q, d \in \{0, 1\};$$

$$\varphi(4, l, r, 2, d) = 2 \text{ для любых } l, r \in E_5, d \in \{0, 1\};$$

$$\varphi(2, l, r, 3, d) = 1 \text{ для любых } l, r \in E_5, d \in \{0, 1\};$$

$$\varphi(2, l, r, 2, d) = 0 \text{ для любых } l, r \in E_5, d \in \{0, 1\};$$

$$\varphi(2, l, r, u, d) = u \text{ для любых } l, r \in E_5, d \in \{2, 3, 4\}, u \in \{2, 3\};$$

$$\varphi(3, l, r, 2, d) = 2 \text{ для любых } l, r \in E_5, d \in \{2, 4\};$$

$$\varphi(q, l, r, u, d) = q \text{ в остальных случаях.}$$

На рисунке 5 изображены переходы состояний элементарного автомата.

Опишем внешний автомат, который соответствует изображению с кодом  $K$ . Поскольку левый и правый входы элементарного автомата не влияют на переходы состояний, то на левую границу экрана будем подавать нулевые векторы. Векторы, подаваемые на верхнюю границу экрана будем строить по следующим правилам: в первый такт это вектор, в каждом бите которого стоит 4:  $(4, \dots, 4)_m$ . Далее при  $1 \leq i \leq n$  в  $(2i)$ -й такт это вектор, в каждом бите которого стоит 2:  $(2, \dots, 2)_m$ . В  $(2i + 1)$ -й такт это вектор, в котором на  $j$ -ом месте стоит 3, если в коде изображения  $K$  в клетке с координатами  $(m - (i - 1), j)$  стоит 1, иначе на  $j$ -м месте в векторе стоит 2. Начиная с  $(2n + 2)$ -го такта выходной вектор нулевой. В  $(2n + 2)$ -й такт на экране появится изображение, соответствующее коду  $K$ .

Поясним, как работает алгоритм. За один такт состояние 4 распространяется вниз на одну клетку. За ним вниз продвигается последовательность из 2 и 3. Когда 4 достигает границы экрана, последняя клетка переходит в состояние 2, а затем в 0, если клетка над ним находится в состоянии 2, или в 1, если верхняя клетка находится в состоянии 3. Таким образом снизу вверх состояния клеток преобразуются в 0 или 1 в зависимости от элементов последовательностью. Шаг за шагом на экране восстановится нужное изображение. Этот алгоритм назовем *Алгоритм 5*. На рисунке 6 приведен пример построения изображения.

В общем случае получаем оценку  $T(n, m, 5) \leq 2 \min(n, m) + 2$ , что и требовалось доказать.

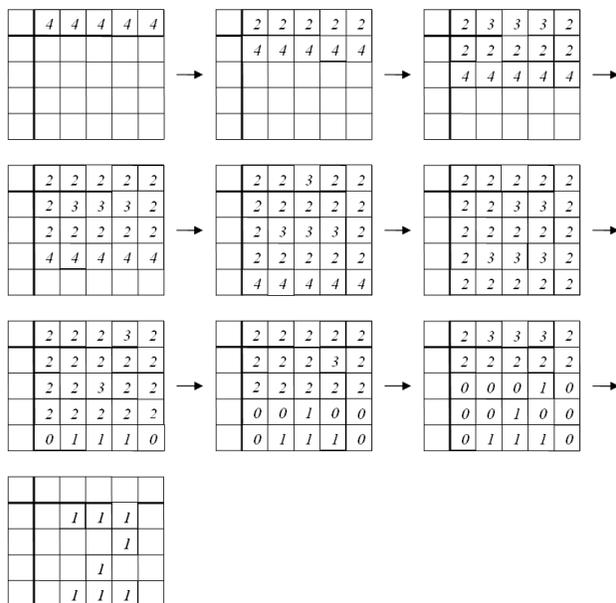


Рис. 6. Пример построения изображения по Алгоритму 5.

Из лемм 1, 2 и 3 следует утверждение теоремы 1.

#### 4. Оценка времени конструирования изображения при растущем числе состояний элементарного автомата

В этом разделе мы объединим все полученные результаты. В [3] была получена нижняя оценка времени построения изображения, она равна  $n$  (это кратчайшее время, за которое сигнал может достигнуть нижней угловой клетки). Также в этой работе приведен генератор с растущим числом состояний элементарного автомата, с помощью которого эта оценка достигается. Внешний автомат генерирует ненулевые выходы для свободных входов экрана за  $n$  тактов, начиная с  $(n + 1)$ -го такта выходные векторы нулевые и в  $(n + 1)$ -й такт на



Рис. 7. Зависимость времени конструирования изображений от числа состояний элементарного автомата.

экране появляется изображение, соответствующее заранее заданному коду. Будем называть этот алгоритм построения изображений *Алгоритм  $A_{min}$* , так как на нем достигается нижняя оценка времени конструирования изображений. В этом алгоритме элементарный автомат имеет  $2n + 2$  состояния (два из этих состояний элементарные автоматы никогда не принимают, но получают их на свободный вход в первый такт).

Итак, для  $(n, m)$ -экрана  $S$  (считаем, что  $n \leq m$ ) мы имеем оценки времени при  $Q(S)$  равном 3, 4, 5 и  $2n + 2$ . Рассмотрим случай, когда  $5 < Q(s) < 2n + 2$ . Пусть  $Q(S) \leq 5 + 2s$ , где  $1 \leq s \leq n - 2$ . В этом случае для построения изображений будем использовать алгоритм, который включает в себя два ранее описанных: *Алгоритм  $A_{min}$*  и *Алгоритм 5*. Нижние  $s$  строк будем строить по *Алгоритму  $A_{min}$* , то есть за минимально возможное время, а остальные  $n - s$  строк

построим с помощью *Алгоритма 5*. Скажем, что состояния 0, 1, 2, 3, 4, 5 соответствуют *Алгоритму 5* (назовем их состояниями первой группы), остальные состояния (состояния второй группы) — *Алгоритму  $A_{min}$* . Функция переходов состояний получается объединением равенств, задающих функции переходов для элементарных автоматов обоих алгоритмов. При этом легко дополнить эти уравнения такими уравнениями, чтобы состояния второй группы на входе автомата воспринимались им как 0 и 1, если сам автомат находится в одном из состояний первой группы. Таким образом получаем, что элементарный автомат имеет  $5 + 2s + 2$  состояния. Но состояния-метки из первой и второй группы можно отождествить друг с другом, то есть состояния 0 и 1 для обеих групп будут общими (уравнения, задающие функции переходов сохраняем, переименовывая состояния в 0 и 1 соответственно). Получаем, что элементарный автомат имеет  $5 + 2s$  состояний. Время построения изображений с помощью такого комбинированного алгоритма равно  $s + (2(n - s) + 1) + 1 = 2n - s + 2$ . То есть, с ростом числа состояний время построения изображений уменьшается, и эта зависимость линейна. Описанный комбинированный алгоритм будем называть *Алгоритм 6*.

На рисунке 7 изображен график, иллюстрирующий зависимость времени конструирования изображений от числа состояний элементарного автомата при  $3 \leq Q(s) \leq 2n + 2$ , то есть начиная с минимального числа состояний элементарного автомата и заканчивая  $(2n + 2)$  состояниями, когда время построения изображения достигает своей нижней оценки.

## 5. Верхняя оценка времени построения изображения на экране с одним входом при ограниченном числе состояний элементарного автомата

Переформулируем теперь задачу для экрана, который имеет единственный свободный вход — верхний вход элементарного автомата, стоящего в первой строке и первом столбце экрана. Другие неопреде-

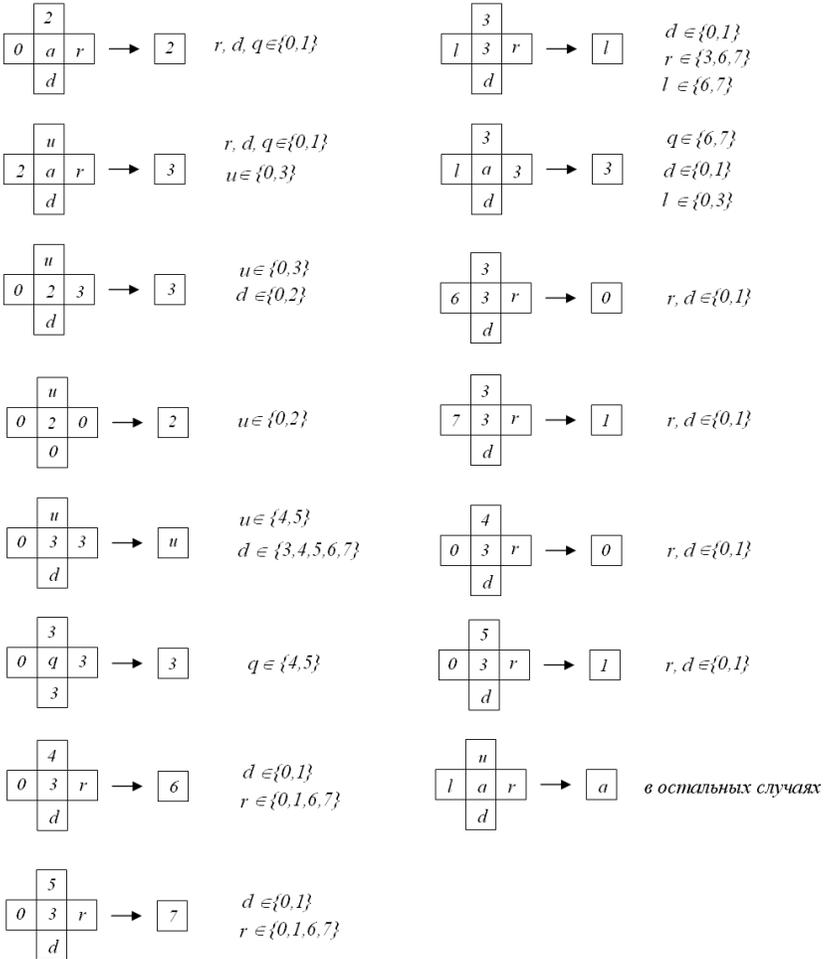


Рис. 8. Переходы состояний элементарного автомата с восемью состояниями.

ленные входы граничных автоматов доопределим нулями. Внешний автомат теперь имеет один выход. Таким образом, задача состоит в поиске для любого наперед заданного изображения такого генерато-

ра, который строит это изображение на экране с одним свободным входом.

Чтобы получить внешний и элементарный автоматы для этой задачи, модифицируем *Алгоритм 5*. В этом алгоритме мы использовали специальное состояние 4 элементарного автомата, которое распространялось на весь экран в направлении сверху вниз и по нему проходили сигналы, которые кодировали 0 или 1. Это вспомогательное состояние помогало распознавать границу экрана, то есть тот момент, когда коды должны расшифровываться в 0 или 1 соответственно. Воспользуемся этой же идеей. Но теперь, чтобы перевести все клетки экрана в одно вспомогательное состояние, будем использовать 2 вида сигнала — 2 и 3, причем 2 распространяется по вертикали вниз и одновременно порождает сигнал 3, который распространяется по горизонтали слева направо. Таким образом можно перевести все клетки экрана в состояние 3. Далее, будем зашифровывать изображение, используя состояния 4 и 5, соответствующие 0 и 1 в коде изображения. Эти сигналы распространяются сверху вниз по клеткам, находящимся в состоянии 3, пока не встречаются границу экрана, либо уже построенную часть изображения. В этот момент состояния 4 и 5 переходят в состояния 6 и 7 соответственно. Эти два сигнала передаются горизонтально слева направо, пока не дойдут до границы экрана. Как и в *Алгоритме 5*, дойдя до границы, эти состояния расшифровываются в 0 и 1 соответственно. Таким образом, элементарный автомат имеет 8 состояний. Указанный алгоритм будем называть *Алгоритм 7*. Более строго он описан в доказательстве теоремы 2.

### Доказательство теоремы 2. Алгоритм 7.

Без ограничения общности будем считать, что  $m \geq n$ .

Опишем элементарный автомат. По условию он имеет 8 состояний, то есть множество состояний —  $E_q = \{0, 1, 2, 3, 4, 5, 6, 7\}$ . Функцию переходов состояний зададим следующим образом:

$$\varphi(q, 0, r, 2, d) = 2 \text{ для любых } r, q, d \in \{0, 1\};$$

$$\varphi(q, 2, r, u, d) = 3 \text{ для любых } r, q, d \in \{0, 1\}, u \in \{0, 3\};$$

$$\varphi(2, 0, 3, u, d) = 3 \text{ для любых } u \in \{0, 3\}, d \in \{0, 2\};$$

$$\varphi(2, 0, 0, u, 0) = 2 \text{ для любых } u \in \{0, 2\};$$

$$\varphi(3, 0, 3, u, d) = u \text{ для любых } u \in \{4, 5\}, d \in \{3, 4, 5, 6, 7\};$$

$$\varphi(q, 0, 3, 3, 3) = 3 \text{ для любых } q \in \{4, 5\};$$

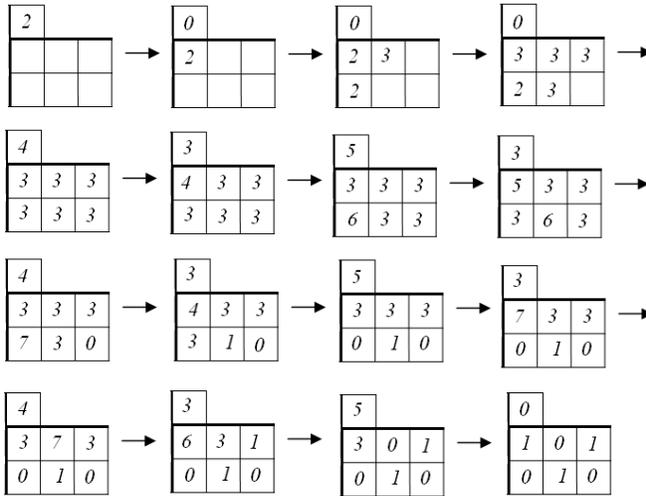


Рис. 9. Пример построения изображения по Алгоритму 7.

- $\varphi(3, 0, r, 4, d) = 6$  для любых  $d \in \{0, 1\}$ ,  $r \in \{0, 1, 6, 7\}$ ;
- $\varphi(3, 0, r, 5, d) = 7$  для любых  $d \in \{0, 1\}$ ,  $r \in \{0, 1, 6, 7\}$ ;
- $\varphi(3, l, r, 3, d) = l$  для любых  $d \in \{0, 1\}$ ,  $r \in \{3, 6, 7\}$ ,  $l \in \{6, 7\}$ ;
- $\varphi(q, l, 3, 3, d) = 3$  для любых  $q \in \{6, 7\}$ ,  $d \in \{0, 1\}$ ,  $l \in \{0, 3\}$ ;
- $\varphi(3, 6, r, 3, d) = 0$  для любых  $r, d \in \{0, 1\}$ ;
- $\varphi(3, 7, r, 3, d) = 1$  для любых  $r, d \in \{0, 1\}$ ;
- $\varphi(3, 0, r, 4, d) = 0$  для любых  $r, d \in \{0, 1\}$ ;
- $\varphi(3, 0, r, 5, d) = 1$  для любых  $r, d \in \{0, 1\}$ ;
- $\varphi(q, l, r, u, d) = q$  в остальных случаях.

На рисунке 8 изображены переходы состояний элементарного автомата.

Опишем внешний автомат, который соответствует изображению с кодом  $K$ . Он имеет единственный выход  $a$ . В первые 4 такта выход будет одинаковым для любого изображения, а именно:

- в первый такт  $a = 2$ ,
- при  $k \in \{2, 3, 4\}$  в  $k$ -й такт  $a = 0$ .

При  $5 \leq k < 4 + 2mn$  выход строится по следующим правилам:

при  $k = 4 + 2ms + (2t - 1)$ , где  $0 \leq s \leq n - 1$ ,  $1 \leq t \leq m$ , в  $k$ -й такт  $a = 4$ , если в коде  $K$  изображения в клетке с координатами  $(n - s, m - (t - 1))$  стоит 0,

$a = 5$ , если в коде  $K$  изображения в клетке с координатами  $(n - s, m - (t - 1))$  стоит 1,

при  $k = 4 + 2ms + 2t$ ,  $0 \leq s \leq n - 1$ ,  $1 \leq t \leq m$  в  $k$ -й такт  $a = 3$ ,

при  $k \geq 4 + 2mn$  в  $k$ -й такт  $a = 0$ .

При подаче выходов такого внешнего автомата на свободный вход экрана, в следующий такт после подачи последней ненулевой буквы на экране появится изображение, соответствующее заранее заданному коду. Таким образом время построения изображения равно  $2mn + 4$  и теорема 2 доказана.

На рисунке 9 приведен пример построения изображения по *Алгоритму 7*.

## Список литературы

- [1] Кудрявцев В. Б., Подколзин А. С., Болотов А. А. Основы теории однородных структур. М.: Наука, 1990.
- [2] Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. М.: Наука, 1985.
- [3] Титова Е. Е. Конструирование изображений клеточными автоматами // Интеллектуальные системы. Т. 12, вып. 1–4. М., 2008. С. 105–121.