

ФГБОУ ВО "Московский государственный университет имени М.В. Ломоносова"

Механико-математический факультет

Кафедра математической теории интеллектуальных систем

На правах рукописи

ТИТОВА ЕЛЕНА ЕВГЕНЬЕВНА

КОНСТРУИРОВАНИЕ ИЗОБРАЖЕНИЙ КЛЕТОЧНЫМИ
АВТОМАТАМИ

Специальность 01.01.09 — дискретная математика и
математическая кибернетика

Диссертация

на соискание ученой степени

кандидата физико-математических наук

Научный руководитель:

доктор физико-математических наук,

профессор Э.Э.Гасанов

Москва 2015

Оглавление

Введение	3
1 Общая характеристика работы	3
2 Краткое содержание работы	8
1 Конструирование изображений клеточными автоматами	26
1 Точная оценка числа состояний элементарного автомата	26
2 Точное значение времени построения изображения при неограниченном числе состояний элементарного автомата	33
3 Линейная оценка времени построения изображения при ограниченном числе состояний элементарного автомата	36
4 Оценка времени конструирования изображения при растущем числе состояний элементарного автомата	39
5 Верхняя оценка времени построения изображения на экране с одним входом при ограниченном числе состояний элементарного автомата	41
6 Построение изображений на многомерных экранах	46
2 Сложность управляющего автомата для построения изображений на универсальном экране	48
1 Предобработка кода изображения	48
2 Сложность управляющего автомата	50
3 Конструирование движущихся изображений клеточными автоматами	57
1 Движение точки на конечном экране	57
2 Движение многоточечных изображений на конечном экране	62
3 Движущиеся изображения на бесконечном экране	69
4 Движение с ограниченной скоростью на бесконечном экране	73
5 Автономно движущиеся изображения на бесконечном экране	76

Введение

1 Общая характеристика работы

Актуальность темы и степень ее разработанности

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 протекающими 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 автома-

тов, 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 [6], Майхиллом и другими [7] [10]. На механико-математическом факультете МГУ имени М. В. Ломоносова исследованием свойств однородных структур занимались В. Б. Кудрявцев [2] [3] [4], А. С. Подколзин [8] [9], А. А. Болотов [1]; результатом их работы также стала монография [5].

В настоящей работе рассматривается клеточный автомат, представляющий собой однородную структуру, заданную на конечном прямоугольнике, причем состояние структуры управляется входами, расположенными по верхней и левой границам 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. Далее решается задача для экрана с од-

ним управляющим входом. Наконец, рассматривается задача построения движущихся изображений на прямоугольном экране, а затем на бесконечной в одну сторону полосе.

Цель работы

Цель работы состоит в построении экранов, на которых возможно сконструировать любое стационарное изображение, а также в оценке сложности экранов и времени построения изображений. Для движущихся изображений цель состоит в нахождении классов законов движения, которые можно реализовать на одномерных экранах.

Методы исследования

В работе используются методы дискретной математики, теории автоматов и математического анализа.

Научная новизна

В работе получены оценки различных характеристик универсальных экранов (экранов, на которых можно построить любое наперед заданное изображение). А именно, получены оценки числа состояний элементарного автомата, время построения изображений при ограниченном и неограниченном числе состояний элементарного автомата. Построен ряд универсальных экранов, удовлетворяющих полученным оценкам.

Особенность построенных алгоритмов состоит в том, что они реализуемы для экранов любого размера, то есть, один и тот же алгоритм может быть модифицирован для экрана любой длины и ширины. Построенные алгоритмы являются универсальными не только в смысле произвольного выбора изображения, которое может быть построено, но и универсальны в смысле размеров экрана.

Также получены оценки сложности управляющего автомата для построенных алгоритмов.

В работе получены оценки аналогичных характеристик для экранов, на которых реализуются движущие изображения. Универсальность тут рассматривается в смысле реализации на экране произвольного закона движения заданного изображения. Установлено, что существует универсальный конечный экран, и не существует универсального бесконечного экрана.

Практическая ценность

Результаты работы могут быть использованы при создании экранов для наружной рекламы. Преимущество таких экранов в том, что экран состоит из множества просто организованных, а потому дешевых элементов. Кроме того, не нужно подводить вход к каждому элементу, что также упрощает физическую реализацию экрана. В то же время можно достичь высокого разрешения изображения. В работе описываются экраны и алгоритмы построения черно-белых изображений, но с помощью этой модели легко реализовать и цветные изображения.

Наличие большого числа выходов и малого числа входов экрана является типичной для чипов. Поэтому все упомянутые алгоритмы построения изображений могут применяться в перепрограммировании перепрограммируемых чипов (ПЛИС или FPGA).

Апробация работы

Результаты работы докладывались и публиковались в тезисах следующих конференций.

Конференция студентов, аспирантов и молодых ученых мех-мат. ф-та МГУ (2008, Москва).

Международная конференция «Современные проблемы математики, меха-

ники и их приложений», посвященная 70-летию академика В.А.Садовниченко (2009, Москва).

Международная конференция студентов, аспирантов и молодых ученых «Ломоносов» (2009, 2011, 2012, 2013, 2014, Москва).

I Международная научно-практическая конференция «Интеллектуальные машины» (2009, Москва).

X Международный семинар «Дискретная математика и ее приложения» (2010, Москва).

X Международная конференция «Интеллектуальные системы и компьютерные науки» (Москва, 2011).

XI Международный семинар «Дискретная математика и ее приложения», посвященный 80-летию со дня рождения академика О.Б.Лупанова (2012, Москва).

Также результаты докладывались на семинарах механико-математического факультета МГУ имени М. В. Ломоносова: на семинаре «Теория автоматов» под руководством академика, профессора, д.ф.м.н. В. Б. Кудрявцева (2011, 2014 гг.), на семинаре «Вопросы сложности алгоритмов поиска» под руководством профессора, д.ф.м.н. Э.Э.Гасанова (2007-2014 гг.).

Публикации

По теме диссертации опубликовано 14 печатных работ, из них 4 работы в журналах ВАК. Работ, написанных в соавторстве нет.

Структура и объем работы

Диссертация состоит из введения, трех глав и заключения. Объем диссертации 92 страницы. Список литературы содержит 24 наименования.

Благодарность

Автор выражает искреннюю благодарность научному руководителю д.ф.-м.н., профессору Э. Э. Гасанову за постановку задачи и внимание к работе. Также хочется высказать признательность заведующему кафедрой академику, профессору В. Б. Кудрявцеву и всему коллективу кафедры за опыт и знания, полученные во время обучения, за доброжелательную и творческую атмосферу.

2 Краткое содержание работы

В первой главе рассматривается задача конструирования изображений клеточными автоматами, приводится нижняя оценка числа состояний элементарного автомата универсального экрана, оценки времени построения изображений, приводятся алгоритмы построения изображений на универсальных экранах.

Будем использовать следующие обозначения:

Для $q \in \mathbb{N}$ будем обозначать $E_q = \{0, 1, \dots, q - 1\}$.

Конечный инициальный автомат — это шестерка

$$\mathcal{A} = (A, Q, B, \varphi, \psi, q_0),$$

где A — входной алфавит, Q — множество состояний, которое является конечным подмножеством некоторого фиксированного счётного множества, B — выходной алфавит, $\varphi : Q \times A \rightarrow Q$ — функция переходов, $\psi : Q \times A \rightarrow B$ — функция выходов, q_0 — начальное состояние.

Элементарным автоматом будем называть конечный инициальный автомат \mathcal{A} с четырьмя входами и одним выходом, такой что:

$$A = E_q^4, B = Q = E_q \text{ для некоторого } q \in \mathbb{N} \setminus \{1\}, \psi = \varphi = \varphi(q, l, r, u, d),$$

причем имеет место свойство $\varphi(0, 0, 0, 0, 0) = 0$.

(n, m) -экраном S будем называть следующую конструкцию.

Пусть имеется прямоугольник размера $n \times m$, $m, n \in \mathbb{N}$. В каждую клетку прямоугольника поместим по одному экземпляру одного и того же элементар-

ного автомата \mathcal{A} . К входам этого автомата присоединим выходы автоматов, стоящих в четырех соседних с ним клетках, то есть у автомата имеется *левый* вход l , *правый* вход r , *верхний* вход u и *нижний* вход d соответственно и q — текущее состояние автомата. Выходом автомата в заданный момент времени 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 автоматами*. 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* или *управляющими входами*. То есть (n, m) -экран S — это тройка $S = \langle \mathcal{A}, n, m \rangle$, где \mathcal{A} — элементарный автомат, $n, m \in \mathbb{N}$ — соответственно количество строк и столбцов прямоугольника.

Состояния 0 и 1 элементарного автомата будем называть *метками*.

Черно-белой конфигурацией будем называть такую конфигурацию состояний элементарных автоматов экрана, при которой каждый элементарный автомат экрана находится в нулевом или единичном состоянии.

Изображение — это черно-белая конфигурация, которую можно удерживать на экране сколь угодно долго, подавая на все свободные входы элементарных автоматов нулевые значения.

Кодом K изображения назовем матрицу $n \times m$, $n, m \in \mathbb{N}$, состоящую из нулей и единиц. Множество всех кодов изображений размера $n \times m$, $n, m \in \mathbb{N}$ будем обозначать через $\mathcal{K}(n, m)$. Изображение \mathfrak{F}_K на (n, m) -экране *соответствует данному коду* K , если положение нулей и единиц в изображении и в коде совпадают.

Внешний или *управляющий* автомат для экрана $\langle \mathcal{A}, n, m \rangle$ — это автономный инициальный автомат \mathcal{A}_e с множеством состояний $Q = E_{q'}$ для некоторого

$q' \in \mathbb{N}$, $q_0 = 0$, $B = E_q^{n+m}$. Это автомат, который генерирует последовательности входных символов из множества E_q для свободных входов экрана. Выход внешнего автомата — это слово $a_1 \dots a_n b_1 \dots b_m$, где $a_i, b_j \in E_q, i = 1, \dots, n, j = 1, \dots, m$, причем первые n букв будут подаваться на свободные входы по левой границе экрана, остальные m букв будут подаваться на свободные входы по верхней границе экрана. Далее для удобства будем считать, что внешний автомат на выходе выдает два слова $a_1 \dots a_n$ и $b_1 \dots b_m$.

Обозначим $\mathfrak{S}(n, m)$ — множество изображений размера $(n \times m)$, соответствующих всем возможным кодам размера $n \times m, n, m \in \mathbb{N}$.

Генератором G назовем пару $G = \langle \mathcal{A}_e, S \rangle$, где S — экран, \mathcal{A}_e — внешний автомат для экрана S .

Пусть на экране S находится некоторая черно-белая конфигурация. Скажем, что генератор $\langle \mathcal{A}_e, S \rangle$ формирует изображение с кодом K , если при подаче выходов \mathcal{A}_e на свободные входы экрана через некоторое время на экране появляется изображение, соответствующее коду K , которое после своего появления может удерживаться сколь угодно долго.

Если $n, m \in \mathbb{N}$, то 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{S}_K , соответствующее коду K .

Множество всех универсальных (n, m) -экранов обозначим через $\mathcal{U}(n, m)$.

Если $S = \langle \mathcal{A}, n, m \rangle$ — экран, то через $Q(S)$ обозначим число состояний элементарного автомата \mathcal{A} .

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

Теорема 1. *Если $n, m \in \mathbb{N}$ и*

если $\min(n, m) = 1$, то $Q(n, m) = 2$;

если $n, m \geq 2$, то $Q(n, m) = 3$.

Элементарный автомат \mathcal{A} называется *универсальным*, если для любых

$n, m \in \mathbb{N}$ экран $S = \langle \mathcal{A}, n, m \rangle$ является универсальным.

Теорема 2. *Не существует универсального элементарного автомата с двумя состояниями, и существует универсальный элементарный автомат с тремя состояниями.*

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{S}_K , соответствующее коду K , то $T(\mathcal{A}_e, S)$ — время формирования изображения \mathfrak{S}_K .

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

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

Теорема 3. *Если $n, m \in \mathbb{N}$, то*

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

Теорема 4. *Для любых $n, m \in \mathbb{N}$,*

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

$$T(n, m, 4) \leq 2 \min(n, m) + 1.$$

Обозначим через $\mathcal{P}(n, m)$ множество генераторов, соответствующих всем возможным кодам $K \in K(n, m)$. Это множество будем называть *универсальным генератором* $\mathcal{P}(n, m)$.

При доказательстве теоремы 4 получены универсальные генераторы $\mathcal{P}3(n, m)$ и $\mathcal{P}4(n, m)$, элементарные автоматы которых имеют 3 и 4 состояния соответственно.

Получены оценки времени построения изображений при растущем числе состояний элементарного автомата.

Теорема 5. *Если $n, m, s \in \mathbb{N}$, $m, n \geq 2$, $1 \leq s \leq n - 2$, то*

$$T(n, m, 4 + 2s) \leq 2 \min(n, m) - s + 2,$$

$$T(n, m, 2 \min(n, m) + 2) = \min(n, m) + 1.$$

При доказательстве теоремы 5 построен универсальный генератор $\mathcal{P}min(n, m)$ с минимальным временем построения изображений.

(n, m) -экран, на все свободные входы которого, кроме левого входа элементарного автомата, стоящего в первой строке и первом столбце, все время подается значение 0, будем называть (n, m) -экраном с одним свободным входом. То есть свободным входом данного экрана является только левый вход первого автомата самой верхней строки.

Обозначим через $T^1(n, m, q)$ минимальное время построения изображения на универсальном экране с одним свободным входом.

Для экрана с одним свободным входом получена оценка времени построения изображения.

Теорема 6. *Для любых $n, m \in \mathbb{N}$, то*

$$T^1(n, m, 8) \leq 2mn + 5.$$

При доказательстве теоремы 6 получен универсальный генератор $\mathcal{P}8(n, m)$, элементарный автомат которого имеет 8 состояний и экран имеет один свободный вход.

Рассмотрим параллелепипед $n_1 \times n_2 \times \dots \times n_k$ в k -мерном пространстве \mathbb{N}^k , где $k, n_1, n_2, \dots, n_k \in \mathbb{N}, k \geq 3, y_1, y_2, \dots, y_k$ — координатные оси в пространстве \mathbb{N}^k и будем считать, что одна из вершин параллелепипеда совпадает с началом координат O .

То есть, в данном случае экран — это параллелепипед в многомерном пространстве, разделенный $(k-1)$ -мерными гиперплоскостями на многомерные кубики с единичной длиной ребра. Эти кубики являются ячейками экрана. Для каждой ячейки экрана соседние с ней ячейки — те, которые имеют с ней общую грань (часть гиперплоскости, разделяющей данные ячейки). Для каждой ячейки пронумеруем соседние с ней ячейки в следующем порядке: номера $2i$ и $2i+1$ присвоим ячейкам, имеющим с данной общую грань, параллельную гиперплоскости, проходящей через оси $y_1, y_2, \dots, y_{i-1}, y_{i+1}, \dots, y_k$, то есть, через все координатные оси кроме i -й, причем из двух ячеек номер $2i$ присвоим ячейке, ближайшей к началу координат, а $2i+1$ — более отдаленной от начала координат.

Элементарный автомат \mathcal{A} в данном случае имеет $2k$ входов и один выход, для входного и выходного алфавитов и множества состояний выполняется $A = E_q^{2k}, B = Q = E_q$ для некоторого $q \in \mathbb{N} \setminus \{1\}$, функция выхода совпадает с функцией переходов состояний $\psi = \varphi = \varphi(q, x_1, x_2, \dots, x_k)$, причем имеет место свойство $\varphi(0, \dots, 0) = 0$.

Элементарный автомат помещается в каждую ячейку экрана. i -й вход автомата соединяется с выходом автомата, стоящего в соседней ячейке с номером i . Автоматы, стоящие в первом и n_i -м рядах ячеек параллелепипеда вдоль оси y_i , считая от начала координат, будем называть граничными автоматами. Для них определены не все входы.

Без ограничения общности будем считать, что $n_1 = \min(n_1, n_2, \dots, n_k), n_2 = \min(n_2, \dots, n_k)$.

Входы граничных автоматов, лежащие на грани, параллельной гиперплоскости $y_2 \dots y_k$, и проходящей через начало координат, будем называть *свободными* или *управляющими входами*. Также свободными назовем входы автоматов, лежащие на грани, параллельной гиперплоскости $y_1 y_3 \dots y_k$, и проходящей через начало координат. Остальные не определенные входы граничных автоматов определим нулями. Грани $y_2 \dots y_k$ и $y_1 y_3 \dots y_k$ будем далее называть верхней и левой гранями параллелепипеда соответственно.

То есть (n_1, n_2, \dots, n_k) -экран S — это совокупность элементарного автомата \mathcal{A} и k натуральных чисел — размеров экрана. Будем обозначать (n_1, n_2, \dots, n_k) -экран через $S = \langle \mathcal{A}, n_1, n_2, \dots, n_k \rangle$.

Как и ранее, состояния 0 и 1 элементарного автомата будем называть *метками*.

Черно-белой конфигурацией будем называть такую конфигурацию состояний элементарных автоматов экрана, при которой каждый элементарный автомат экрана находится в состоянии 0 или 1.

Изображением, как и ранее, будем называть такую черно-белую конфигурацию всех элементарных автоматов экрана, что при подаче нулей на свободные входы экрана эту конфигурацию можно удерживать на экране сколь угодно долго.

Кодом K изображения размера $n_1 \times \dots \times n_k$ назовем последовательность из $n_3 \times n_4 \times \dots \times n_k$ матриц размером $n_1 \times n_2$, $n_i \in \mathbb{N}$, состоящих из нулей и единиц. Множество всех кодов изображений размера $n_1 \times \dots \times n_k$, $n_i \in \mathbb{N}$ будем обозначать через $\mathcal{K}(n_1, \dots, n_k)$. Изображение \mathfrak{S}_K на (n_1, n_2, \dots, n_k) -экране *соответствует заданному коду* K , если положение нулей и единиц в изображении и в коде совпадают.

Внешний или *управляющий* автомат для экрана $\langle \mathcal{A}, n_1, n_2, \dots, n_k \rangle$ — это автономный инициальный автомат \mathcal{A}_e с множеством состояний $Q = E_{q'}$ для

некоторого $q' \in \mathbb{N}$, $q_0 = 0$, $B = E_q^{(n_2 n_3 \dots n_k + n_1 n_3 \dots n_k)}$. Это автомат, который генерирует последовательности входных символов из множества E_q для управляющих входов экрана.

Выход внешнего автомата — это последовательность из $n_3 \cdot \dots \cdot n_k$ слов $a_1^j \dots a_{n_1}^j b_1^j \dots b_{n_2}^j$ длины $n_1 + n_2$ где $a_i, b_l \in E_q, i = 1, \dots, n_1, l = 1, \dots, n_2, j = 1, \dots, n_3 \dots n_k - 1, n_3 \dots n_k$, причем первые n_1 букв каждого такого слова будут подаваться на управляющие входы на левой грани экрана, а остальные n_2 букв будут подаваться на управляющие входы на верхней грани экрана.

Обозначим $\mathfrak{F}(n_1, \dots, n_k)$ — множество изображений размера $n_1 \times \dots \times n_k$, соответствующих всем возможным кодам размера $n_1 \times \dots \times n_k$, $n_1, \dots, n_k \in \mathbb{N}$.

Генератором G назовем пару $G = \langle \mathcal{A}_e, S \rangle$, где S — экран, \mathcal{A}_e — внешний автомат для экрана S .

Пусть на экране S находится некоторая черно-белая конфигурация. Скажем, что генератор $\langle \mathcal{A}_e, S \rangle$ формирует изображение с кодом K , если при подаче выходов \mathcal{A}_e на свободные входы экрана через некоторое время на экране появляется изображение, соответствующее коду K .

Если $n_1, \dots, n_k \in \mathbb{N}$, то экран $S = \langle \mathcal{A}, n_1, \dots, n_k \rangle$ — универсальный, если для любого кода $K \in \mathcal{K}(n_1, \dots, n_k)$ 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_1, \dots, n_k) -экранов обозначим через $\mathcal{U}(n_1, \dots, n_k)$.

Если $S = \langle \mathcal{A}, n, m \rangle$ — экран, то через $Q(S)$ обозначим число состояний элементарного автомата \mathcal{A} .

Обозначим $Q(n_1, \dots, n_k) = \min_{S \in \mathcal{U}(n_1, \dots, n_k)} Q(S)$.

Для (n_1, n_2, \dots, n_k) -экрана S введем несколько обозначений по аналогии с данными ранее определениями. Если \mathfrak{F} — изображение, то через $\mathcal{G}(S, \mathfrak{F})$ обо-

значим множество генераторов $\langle \mathcal{A}_e, S \rangle$, формирующих изображение \mathfrak{S} . Обозначим

$$\begin{aligned} T(S, \mathfrak{S}) &= \min_{\langle \mathcal{A}_e, S \rangle \in \mathcal{G}(S, \mathfrak{S})} T(\mathcal{A}_e, S), \\ T(S, n_1, n_2, \dots, n_k) &= \max_{\mathfrak{S} \in \mathfrak{S}(n_1, n_2, \dots, n_k)} T(S, \mathfrak{S}), \\ T(n_1, n_2, \dots, n_k) &= \min_{S \in \mathcal{U}(n_1, n_2, \dots, n_k)} T(S, n_1, n_2, \dots, n_k), \\ T(n_1, n_2, \dots, n_k, q) &= \min_{S \in \mathcal{U}(n_1, n_2, \dots, n_k), Q(S \leq q)} T(S, n_1, n_2, \dots, n_k). \end{aligned}$$

Теорема 7. Если $n_1, n_2, \dots, n_k \in \mathbb{N}$, $n_1 = \min(n_1, n_2, \dots, n_k)$, $n_2 = \min(n_2, \dots, n_k)$, то

$$T(n_1, n_2, \dots, n_k, 3) \leq n_1 + n_2 + \min(n_1, n_2) + 1,$$

$$T(n_1, n_2, \dots, n_k, 4) \leq 2n_1 + 1.$$

Теорема 8. Если $n_1, n_2, \dots, n_k, s \in \mathbb{N}$, $n_1 = \min(n_2, n_3, \dots, n_k)$, $n_2 = \min(n_3, \dots, n_k)$, то $n_1, n_2 \geq 2$, $1 \leq s \leq n_1 - 2$, то

$$T(n_1, n_2, \dots, n_k, 4 + 2s) \leq 2n_1 - s + 2,$$

$$T(n_1, n_2, \dots, n_k, 2n_1 + 2) = n_1 + 1.$$

В доказательстве теоремы 7 используется разбиение экрана на экраны размера $n_1 \times n_2 \times 1 \times \dots \times 1$. Было выбрано именно такое разбиение с целью минимизировать время построения изображения по уже имеющимся алгоритмам. Вообще говоря, можно выбирать любые $i < j$ и рассматривать разбиение на экраны размера $1 \times \dots \times 1 \times n_i \times 1 \times \dots \times 1 \times n_j \times 1 \times \dots \times 1$, при этом изменяя расположение и количество свободных входов. Количество свободных входов при этом не увеличится.

Во второй главе внешний автомат рассматривается как автомат со входом. На его вход поступает код изображения — матрица из нулей и единиц, размер которой совпадает с размером экрана, позиции единиц в матрице соответствуют позициям точек в изображении, нули — отсутствию точки в этой позиции.

Автомат может обращаться к элементам матрицы в том порядке, в котором требует конкретный алгоритм. В некоторый момент после начала работы автомат начинает выдавать последовательности для свободных входов экрана.

Работа управляющего автомата условно разбивается на две части: предобработку кода изображения и непосредственно генерацию выходной последовательности. Предобработка осуществляется набором элементов, являющихся стандартными для для обработки потоков данных: это перестановка; устройство, которое собирает отдельные элементы в вектор из элементов; устройство, которое вставляет между элементами последовательности вспомогательные элементы; устройство, которое вставляет в начало последовательности заданное количество вспомогательных элементов. Будем называть эти элементы соответственно перестановка π , разреживатель с коэффициентом d_1 , разреживатель с коэффициентом d_2 , предобработчик G_a^k .

Примером предобработки является работа следующей конфигурации элементов. На вход перестановки π поступает код K изображения. π формирует из него некоторое слово, которое поступает на вход разреживателя R_{d_1} . R_{d_1} объединяет буквы входного слова в подслова по d_1 букв и подает эти подслова на вход разреживателю $R_{a_1}^{d_2}$, который между подсловами вставляет по d_2 элементов a_1 . Полученное слово поступает на вход предобработчика $G_{a_2}^k$, который в начало слова добавляет k букв a_2 . Полученная таким образом последовательность поступает на вход автомата A_e , который генерирует последовательности входных слов для свободных входов экрана.

Ниже даны строгие определения перечисленных устройств.

Перестановкой π для $n, m \in \mathbf{N}$ будем называть функцию

$$\pi : \mathcal{K}(n, m) \rightarrow E_2^r,$$

где $r \in \mathbf{N}$ — некоторое натуральное число.

Разреживателем R_d с коэффициентом $d \in \mathbf{N}$ для некоторого множества

V^{dk} , $k \in \mathbf{N}$ будем называть функцию

$$R_d : V^{dk} \rightarrow (V^d)^k,$$

такую что если $v = v_1 \dots v_{d \cdot k} \in V^{dk}$, то $R_d(v) = (v_1 \dots v_d) \dots (v_{d(k-1)+1} \dots v_{dk})$, то есть исходное слово v разбивается на подслова длины d .

Разреживателем R_a^d с коэффициентом $d \in \mathbf{N}$ и параметром $a \in V$ для некоторого множества $(V^l)^k$, $l, k \in \mathbf{N}$ будем называть функцию

$$R_a^d : (V^l)^k \rightarrow V^l \times \{a\}^{d-1} \times V^l \times \dots \times V^l \times \{a\}^{d-1} \times V^l,$$

(возможно $a \in V$), такую что если $v = v_1 \dots v_k \in (V^l)^k$, где $v_i \in V^l$, то

$$R_a^d(v) = v_1 a \dots a v_2 a \dots a \dots v_{k-1} a \dots a v_k,$$

где между каждыми l буквами слова v вставлено по $(d - 1)$ -й букве a .

Предобработчиком G_a^k , $k \in \mathbf{N} \cup \{0\}$ для некоторого множества V^s (возможно $a \in V$), $s \in \mathbf{N}$ будем называть функцию

$$G_a^k : V^s \rightarrow \{a\}^k \times V^s,$$

такую что если $v = v_1 \dots v_s \in V^s$, то $G_a^k(v) = a \dots a v_1 \dots v_s$, то есть буква a вставлена в начало исходного вектора k раз.

Поскольку все перечисленные устройства, кроме автомата \mathcal{A}_e , являются стандартными, считаем, что они имеются в наличии и могут нами использоваться. В этой ситуации интерес представляет сложность автомата \mathcal{A}_e , который непосредственно генерирует выходную последовательность и при этом не является стандартным устройством. Далее под сложностью управляющего автомата будем подразумевать сложность автомата \mathcal{A}_e .

Если $\mathcal{P}(n, m)$ — универсальный генератор, то множество всех универсальных генераторов, строящих такие же последовательности входных слов, что и $\mathcal{P}(n, m)$, обозначим через $\mathcal{G}(\mathcal{P}(n, m))$. Тогда если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P})$, то обозначим $Q(\mathcal{A}_e)$ — число состояний внешнего автомата \mathcal{A}_e .

Для универсальных генераторов, построенных в главе 1, имеют место следующие оценки.

Теорема 9. Если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P}_3(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то $\min_{G \in \mathcal{G}(\mathcal{P}_3(n, m))} Q(\mathcal{A}_e) = 2$.

Если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P}_4(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то $\min_{G \in \mathcal{G}(\mathcal{P}_4)} Q(\mathcal{A}_e) = 1$.

Если $G = \langle \mathcal{A}_e, S(n, m) \rangle = G \in \mathcal{G}(\mathcal{P}_{\min}(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то $\min_{G \in \mathcal{G}(\mathcal{P}_{\min})} Q(\mathcal{A}_e) = n$.

Если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P}_8(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то $\min_{G \in \mathcal{G}(\mathcal{P}_8)} Q(\mathcal{A}_e) = 1$.

В третьей главе изучаются движения изображений на экране. Рассмотрим экран, представляющий из себя конечную последовательность из m одинаковых элементарных автоматов \mathcal{A} с двумя входами. Входы автомата \mathcal{A} называются левым и правым и ими соответственно являются состояния левого и правого соседа. Правый вход последнего m -го автомата доопределяется как тождественный ноль, а левый вход первого автомата называется свободным и подключен к выходу управляющего автомата \mathcal{A}_e . При этом функцию переходов состояний автомата \mathcal{A} обозначим через $\varphi(l, q, r)$, где q — текущее состояние автомата, l, r — состояния его левого и правого соседей соответственно. Положим для любого экрана выполнено свойство $\varphi(0, 0, 0) = 0$. Через $S(\mathcal{A}, m)$ будем обозначать экран длины $m \in \mathbb{N}$ с элементарным автоматом \mathcal{A} . Через $Q(S)$ будем обозначать мощность множества состояний элементарного автомата экрана S . Тройку $G = \langle \mathcal{A}_e, \mathcal{A}, m \rangle$ будем называть *генератором*. Среди состояний элементарного автомата выделим непустое подмножество L , не содержащее элемент 0, и элементы этого множества будем называть *метками*.

Изображением будем называть слово в алфавите $\{0, 1\}$, начинающееся и заканчивающееся единицей. Длиной слова A будем называть количество букв в

слове A и обозначать $|A|$. Если изображение имеет длину 1, то будем называть его *точкой* и обозначать через I^1 , то есть I^1 — слово, состоящее из одной буквы 1. Для произвольного изображения I ненулевые буквы изображения будем называть *точками*. Также *точками* будем называть метки на экране.

Законом движения F для экрана $S(\mathcal{A}, m)$ будем называть слово в алфавите $\{0, 1\}$, которое заканчивается единицей и содержит ровно m единиц. Множество всех законов движения для экрана $S(\mathcal{A}, m)$ будем обозначать через \mathcal{F}_m .

Будем говорить, что *на экране $S(\mathcal{A}, m)$ реализуется движение изображения I^1 по закону F из \mathcal{F}_m* , если выполняются следующие условия:

- 1) в некоторый момент времени в самой левой ячейке экрана появляется метка (до этого на экране нет меток), этот момент будем называть *моментом начала движения* или *началом движения*;
- 2) изменение позиции метки на экране в i -й момент от начала движения ($1 < i \leq |F|$) соответствует i -й букве в слове F , а именно, если $F(i) = 0$, то в $(i + 1)$ -й момент метка остается в той же ячейке, где была в текущий момент, если $F(i) = 1$, то в $(i + 1)$ -й момент метка сдвинется на одну ячейку вправо, по сравнению со своим текущим положением;
- 3) в каждый момент времени после начала движения и до $|F|$ -го такта от начала движения (этот момент называем *моментом окончания движения*) на экране есть ровно одна метка.

Понятно, что поскольку в законе движения F ровно m единиц и F заканчивается на единицу, то после окончания движения на экране не будет ни одной метки.

Если $S(\mathcal{A}, m)$ — экран, I — изображение, содержащее l единиц, причем $1 = i_1 < i_2 < \dots < i_l = |I|$ — позиции единиц, считая слева направо и $F \in \mathcal{F}_{m+|I|-1}$ — закон движения, то будем говорить, что на экране S реализуется движение

изображения I по закону F , если выполнены следующие условия:

- 1) до некоторого момента t^* на экране нет ни одной метки, а в момент t^* появляется точка, которая далее движется по закону F , при этом когда мы говорим, что точка движется по закону F , предполагаем, что пункт 3) определения движения точки может не выполняться, поскольку на экране могут двигаться сразу несколько точек изображения (здесь t^* — *момент начала движения*, а появившаяся точка — это последняя точка изображения, то есть находящаяся в позиции $i_l = |I|$);
- 2) в момент $t^* + i_l - i_k$, $1 \leq k < l$ на экране появляется k -я точка изображения (то есть находящаяся в позиции i_k), которая далее движется по закону F начиная с буквы $F(i_l - i_k + 1)$;
- 3) в каждый момент времени после начала движения l -й точки изображения до окончания движения первой точки изображения на экране находится не более l меток.

Экран $S(\mathcal{A}, m)$ будем называть *универсальным для изображения I и множества законов движения \mathcal{F}* , если для любого закона движения F из \mathcal{F} существует такой управляющий автомат \mathcal{A}_e , что генератор $G = \langle \mathcal{A}_e, \mathcal{A}, m \rangle$ формирует на экране изображение I , движущееся закону F . Множество всех таких экранов будем обозначать через $\mathcal{U}(I, \mathcal{F}, m)$.

Экран $S(\mathcal{A}, m)$ будем называть *универсальным для изображения I* , если он является универсальным для изображения I и множества $\mathcal{F}_{m+|I|-1}$ всех законов движения для экрана S и изображения I .

Если $\mathcal{F} \subseteq \mathcal{F}_{m+|I|-1}$, то обозначим $Q(I, \mathcal{F}, m) = \min_{S \in \mathcal{U}(I, \mathcal{F}, m)} Q(S)$, $Q(I, m) = Q(I, \mathcal{F}_{m+|I|-1}, m)$.

Через $\mathcal{F}^{r,d}$ обозначим множество таких законов движения F , в которых начиная с момента $d+1$ не встречается более чем r единиц подряд, $r, d \in \mathbb{N} \cup \{0\}$.

То есть, для любого отрезка единиц с длиной больше r , правая граница этого отрезка имеет координату меньше $d + 1$.

Законы движения из $\mathcal{F}^{r,d}$ будем называть *законами движения с ограниченной скоростью с параметром r* , $r \in \mathbb{N}$.

$\mathcal{F}^{r,0}$ — множество законов движения, в которых не встречается более чем r единиц подряд.

Теорема 10. *Если $m \geq 2$, $m \in \mathbb{N}$, то $Q(I^1, \mathcal{F}^{1,0} \cap \mathcal{F}_m, m) = 4$.*

Теорема 11. *Если $m \in \mathbb{N}$ и $m \geq 2$, то $4 \leq Q(I^1, m) \leq 7$.*

Теорема допускает обобщение на изображения с произвольным количеством точек.

Теорема 12. *Для любого натурального k существует такой элементарный автомат A с 7^k состояниями, что для любого натурального m экран $S(A, m)$ является универсальным для любого изображения I , состоящего из не более чем k точек.*

Если на изображения наложить некоторые ограничения, то можно построить более дешевые универсальные экраны. Приведем несколько таких примеров.

Изображение вида $I_{seg}^1 = 1^k$ для натурального $k > 1$ будем называть отрезком длины k .

Теорема 13. *Если $k, m \geq 2$, $k, m \in \mathbb{N}$ и $I_{seg}^1 = 1^k$, то*

$$Q(I_{seg}^1, \mathcal{F}^{1,0} \cap \mathcal{F}_{m+k-1}, m) \leq 12.$$

Через $\mathcal{F}^{r,d,h}$ обозначим множество таких законов движения F , в которых начиная с момента $d + 1$ не встречается более чем r единиц подряд, и не встречается более чем h нулей подряд $r, d, h \in \mathbb{N}$. То есть, для любого отрезка единиц с длиной больше r и для любого отрезка нулей с длиной больше h правые границы этих отрезков имеют координаты меньше $d + 1$.

Обозначим через \mathcal{I}^k множество изображений, состоящих из k точек, позиции которых в изображении равны соответственно $1 = i_1 < i_2 < \dots < i_k$ и удовлетворяют условию $i_j - i_{j-1} \geq j + 1$ для любого j , $2 \leq j \leq k$.

Теорема 14. *Для любых натуральных $k, m \geq 2$ и любого изображения I из \mathcal{I}^k , такого, что $s = |I| + k \geq k^2/2 + 3k/2$, выполнено*

$$Q(I, (\mathcal{F}^{1,0} \setminus \mathcal{F}^{1,0,s-1}) \cap \mathcal{F}_{m+|I|-1}, m) \leq 9.$$

Обозначим через \mathcal{I}_{seg}^k множество изображений, состоящих из k отрезков, $k \in \mathbb{N}$ с координатами левых концов, $1 = l_1 < l_2 < \dots < l_k$, и с координатами правых концов, $1 \leq r_1 < r_2 < \dots < r_k$, таких, что $l_j - l_{j-1} \geq j + 1$ и $r_j - r_{j-1} \geq j + 1$ для любого j , $2 \leq j \leq k$.

Теорема 15. *Для любых натуральных $k, m \geq 2$ и любого изображения I из \mathcal{I}_{seg}^k , такого, что $s = |I| + k \geq k^2/2 + 3k/2$, выполнено*

$$Q(I, \mathcal{F}^{1,0} \setminus \mathcal{F}^{1,0,s-1} \cap \mathcal{F}_{m+|I|-1}, m) \leq 54.$$

Через $S(\mathcal{A})$ будем обозначать бесконечный в одну сторону (вправо) экран.

Законом движения для бесконечного экрана будем называть сверхслово $F \in \{0, 1\}^\infty$.

Переформулируем определение движения для случая бесконечного экрана. Будем говорить, что на экране $S(\mathcal{A})$ реализуется движение изображения I^1 по закону F , если выполняются следующие условия:

- 1) в некоторый момент времени в самой левой ячейке экрана появляется метка (до этого на экране нет меток), этот момент будем называть *моментом начала движения* или *началом движения*;
- 2) изменение позиции метки на экране в i -й момент от начала движения соответствует i -й букве в сверхслове F , а именно, если $F(i) = 0$, то в $(i + 1)$ -й

момент метка остается в той же ячейке, где была в текущий момент, если $F(i) = 1$, то в $(i + 1)$ -й момент метка сдвинется на одну ячейку вправо, по сравнению со своим текущим положением;

- 3) в каждый момент времени после начала движения на экране есть ровно одна метка.

Экран $S(\mathcal{A})$ будем называть *универсальным для изображения I^1 и множества законов движения \mathcal{F}* , если для любого F из \mathcal{F} существует такой управляющий автомат \mathcal{A}_e , что генератор $G = \langle \mathcal{A}_e, \mathcal{A}, \infty \rangle$ формирует на экране изображение I^1 , движущееся закону F . Множество всех таких экранов будем обозначать через $\mathcal{U}(\mathcal{F})$.

Если $\mathcal{F} \subseteq \{0, 1\}^\infty$, то обозначим $Q(\mathcal{F}) = \min_{S \in \mathcal{U}(\mathcal{F})} Q(S)$.

Теорема 16. *Для любого бесконечного экрана $S(\mathcal{A})$ существует такой закон движения $F \in \{0, 1\}^\infty$, что на экране $S(\mathcal{A})$ невозможно реализовать движение изображения I^1 по закону F .*

Теорема 17. *Существует закон движения $F \in \{0, 1\}^\infty$, движение по которому невозможно реализовать ни на каком экране $S(\mathcal{A})$.*

Как и ранее через $\mathcal{F}^{r,d}$ обозначим множество таких законов движения F (но теперь уже сверхслов), в которых начиная с момента $d + 1$ не встречается более чем r единиц подряд, $r, d \in \mathbb{N} \cup \{0\}$.

Теорема 18. *Если $s, d \in \mathbb{N}$, то имеют место следующие оценки*

$$3 \leq Q(\mathcal{F}^{s,0}) \leq 2s + 2;$$

$$3 \leq Q(\mathcal{F}^{s,d}) \leq 2s + 9.$$

Движение изображения на экране будем называть *автономным*, если начиная с некоторого момента $t \in \mathbb{N}$, $t < \infty$ на свободный вход экрана поступают только нулевые значения.

Обозначим $\nu_F(t)$ — количество единиц в префиксе длины t закона движения F .

Если для закона движения F существует предел $\lim_{t \rightarrow \infty} \frac{\nu_F(t)}{t}$, то будем называть его *скоростью движения точки по закону F* и обозначать через $v(F)$.

Теорема 19. *Для любого рационального числа a , $0 \leq a \leq 1$, существует непериодический закон F_a , такой что $v(F_a) = a$ и $\mathcal{U}(\{F_a\}) \neq \emptyset$.*

Скажем, что в законе движения F реализуется скорость $v \in \mathbb{R}$, если существует такая бесконечная возрастающая последовательность $\{t_i\}$, $t_i \in \mathbb{N}$, что $\lim_{i \rightarrow \infty} \frac{\nu_F(t_i)}{t_i} = v$.

Теорема 20. *Существует бесконечный экран $S(\mathcal{A})$, на котором реализуется такой закон движения F изображения I^1 , что в F реализуются все скорости v из отрезка $[0, 1]$.*

Следствие 1. *Существует автономный бесконечный экран, реализующий для изображения I^1 закон движения с растущим числом идущих подряд нулей и растущим числом идущих подряд единиц.*

Глава 1

Конструирование изображений клеточными автоматами

1 Точная оценка числа состояний элементарного автомата

В первой главе рассматривается задача конструирования изображений клеточными автоматами. В данном разделе получена нижняя оценка числа состояний элементарного автомата универсального экрана, то есть, такого экрана, на котором можно построить любое наперед заданное изображение.

Лемма 1. *Для любых $n, m \in \mathbb{N}$ имеем $Q(n, m) \geq 2$;*

если $m, n \geq 2$, то $Q(n, m) > 2$.

Доказательство. Первое утверждение леммы очевидно. Если элементарный автомат имеет только одно состояние, то конфигурация на экране никогда не изменится, и мы не сможем получить никакое другое изображение кроме изображения, состоящего из одних нулей. Таким образом, чтобы на экране можно было получать различные изображения, клеточный автомат должен иметь не менее двух состояний, то есть $Q(S) \geq 2$.

Второе утверждение леммы будем доказывать от противного. Пусть $n, m \geq$

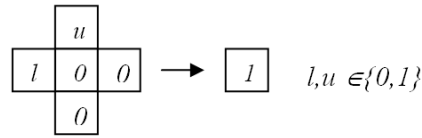


Рис. 1.1: Конфигурация входов элементарного автомата, переводящая состояние 0 в состояние 1.

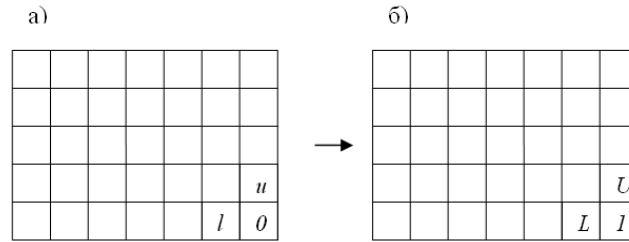


Рис. 1.2: Пример изображения, которое сохраняется только один такт.

2, $S = \langle \mathcal{A}, n, m \rangle$ — универсальный (n, m) -экран и пусть $Q(S) = 2$, то есть элементарный автомат имеет два состояния из множества $E_2 = \{0, 1\}$. Так как экран универсальный, то для любого изображения существует внешний автомат \mathcal{A}_e такой, что генератор $\mathcal{G} = \langle \mathcal{A}_e, S \rangle$ формирует это изображение. То есть это условие выполнено и для изображения, содержащего одновременно и нулевые и единичные состояния клеточных автоматов. Значит существует конфигурация, переводящая состояние 0 элементарного автомата в состояние 1 на экране с начальной конфигурацией экрана только из нулевых состояний клеток. При этом ненулевыми могут быть лишь левый и верхний входы клетки, то есть существуют такие $l, u \in E_2$, что $\varphi(0, l, 0, u, 0) = 1$, причем хотя бы одна из l и u не равна нулю (см. рисунок 1.1). Но тогда мы не сможем получить изображение, приведенное на рисунке 1.2 а) (считаем, что в пустых клетках стоят нули), так как оно сохраняется только один такт и переходит в конфигурацию, изображенную на рисунке 1.2 б). Это противоречит определению универсального экрана. Лемма доказана. ■

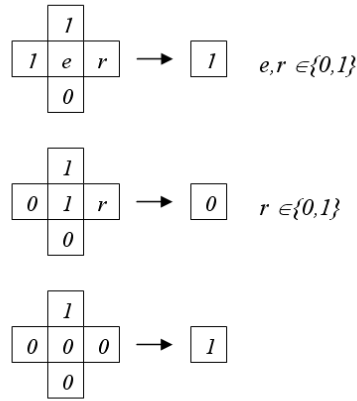


Рис. 1.3: Переходы состояний элементарного автомата с двумя состояниями.

Лемма 2. *Существует универсальный (n, m) -экрэн $S = \langle \mathcal{A}, n, m \rangle$, такой что:*

если $\min(n, m) = 1$, то $Q(S) = 2$;

если $m, n \geq 2$, то $Q(S) = 3$.

Доказательство. Рассмотрим сначала случай, когда $\min(n, m) = 1$. Пусть для определенности $n = 1, m \geq 1$ (в противном случае необходимо переименовать входы автомата, описанного ниже, следующим образом: l в u , u в l и r в d , d в r). Приведем пример универсального экрана, элементарный автомат которого имеет два состояния.

Имеем множество состояний $E_q = \{0, 1\}$. Опишем функцию переходов состояний:

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

$$\varphi(1, 0, r, 1, 0) = 0 \text{ для любого } r \in 0, 1;$$

$$\varphi(0, 0, 0, 1, 0) = 1;$$

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

На рисунке 1.3 показано, как меняются состояния элементарного автомата в зависимости от конфигурации входных элементов.

Построим внешний автомат для заданного двоичного кода изображения K .

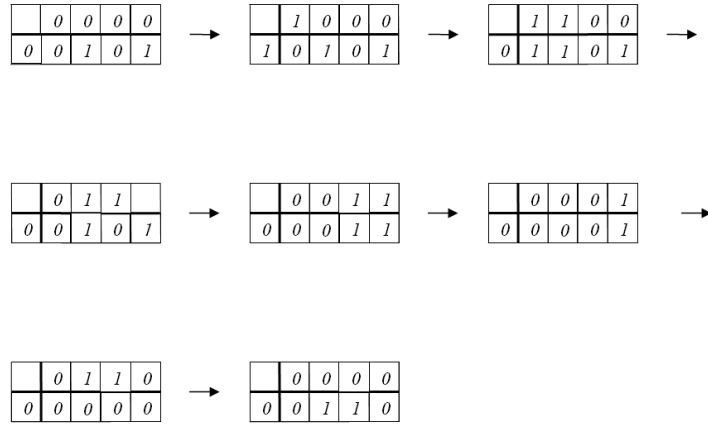


Рис. 1.4: Пример построения изображения на (1,4)-экране по алгоритму $\mathcal{P}2$.

Сначала внешний автомат подает последовательности входных слов, чтобы очистить экран, то есть перевести все клетки экрана в нулевое состояние, так как в начальном состоянии на экране могла находиться любая конфигурация из нулей и единиц. Для этого для каждой клетки экрана поочередно слева направо значение левого и верхнего входов переводятся в состояние 1. Функция переходов задана так, что после этого состояние самой клетки тоже становится равным единице. К этому моменту на верхний вход этой клетки подается значение 1, а на левый — 0, и в следующий такт значение клетки экрана становится равным нулю. Так последовательно обнуляются значения всех клеток экрана.

Затем за один такт внешний автомат подает сверху слово, равное коду изображения, которое необходимо построить, и на экране формируется изображение, соответствующее этому коду.

Итак, первый такт внешний автомат подает слова 1 и 10^{m-1} ; во второй такт — слова 0 и $1^2 0^{m-2}$; в третий — слова 0 и $(01^2 0^{m-3})_m$ и так далее в момент k , ($k \leq m$) он подает слова 0 и $0^{k-1} 1^2 0^{m-k-1}$, в $(m+1)$ -й такт — слова 0 и $0^{m-1} 1$. В следующий такт последняя клетка экрана переходит в нулевое состояние и автомат подает слово 0 и слово $a_1 a_2 \dots a_m$, совпадающее с кодом изображения. В следующий такт все автоматы принимают нужные состояния и на экране

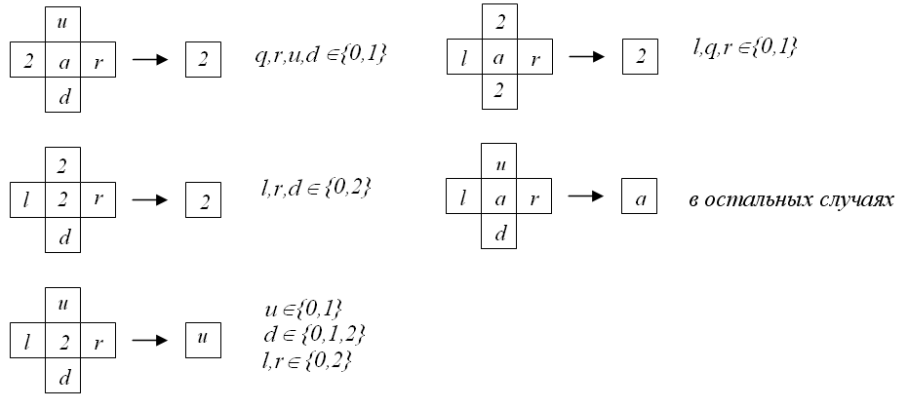


Рис. 1.5: Переходы состояний элементарного автомата с тремя состояниями.

появляется заданное изображение.

Назовем построенный алгоритм $\mathcal{P}2$.

На рисунке 1.4 приведен пример конструирования изображения на $(1, 4)$ -экране.

Пусть теперь $m, n \geq 2$, приведем универсальный (n, m) -экран, элементарный автомат которого имеет три состояния. Без ограничения общности будем считать, что $m \geq n$.

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

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

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

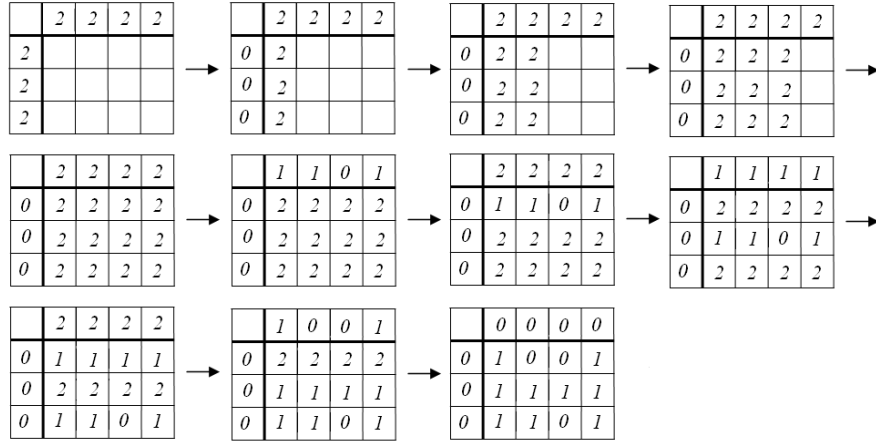
$$\varphi(2, l, r, u, d) = u \text{ для любых } u \in \{0, 1\}, d \in \{0, 1, 2\}, l, r \in \{0, 2\};$$

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

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

На рисунке 1.5 изображены переходы состояний элементарного автомата.

Опишем работу внешнего автомата, который соответствует изображению с заданным кодом K . Построим сначала первое выходное слово. В первый такт оно равно 2^n , во второй и все последующие такты первое выходное слово равно

Рис. 1.6: Пример работы универсального генератора $\mathcal{P}3(n, m)$.

0^n .

Второе выходное слово $b_1 \dots b_m$, подаваемое на верхнюю границу экрана, будем строить по следующим правилам:

при $1 \leq i \leq m + 1$ в i -й такт это слово равно 2^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 выходное слово равно $(m - s)$ -й строке кода K ;

если $i = (m + 2) + 2s + 1$, $0 \leq s \leq n - 1$, то в i -й такт выходное слово равно 2^m ;

при $i \geq m + 2n + 1$ выходное слово равно 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 не содержит двоек, то строка кода останавливается и дальше не двигается. Таким образом, на экране снизу вверх по строкам восстанавливается изображение, соответствующее заданному коду. Оно появ-

ляется на экране в следующий такт после подачи на свободные входы экрана последнего ненулевого слова, то есть на $(m + 2n + 1)$ -й такт. Итак, получена оценка $T(n, m, 3) \leq m + n + \min(n, m) + 1$, что и требовалось доказать ■

Описанный универсальный генератор будем называть $\mathcal{PZ}(n, m)$.

Из лемм 1 и 2 следует

Теорема 1. *Если $n, m \in \mathbb{N}$ и*

если $\min(n, m) = 1$, то $Q(n, m) = 2$;

если $n, m \geq 2$, то $Q(n, m) = 3$.

Теорема 2. *Не существует универсального элементарного автомата с двумя состояниями, и существует универсальный элементарный автомат с тремя состояниями.*

Доказательство. Очевидно, что элементарный автомат построенного в лемме 2 универсального экрана является универсальным элементарным автоматом.

Из леммы 1 ясно, что не существует универсального элементарного автомата с двумя состояниями. ■

Для описанных алгоритмов легко получаются оценки времени построения изображений.

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

$$T(1, m, 2) = T(m, 1, 2) \leq m + 3,$$

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

На рисунке 1.6 приведен пример построения изображения по этому алгоритму.

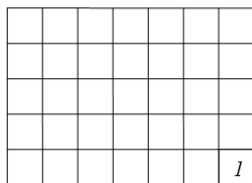


Рис. 1.7: Пример изображения, которое невозможно сконструировать быстрее чем за $\min(n, m) + 1$ тактов.

Результаты данного раздела анонсированы в [15] [16] [17] [18] и опубликованы в [12]

2 Точное значение времени построения изображения при неограниченном числе состояний элементарного автомата

В данном разделе получена оценка наименьшего времени построения изображения на универсальном экране с неограниченным числом состояний элементарного автомата. Полученная оценка не может быть улучшена.

Лемма 3. *Если $n, m \in \mathbb{N}$, то*

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

Доказательство. Приведем пример изображения, которое нельзя сконструировать быстрее чем за $\min(n, m) + 1$ тактов. Это изображение изображено на рисунке 1.7. Оно состоит из одной единственной точки с координатами (n, m) . Допустим, что мы уже построили эту точку. Это означает, что был такой момент, когда не все входы у нее были нулевыми. А значит, еще раньше был момент, когда не все входы у автоматов из ее окрестности были нулевыми и так далее. Так как каждый автомат видит только состояния соседних с ним автоматов, то за один такт ненулевое состояние распространяется не более чем на

одну клетку в каждом направлении. Это означает, что если ненулевое значение появилось на каком-то из свободных входов экрана, то клетки с координатами (n, m) оно достигнет не раньше чем в $\min(n, m) + 1$ -й такт. ■

Лемма 4. Если $n, m \in \mathbb{N}$, то

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

Доказательство. Построим генератор, который конструирует любое изображение за время $\min(n, m) + 1$. Без ограничения общности считаем, что $m \geq n$.

Построим элементарный автомат. Будем считать, что состояние элементарного автомата состоит из двух компонент: первая — счетчик (от 0 до n), а вторая — метка (0 или 1). В начальный момент времени счетчик автомата и метка равны нулю. Существенным входом элементарного автомата является только состояние верхнего автомата. Если значение счетчика автомата равно нулю, автомат выдает метку. В каждый момент времени автомат берет значение счетчика и метку у верхнего автомата. Если значение счетчика больше нуля, автомат уменьшает его на единицу, присваивает это значение своему счетчику и берет у верхнего автомата новое значение метки. Таким образом определяется новое состояние элементарного автомата. Если значение счетчика верхнего автомата равно нулю, автомат не меняет значение своего счетчика и своей метки. Таким образом, элементарные автоматы имеют по $2n + 2$ состояния, $A = B = Q = E_{2n+2} = \{0, 1, \dots, 2n + 1\}$. Для состояний 0 и 1 значения счетчика равны нулю, значения метки равны 0 и 1 соответственно. Для состояний с 2-го по $(n + 1)$ -е значение метки равно 0 и значения счетчика равно $1, 2, \dots, n$ соответственно. Для состояний с $(n + 2)$ -го по $(2n + 1)$ -е значение метки равно 1 и значения счетчика равно $1, 2, \dots, n$ соответственно. Переходы состояний описаны выше.

Построим внешний автомат, соответствующий заданному двоичному коду K . В начальном (нулевом) состоянии автомат подает на верхнюю горизонталь n -ю

строку кода изображения и значение счетчика n . В следующий такт — $(n - 1)$ -ю строку изображения и значение счетчика $(n - 1)$ и так далее, в n -й такт подает первую строку и значение счетчика 1, в $(n + 1)$ -й такт возвращается в начальное состояние. То есть в i -м состоянии автомат подает $(n - i)$ -ю строку и значение счетчика $(n - i)$. На свободные входы экрана с левой границы автомат в каждый такт подает нулевое слово.

Поясним этот алгоритм. Внешний автомат подает последовательности из кода изображения с соответствующими счетчиками с верхней границы в соответствующие столбцы экрана (на левую границу всегда подаются нули). Сначала подает метку самой нижней клетки столбца и значение счетчика n , затем метку $(n - 1)$ -й клетки столбца и значение счетчика $(n - 1)$ и так далее. За один такт сигнал продвигается на одну клетку вниз и значение счетчика уменьшается на единицу. Поэтому, когда сигнал достигнет нижней строки, счетчик обнулится и автомат выдаст нужную метку. Аналогично со всеми остальными клетками столбца. Поскольку значения счетчиков подавались каждый раз на единицу меньше, то все счетчики обнулятся одновременно через n тактов после подачи первого входного вектора. То есть, на $n + 1$ -м такте на экране построится изображение, соответствующее заданному двоичному коду. Значит, $T(n, m) \leq n + 1$ или в общем случае $T(n, m) \leq \min(n, m) + 1$, что и требовалось доказать ■

Алгоритм построения изображений, построенный в лемме 4 будем называть *Рmin*.

Из лемм 3 и 4 следует теорема.

Теорема 3. Если $n, m \in \mathbb{N}$, то

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

Результаты этого параграфа анонсированы в [15] [16] [17] и опубликованы в [11].

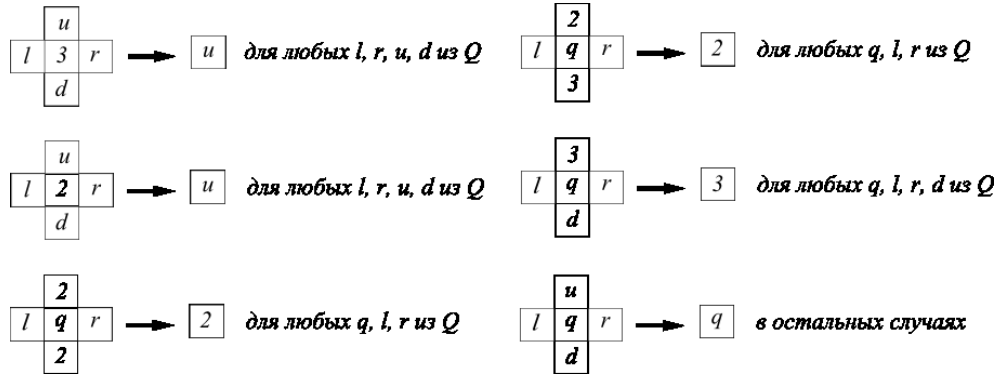


Рис. 1.8: Переходы состояний элементарного автомата с четырьмя состояниями.

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

В данном разделе построены универсальные экраны с конечным числом состояний элементарного автомата, на которых любое изображение строится за время, линейно зависящее от длины и ширины экрана.

Лемма 5. Для любых $n, m \in \mathbb{N}$

$$T(n, m, 4) \leq 2 \min(n, m) + 1.$$

Доказательство. Построим универсальный (n, m) -экран и внешний автомат, для которых $T(n, m, 4) \leq 2 \min(n, m) + 1$.

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

Построим элементарный автомат. По условию он имеет 4 состояния, то есть множество состояний равно $Q = E_q = \{0, 1, 2, 3\}$. Левый и правый выходы элементарных автоматов будем считать не существенными. Функцию переходов состояний зададим следующим образом:

$$\varphi(3, l, r, u, d) = u \quad \text{для любых } l, r, u, d \in Q,$$

$$\varphi(2, l, r, u, d) = u \quad \text{для любых } l, r, u, d \in Q,$$

$$\varphi(q, l, r, 2, 2) = 2 \quad \text{для любых } q, l, r \in Q,$$

$\varphi(q, l, r, 2, 3) = 2$ для любых $q, l, r \in Q$,

$\varphi(q, l, r, 3, d) = 3$ для любых $q, l, r, d \in Q$,

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

Допустим, мы хотим расставить в столбце экрана единицы на те же места, что и в слове $i_1 i_2 \dots i_n$, где $i_j \in \{0, 1\}$ при $1 \leq j \leq n$. Запустим сначала состояние 3, которое каждый такт будет двигаться вниз на одну клетку. Далее, будем подавать на вход слово $i_n 2 i_{n-1} 2 \dots i_2 2 i_1$. Пока на экране будет оставаться состояние 3, все слово будет двигаться вниз на одну клетку каждый такт. Как только состояние 3 дойдет до границы экрана и исчезнет, i_n и следующая за ней двойка перестанут двигаться вниз, при этом состояние i_n примет последняя n -я ячейка экрана. В следующий такт $(n-1)$ -я ячейка примет состояние i_{n-1} и так далее все слово $i_1 i_2 \dots i_n$ восстановится в столбце экрана. На это потребуется $2n$ тактов: n тактов нужно, чтобы состояние 3 дошло до последней ячейки экрана. Далее каждый такт одно состояние i_j доходит до своей позиции и останавливается, то есть, чтобы все значения заняли свои позиции, нужно еще n тактов.

Аналогично формируются входные последовательности для свободного входа каждого столбца экрана. На свободные входы, находящиеся на левой границе экрана, в каждый такт будем подавать слово, состоящее только из нулевых значений.

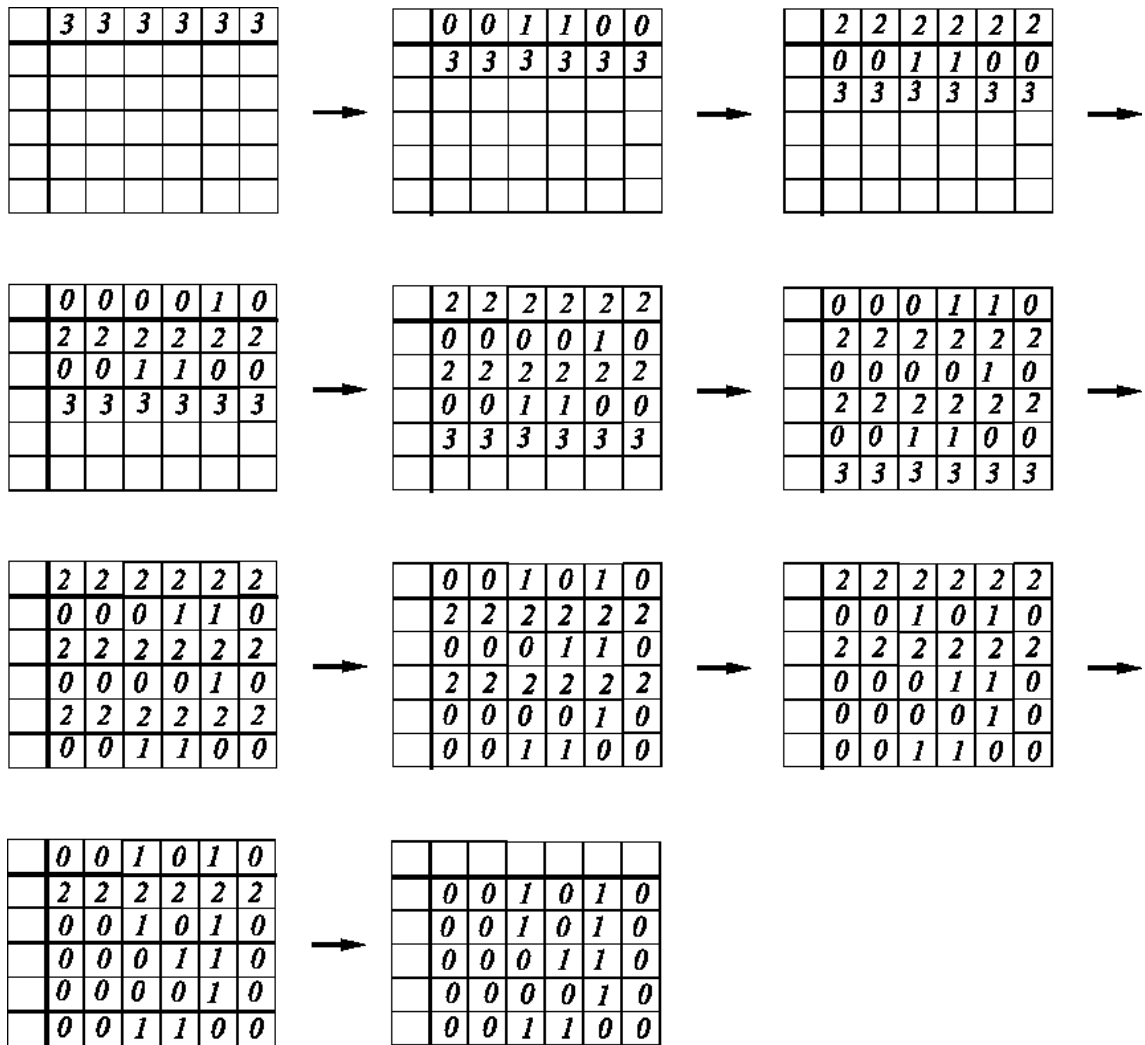
Итак, считая первым тактом подачу первого ненулевого слова на верхние свободные входы экрана, получаем оценку времени $2 \min(n, m) + 1$.

На рисунке 1.8 изображены переходы состояний элементарного автомата.

Описанный алгоритм построения изображений будем называть $\mathcal{P}4$. ■

На рисунке 1.9 приведен пример построения изображения.

Из замечания 1 и леммы 5 следует

Рис. 1.9: Пример построения изображения по алгоритму \mathcal{P}_4 .

Теорема 4. Для любых $n, m \in \mathbb{N}$,

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

$$T(n, m, 4) \leq 2 \min(n, m) + 1.$$

Результаты данного раздела анонсированы в [18] и опубликованы в [12].

4 Оценка времени конструирования изображения при растущем числе состояний элементарного автомата

В этом разделе мы объединим полученные ранее результаты. Нами была получена нижняя оценка времени построения изображения. Также в работе приведен генератор с растущим числом состояний элементарного автомата, с помощью которого эта оценка достигается.

Итак, для (n, m) -экрана S (считаем, что $n \leq m$) мы имеем оценки времени при $Q(S)$ равном 3, 4 и $2n + 2$. Рассмотрим случай, когда $4 < Q(s) < 2n + 2$. Пусть $Q(S) \leq 4 + 2s$, где $1 \leq s \leq n - 2$. Для построения изображений будем использовать алгоритм, который включает в себя два ранее построенных: алгоритм $\mathcal{P}min$ и алгоритм $\mathcal{P}4$.

Новый алгоритм будет следующим. Нижние $n - s$ строк изображения построим с помощью алгоритма $\mathcal{P}4$, а остальные s строк будем строить по алгоритму $\mathcal{P}min$, то есть, за минимально возможное время. Скажем, что состояния 0, 1, 2, 3 соответствуют $\mathcal{P}4$ (назовем их состояниями первой группы), остальные состояния (состояния второй группы) соответствуют алгоритму $\mathcal{P}min$. 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

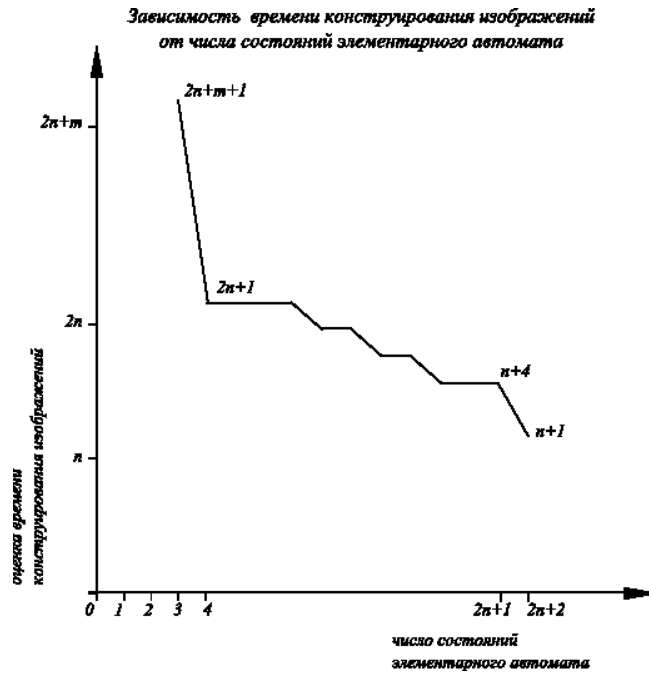


Рис. 1.10: Зависимость времени конструирования изображений от числа состояний элементарного автомата.

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 $4 + 2s + 2$ 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 $4 + 2s$ состояний. Время построения изображений с помощью такого комбинированного алгоритма равно $(2(n - s) + 1) + s + 1 = 2n - s + 2$. То есть, с ростом числа состояний время построения изображений уменьшается, и эта зависимость линейна. Описанный комбинированный алгоритм будем называть \mathcal{P}_{con} .

Объединив этот и полученные ранее результаты получаем

Теорема 5. Если $n, m, s \in \mathbb{N}$, $m, n \geq 2$, $1 \leq s \leq n - 2$, то

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

$$T(n, m, 4) \leq 2 \min(n, m) + 1,$$

$$T(n, m, 4 + 2s) \leq 2 \min(n, m) - s + 2,$$

$$T(n, m, 2 \min(n, m) + 2) = \min(n, m) + 1.$$

На рисунке 1.10 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 неопределенные входы граничных автоматов доопределим нулями. То есть, свободным входом данного экрана является только левый вход первого автомата самой верхней строки. Такой экран будем называть экраном с одним свободным входом.

Внешний автомат теперь имеет один выход. Теперь задача состоит в поиске для любого наперед заданного изображения такого генератора, который строит это изображение на экране с одним свободным входом.

Чтобы получить внешний и элементарный автоматы для этой задачи, модифицируем $\mathcal{P}4$. В этом алгоритме мы использовали состояние 3 элементарного автомата, которое распространялось на весь экран в направлении сверху вниз и по нему проходили сигналы 0 и 1, разреженные состоянием 2. Состояние 3

помогало распознавать границу экрана, то есть тот момент, когда состояния 0 и 1 должны перестать двигаться вниз и остановиться. Воспользуемся этой же идеей.

Чтобы перевести все клетки экрана в одно вспомогательное состояние, будем использовать два вида сигнала — 2 и 3, причем 2 распространяется по вертикали вниз и одновременно порождает сигнал 3, который распространяется по горизонтали слева направо. С помощью этих сигналов можно перевести все клетки экрана в состояние 3.

Далее, сопоставим состояниям 0 и 1 в коде изображения состояния 4 и 5.

Запустим сигнал 6, который будет распространяться сверху вниз по клеткам, находящимся в состоянии 3. Как только этот сигнал достигает нижней строки троек, он порождает сигнал 7, который движется слева направо по клеткам, находящимся в состоянии 3.

Допустим, нижняя строка изображения должна быть равна слову $i_1i_2 \dots i_m$, $i_j \in E_2$, $1 \leq j \leq m$. Чтобы построить ее на экране, заполненном тройками, подадим на вход слово $(i_m + 4)6(i_{m-1} + 4)6 \dots 6(i_1 + 4)6$. Сигналы будут продвигаться вниз, пока не дойдут до нижней строки. Затем сменят направление движения и будут продвигаться слева направо, пока не достигнут границы экрана. Затем, поочередно, каждое i_j расшифруется в 0 или 1 соответственно (за один такт расшифровывается одно значение).

Построенный алгоритм будем называть алгоритм \mathcal{Psn} . Более строго он описан в доказательстве теоремы 6.

Теорема 6. *Если $n, m \in \mathbb{N}$, то*

$$T^1(n, m, 8) \leq 2mn + 5.$$

Доказательство. Алгоритм \mathcal{Psn} .

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

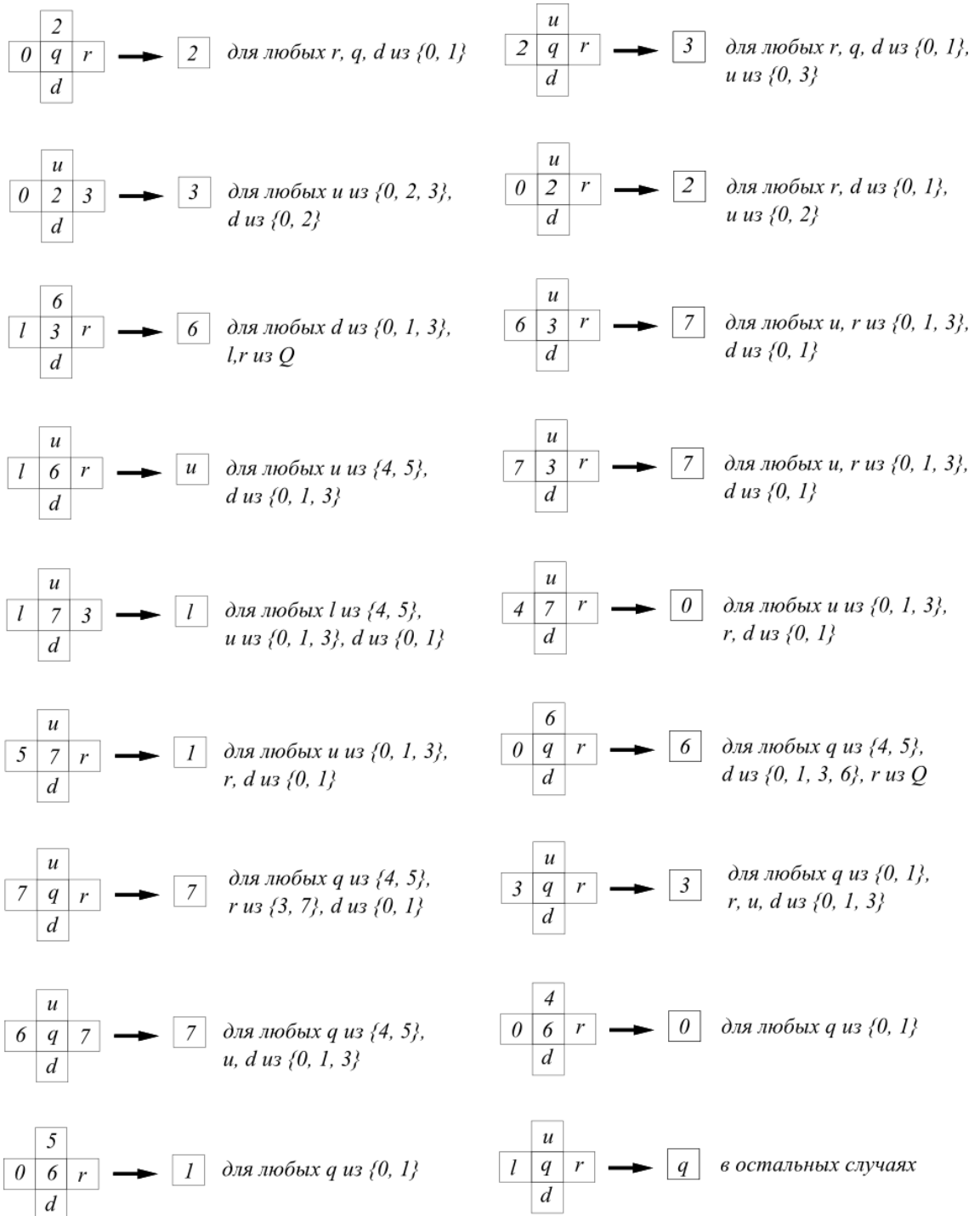
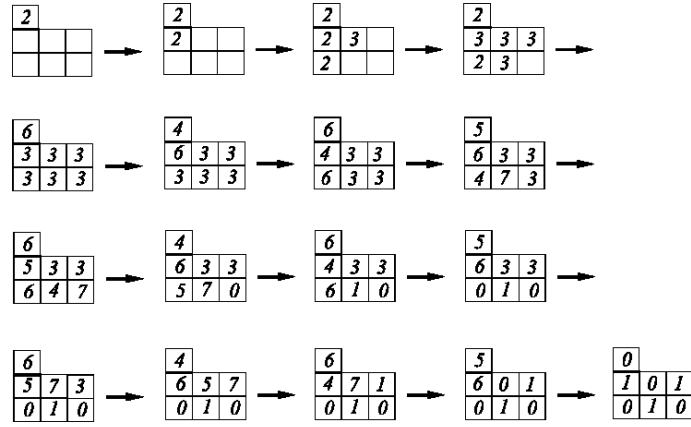


Рис. 1.11: Переходы состояний элементарного автомата с восемью состояниями.

Рис. 1.12: Пример построения изображения по алгоритму $\mathcal{P}8$.

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

- $\varphi(q, 0, r, 2, d) = 2$ для любых $r, q, d \in \{0, 1\}$;
- $\varphi(q, 2, r, u, d) = 3$ для любых $r, q, d \in \{0, 1\}, u \in \{0, 3\}$;
- $\varphi(2, 0, 3, u, d) = 3$ для любых $u \in \{0, 2, 3\}, d \in \{0, 2\}$;
- $\varphi(2, 0, r, u, d) = 2$ для любых $r, d \in \{0, 1\}, u \in \{0, 2\}$;
- $\varphi(3, l, r, 6, d) = 6$ для любого $d \in \{0, 1, 3\}, l, r \in Q$;
- $\varphi(3, 6, r, u, d) = 7$ для любых $r, u \in \{0, 1, 3\}, d \in \{0, 1\}$;
- $\varphi(6, l, r, u, d) = u$ для любых $u \in \{4, 5\}, d \in \{0, 1, 3\}$;
- $\varphi(3, 7, r, u, d) = 7$ для любых $u, r \in \{0, 1, 3\}, d \in \{0, 1\}$;
- $\varphi(7, l, 3, u, d) = l$ для любых $l \in \{4, 5\}, u \in \{0, 1, 3\}, d \in \{0, 1\}$;
- $\varphi(7, 4, r, u, d) = 0$ для любых $u \in \{0, 1, 3\}, r, d \in \{0, 1\}$;
- $\varphi(7, 5, r, u, d) = 1$ для любых $u \in \{0, 1, 3\}, r, d \in \{0, 1\}$;
- $\varphi(q, 0, r, 6, d) = 6$ для любых $q \in \{4, 5\}, d \in \{0, 1, 3, 6\}, r \in E_8$;
- $\varphi(q, 7, r, u, d) = 7$ для любых $q \in \{4, 5\}, r \in \{3, 7\}, d \in \{0, 1\}$;
- $\varphi(q, 3, r, u, d) = 3$ для любых $q \in \{0, 1\}, r, u, d \in \{0, 1, 3\}$;
- $\varphi(q, 6, 7, u, d) = 7$ для любых $q \in \{4, 5\}, u, d \in \{0, 1, 3\}$;
- $\varphi(6, 0, r, 4, d) = 0$ для любых $r, d \in \{0, 1\}$;

$\varphi(6, 0, r, 5, d) = 1$ для любых $r, d \in \{0, 1\}$;

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

На рисунке 1.11 изображены переходы состояний элементарного автомата.

Построим внешний автомат, который соответствует изображению с кодом K .

Он имеет единственный выход a . В первые три такта выход будет одинаковым для любого изображения, а именно:

в первые три такта $a = 2$.

В четвертый такт $a = 6$.

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 $a = 6$,

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 $2mn + 5$ и теорема 6 доказана. ■

На рисунке 1.12 приведен пример построения изображения по алгоритму $\mathcal{P}8$.

Результаты данного раздела были анонсированы в [20] и опубликованы в [12].

6 Построение изображений на многомерных экранах

В предыдущих разделах рассматривались случаи двумерного и одномерного (как частный случай двумерного) экранов. Можно рассмотреть и многомерный случай.

Все описанные алгоритмы построения изображений, а также все возможные алгоритмы для двумерных универсальных экранов, можно применить на k -мерном экране. Для этого нужно выделить любое двумерное подпространство и реализовать алгоритм в каждом сечении экрана вдоль этого подпространства (в "слоях" экрана). В этом случае время построения изображения зависит от размеров сечения экрана вдоль выбранного подпространства. Число состояний экрана совпадает с числом состояний элементарного автомата двумерного экрана, соответствующего используемому алгоритму. Таким образом, оценки времени и числа состояний элементарного автомата для двумерного экрана справедливы и для экранов больших размерностей.

Теорема 7. *Если $n_1, n_2, \dots, n_k \in \mathbb{N}$, $n_1 = \min(n_1, n_2, \dots, n_k)$, $n_2 = \min(n_2, \dots, n_k)$, то*

$$T(n_1, n_2, \dots, n_k, 3) \leq n_1 + n_2 + \min(n_1, n_2) + 1,$$

$$T(n_1, n_2, \dots, n_k, 4) \leq 2n_1 + 1.$$

Доказательство. Разобьем условно экран на $n_3 \cdot \dots \cdot n_k$ k -мерных параллелепипедов размера $n_1 \times n_2 \times 1 \times \dots \times 1$.

Построим вершины первого параллелепипеда. Они будут иметь следующие координаты: $(0, 0, a_3, a_4, \dots, a_k)$, $(n_1, 0, a_3, a_4, \dots, a_k)$, $(0, n_2, a_3, a_4, \dots, a_k)$, $(n_1, n_2, a_3, a_4, \dots, a_k)$, где все a_i принимают значения 0 или 1 и наборы a_3, \dots, a_k упорядочены в лексикографическом порядке.

Каждый следующий параллелепипед можно получить из первого параллельным переносом на вектор $(0, 0, b_3, b_4, \dots, b_k)$, где $0 \leq b_j < n_j$, $j = 3, 4, \dots, k$. Поскольку таких векторов ровно $n_3 \cdot \dots \cdot n_k$, то получим $n_3 \cdot \dots \cdot n_k$ k -мерных параллелепипедов размера $n_1 \times n_2 \times 1 \times \dots \times 1$.

По построению получаем, что каждый построенный параллелепипед имеет по две грани, содержащие только свободные входы экрана, то есть каждый параллелепипед имеет $n_1 + n_2$ свободных входов, причем n_1 вход на одной грани и n_2 входа на другой.

Итак, положим все входы элементарных автоматов кроме первых четырех несущественными. Каждый из полученных параллелепипедов можем теперь рассматривать как двумерный экран и применить к нему все полученные ранее результаты. В частности, имеют место оценки, доказанные в теореме 4. ■

Теорема 8. *Если $n_1, n_2, \dots, n_k, s \in \mathbb{N}$, $n_1 = \min(n_2, n_3, \dots, n_k)$, $n_2 = \min(n_3, \dots, n_k)$, то $n_1, n_2 \geq 2$, $1 \leq s \leq n_1 - 2$, то*

$$T(n_1, n_2, \dots, n_k, 4 + 2s) \leq 2n_1 - s + 2,$$

$$T(n_1, n_2, \dots, n_k, 2n_1 + 2) = n_1 + 1.$$

Доказательство. Доказательство теоремы аналогично доказательству теоремы 7. Здесь используется полученная в теореме 5. ■

В доказательстве теоремы 7 используется разбиение экрана на экраны размера $n_1 \times n_2 \times 1 \times \dots \times 1$. Было выбрано именно такое разбиение с целью минимизировать время построения изображения по уже имеющимся алгоритмам. Вообще говоря, можно выбирать любые $i < j$ и рассматривать разбиение на экраны размера $1 \times \dots \times 1 \times n_i \times 1 \times \dots \times 1 \times n_j \times 1 \times \dots \times 1$, при этом изменяя расположение и количество свободных входов. Количество свободных входов при этом не увеличится.

Глава 2

Сложность управляющего автомата для построения изображений на универсальном экране

1 Предобработка кода изображения

Предыдущая глава была посвящена поиску и изучению универсальных экранов, выявлению связи между сложностью экрана и временем построения изображений. При построении изображения для каждого кода изображения строился внешний автономный автомат. Теперь же внешний автомат рассматривается как автомат со входом. На его вход поступает код изображения — матрица из нулей и единиц, размер которой совпадает с размером экрана, позиции единиц в матрице соответствуют позициям точек в изображении, нули — отсутствию точки в этой позиции. Автомат может обращаться к элементам матрицы в том порядке, в котором требует конкретный алгоритм. В некоторый момент после начала работы автомат начинает выдавать последовательности для свободных входов экрана.

Работа управляющего автомата условно разбивается на две части: предобработку кода изображения и непосредственно генерацию выходной последо-

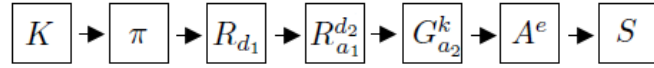


Рис. 2.1: Схема построения последовательностей входных слов для свободных входов экрана вательности. Предобработка осуществляется набором элементов, являющихся стандартными для обработки потоков данных: это перестановка; устройство, которое собирает отдельные элементы в вектор из элементов; устройство, которое вставляет между элементами последовательности вспомогательные элементы; устройство, которое вставляет в начало последовательности заданное количество вспомогательных элементов. Будем называть эти элементы соответственно перестановка π , разреживатель с коэффициентом d_1 , разреживатель с коэффициентом d_2 , предобработчик G_a^k .

Примером предобработки является работа следующей конфигурации элементов. На вход перестановки π поступает код K изображения. π формирует из него некоторый информационный вектор, который поступает на вход разреживателя R_{d_1} . R_{d_1} объединяет элементы входного вектора в группы по d_1 элементов и подает на вход разреживателю $R_{a_1}^{d_2}$, который между этими информационными элементами вектора вставляет элементы a_1 . Полученный вектор поступает на вход предобработчика $G_{a_2}^k$, который в начало вектора добавляет k элементов a_2 . Полученная таким образом последовательность поступает на вход автомата \mathcal{A}_e , который генерирует последовательности для свободных входов экрана. Описанная схема изображена на рисунке 2.1.

Ниже даны строгие определения перечисленных устройств.

Перестановкой π для $n, m \in \mathbf{N}$ будем называть функцию

$$\pi : \mathcal{K}(n, m) \rightarrow E_2^r,$$

где $r \in \mathbf{N}$ — некоторое натуральное число.

Разреживателем R_d с коэффициентом $d \in \mathbf{N}$ для некоторого множества

V^{dk} , $k \in \mathbf{N}$ будем называть функцию

$$R_d : V^{dk} \rightarrow (V^d)^k,$$

такую что если $v = v_1 \dots v_{d \cdot k} \in V^{dk}$, то $R_d(v) = (v_1 \dots v_d) \dots (v_{d(k-1)+1} \dots v_{dk})$, то есть исходное слово v разбивается на подслова длины d .

Разреживателем R_a^d с коэффициентом $d \in \mathbf{N}$ и параметром a для некоторого множества $(V^l)^k$, $l, k \in \mathbf{N}$ будем называть функцию

$$R^d : (V^l)^k \rightarrow V^l \times \{a\}^{d-1} \times V^l \times \dots \times V^l \times \{a\}^{d-1} \times V^l,$$

(возможно $a \in V$), такую что если $v = v_1 \dots v_k \in (V^l)^k$, где $v_i \in V^l$, то

$$R^d(v) = v_1 a \dots a v_2 a \dots a \dots v_{k-1} a \dots a v_k,$$

где между каждыми l буквами слова v вставлено по $(d-1)$ -й букве a .

Предобработчиком G_a^k , $k \in \mathbf{N} \cup \{0\}$ для некоторого множества V^s (возможно $a \in V$), $s \in \mathbf{N}$ будем называть функцию

$$G_a^k : V^s \rightarrow \{a\}^k \times V^s,$$

такую что если $v = v_1 \dots v_s \in V^s$, то $G_a^k(v) = a \dots a v_1 \dots v_s$, то есть буква a вставлена в начало исходного вектора k раз.

Поскольку все перечисленные устройства, кроме автомата A_e , являются стандартными, считаем, что они имеются в наличии и могут нами использоваться. В этой ситуации интерес представляет сложность автомата A_e , который непосредственно генерирует выходную последовательность и при этом не является стандартным устройством. Далее под сложностью управляющего автомата будем подразумевать сложность автомата A_e .

2 Сложность управляющего автомата

В этом разделе приводятся оценки числа состояний внешнего автомата для универсальных генераторов, приведенных в главе 1. Напомним, что универсальные

генераторы $\mathcal{P}3(n, m)$ и $\mathcal{P}4(n, m)$ строят любое изображение за линейное время на универсальном экране, имеющем соответственно 3 и 4 состояния элементарного автомата. $\mathcal{P}_{min}(n, m)$ строит любое изображение за минимально возможное время, а $\mathcal{P}8(n, m)$ строит любое изображение на экране с одним свободным входом и 8 состояниями элементарного автомата.

Если $\mathcal{P}(n, m)$ — универсальный генератор, то множество всех универсальных генераторов, строящих такие же последовательности входных слов, что и $\mathcal{P}(n, m)$, обозначим через $\mathcal{G}(\mathcal{P}(n, m))$. Тогда если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P})$, то обозначим $Q(\mathcal{A}_e)$ — число состояний внешнего автомата \mathcal{A}_e .

Лемма 6. Если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P}3(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то

$$\min_{G \in \mathcal{G}(\mathcal{P}3(n, m))} Q(\mathcal{A}_e) = 2.$$

Доказательство. Построим сначала входную последовательность для управляющего автомата, а затем сам внешний автомат \mathcal{A}_e , который имеет два состояния и который генерирует соответствующие универсальному генератору $\mathcal{P}3(n, m)$ последовательности для свободных входов экрана.

Перестановка π обрабатывает элементы кода изображения начиная с нижней строки. Внутри строки элементы берутся слева направо, и передаются на следующее устройство. Через каждые m элементов перестановка делает перерыв на m тактов. Первый разреживатель имеет параметр m , он собирает полученные от перестановки элементы в слова длины m и передает на вход следующему устройству. Следующий разреживатель имеет параметр 2 и в качестве вспомогательного элемента слово 2^m длины m . То есть, этот разреживатель через один такт получает входные слова, которые являются строками кода изображения, и между этими элементами вставляет по одному элементу — вспомогательному вектору. Затем последовательность передается на вход предобработчика с параметрами $(m + 1)$ и 2^m . В первые $m + 1$ тактов предобработчик выдает вектор

2^m , далее выдает последовательность, полученную им на вход. Если на вход предобработчика ничего не поступает, то он выдает пустое слово. Построенная таким образом последовательность поступает на вход внешнего автомата \mathcal{A}_e .

Внешний автомат имеет следующий вид: $\mathcal{A}_e = (E_3^m \cup \{\Lambda\}, E_3^{n+m}, E_2, \varphi, \psi)$.

Если $q = 0$ и на вход автомата поступает слово 2^m , то выходы — это слова длины 2^n и 2^m соответственно, и автомат переходит в состояние 1.

Если $q = 0$ и на вход автомата поступает слово, отличное от 2^m или пустое слово, то выход автомата — это слова 0^n и 0^m , состояние автомата не меняется.

Если $q = 1$ и на вход автомата поступает слово $a_1 \dots a_m$, где $a_1, \dots, a_m \in E_3$, то выходы равны 0^n и $a_1 \dots a_m$, состояние автомата не меняется.

Если $q = 1$ и на вход автомата поступает пустое слово, то выходы равны 0^n и 0^m , и автомат переходит в состояние 0.

Поскольку в состоянии 0 и 1, получая на вход слово 2^m , автомат выдает разные выходные слова, то эти состояния отличимы, то есть, одного состояния внешнего автомата недостаточно, чтобы реализовать универсальный генератор $\mathcal{P}3(n, m)$. ■

Лемма 7. Если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P}4(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то

$$\min_{G \in \mathcal{G}(\mathcal{P}4(n, m))} Q(\mathcal{A}_e) = 1.$$

Доказательство. Построим сначала входную последовательность для управляющего автомата, а затем сам внешний автомат \mathcal{A}_e , который имеет одно состояние и который генерирует соответствующие универсальному генератору $\mathcal{P}4(n, m)$ последовательности слов для свободных входов экрана.

Перестановка π обрабатывает элементы кода изображения начиная с нижней строки. Внутри строки элементы берутся слева направо, и передаются на следующее устройство. Через каждые m элементов перестановка делает перерыв на m тактов. Первый разреживатель имеет параметр m , он собирает получен-

ные от перестановки элементы в слова длины m и передает на вход следующему устройству. Следующий разреживатель имеет параметр 2 и в качестве вспомогательного элемента слово 2^m длины m . Этот разреживатель через один такт получает входные слова, которые являются строками кода изображения, и между этими элементами вставляет по одному элементу — вспомогательному вектору. Затем последовательность передается на вход преобразовщика с параметрами 1 и 3^m . В первый такт преобразовщик выдает вектор 3^m , далее выдает последовательность векторов, полученную им на вход. Если на вход преобразовщика ничего не поступает, то он выдает пустое слово. Построенная таким образом последовательность поступает на вход внешнего автомата \mathcal{A}_e .

Внешний автомат имеет следующий вид: $\mathcal{A}_e = (E_3^m \cup \{\Lambda\}, E_3^{n+m}, \{0\}, \varphi, \psi)$.

Если на вход внешнего автомата поступает слово $a_1 \dots a_m$, где $a_1, \dots, a_m \in E_3$, то выходные слова всегда равны 0^n и $a_1 \dots a_m$, если на вход поступает пустое слово, то автомат выдает слова 0^n и 0^m .

Это означает, что внешний автомат в данном случае не меняет полученные на вход слова. То есть, входные слова для управляющих входов экрана генерируются стандартными устройствами. ■

Лемма 8. Если $G = \langle \mathcal{A}_e, S(n, m) \rangle = G \in \mathcal{G}(\mathcal{P}min(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то

$$\min_{G \in \mathcal{G}(\mathcal{P}min(n, m))} Q(\mathcal{A}_e) = n.$$

Доказательство. Построим сначала входное слово для управляющего автомата, а затем сам внешний автомат \mathcal{A}_e , который имеет n состояний и который генерирует соответствующие алгоритму $\mathcal{P}min(n, m)$ входные слова для управляющих входов экрана.

Перестановка π берет элементы кода изображения начиная с нижней строки. Внутри одной строки элементы берутся справа налево и передаются на следующее устройство. Через каждые m элементов перестановка делает перерыв на

m тактов. Первый разреживатель имеет параметр m , он собирает полученные от перестановки элементы в слова длины m и передает на вход следующему устройству. Следующий разреживатель имеет параметр 1. Этот разреживатель передает полученное слово на вход следующего устройства без изменений. Преобразователь имеет параметр 0. Он выдает в точности то слово, которое поступило ему на вход. Если на вход преобразователя ничего не поступает, то он выдает пустое слово. Построенная таким образом последовательность поступает на вход внешнего автомата \mathcal{A}_e .

Управляющий автомат имеет следующий вид: $\mathcal{A}_e = ((E_2)^m \cup \{\Lambda\}, (E_{2n+2})^{n+m}, E_n, \varphi, \psi)$.

Выход управляющего автомата в каждый момент времени — это два слова длины n и m соответственно. Первое выходное слово всегда состоит только из буквы 0 и равно 0^n . Второе выходное слово зависит от состояния автомата.

Состояния автомата соответствуют номерам строк в изображении. Состояние задает счетчик, отсчитывающий номер строки, в которой сигнал должен остановиться, а полученное на вход слово задает состояния ячеек этой строки в изображении (0 или 1). Таким образом, выходное слово внешнего автомата формируется из входного слова и счетчика, соответствующего данному состоянию. Поскольку строк n , то и различных счетчиков необходимо n . То есть, состояния, получившие одинаковые коды для различных строк, реализуют различные выходные векторы, а значит, любые два состояния автомата \mathcal{A}_e попарно отличимы. ■

Лемма 9. Если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P}8(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то

$$\min_{G \in \mathcal{G}(\mathcal{P}8(n, m))} Q(\mathcal{A}_e) = 1.$$

Доказательство. Построим сначала входную последовательность для управляющего автомата, а затем сам внешний автомат \mathcal{A}_e , который имеет одно

состояние и строит соответствующие генератору $\mathcal{P}8(n, m)$ последовательности входных слов для свободных входов экрана.

Перестановка π берет элементы кода изображения начиная с нижней строки. Внутри строки элементы берутся слева направо и передаются на следующее устройство. Первый разреживатель имеет параметр 1, он передает последовательность на вход следующему устройству без изменений. Следующий разреживатель имеет параметр 2 и вспомогательный элемент 6. Этот разреживатель через один такт получает входные элементы и между этими элементами вставляет по одной букве 6. Затем слово передается на вход предобработчика с параметрами 4 и 2. В первые четыре такта предобработчик выдает выход равный 2, далее выдает слово, полученное им на вход. Если на вход предобработчика ничего не поступает, то он выдает пустое слово. Построенное таким образом слово поступает на вход внешнего автомата \mathcal{A}_e .

Управляющий автомат имеет следующий вид: $\mathcal{A}_e = (\{0, 1, 2, 4, \Lambda\}, \{0, 2, 4, 5, 6\}, \{0\}, \varphi, \psi)$.

Выход управляющего автомата в один такт — это одна буква, будем обозначать ее b . Состояние автомата всегда остается неизменным.

Если на вход автомата поступает 2, то выход равен 2;

если на вход автомата поступает 0, то выход равен 6;

если на вход автомата поступает 1, то выход равен 5;

если на вход автомата поступает 4, то выход равен 4;

если на вход автомата поступает пустое слово, то выход равен 0. Построенный автомат имеет одно состояние, и последовательности букв, выдаваемые этим автоматом, соответствуют генератору $\mathcal{P}8(n, m)$. ■

Из лемм 6, 7, 8 и 9 следует теорема.

Теорема 9. Если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P}3(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то $\min_{G \in \mathcal{G}(\mathcal{P}3(n, m))} Q(\mathcal{A}_e) = 2$.

Если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P}4(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то $\min_{G \in \mathcal{G}(\mathcal{P}4(n, m))} Q(\mathcal{A}_e) = 1$.

Если $G = \langle \mathcal{A}_e, S(n, m) \rangle = G \in \mathcal{G}(\mathcal{P}min(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то $\min_{G \in \mathcal{G}(\mathcal{P}min(n, m))} Q(\mathcal{A}_e) = n$.

Если $G = \langle \mathcal{A}_e, S(n, m) \rangle \in \mathcal{G}(\mathcal{P}8(n, m))$, $n, m \in \mathbb{N}$, $n \leq m$, то $\min_{G \in \mathcal{G}(\mathcal{P}8(n, m))} Q(\mathcal{A}_e) = 1$.

Результаты данного раздела были анонсированы в [21] [19] и опубликованы в [13].

Глава 3

Конструирование движущихся изображений клеточными автоматами

1 Движение точки на конечном экране

Данный раздел посвящен движению одной точки на конечном экране.

Лемма 10. *Если $m \in \mathbb{N}$ и $m \geq 2$, то $Q(I^1, \mathcal{F}^{1,0} \cap \mathcal{F}_m, m) > 3$.*

Доказательство. Будем доказывать утверждение от противного. Пусть $S(A, m)$ — экран, удовлетворяющий условию леммы и пусть A — его элементарный автомат, причем $Q(A) = 2$. Пусть тогда для определенности $Q = \{0, 1\}$, $1 \in L$, $0 \in Q \setminus L$ и \mathcal{A}_e — управляющий автомат, такой, что для любого $m \in \mathbb{N}$ генератор $\langle \mathcal{A}_e, A, m \rangle$ формирует на экране изображение, состоящее из одной точки, которое движется по закону $F \in \mathcal{F}^{1,0} \cap \mathcal{F}_m$.

Поскольку $\varphi(0, 0, 0) = 0$ и существует момент, когда на экране появится метка, то $\varphi(1, 0, 0) = 1$. Но это означает, что в каждый следующий момент в соседней справа от метки клетке будет появляться метка, то есть изображение всегда будет двигаться по закону $F = 1^m$, что противоречит нашему предположению.

Предположим теперь, что $Q(A) = 3$, $Q = \{0, 1, 2\}$, пусть $1 \in L$, $0 \in Q \setminus L$ и пусть \mathcal{A}_e^F — управляющий автомат, такой что для любого $m \in \mathbb{N}$ генера-

тор $\langle \mathcal{A}_e, A, m \rangle$ формирует на экране изображение, состоящее из одной точки, которое движется по закону $F \in \mathcal{F}^{1,0}$.

Рассмотрим два случая. 1. Пусть $2 \in L$. Тогда $\{\varphi(1, 0, 0), \varphi(2, 0, 0)\} = \{0, a\}$, где a — метка. Иначе метка либо никогда не появится на экране (если $\{\varphi(1, 0, 0), \varphi(2, 0, 0)\} = \{0\}$), либо появится и будет двигаться по закону $F = 1^m$ (если $\{\varphi(1, 0, 0), \varphi(2, 0, 0)\} = \{a, b\}$, где a и b — метки, возможно $a = b$).

Без ограничения общности будем считать $\varphi(1, 0, 0) = 0, \varphi(2, 0, 0) = a$.

Если $\varphi(2, 0, 0) = 2$, то как только на экране появится метка, каждый такт она будет двигаться на одну клетку вправо, то есть будет двигаться по закону $F = 1^m$, что противоречит нашему предположению, значит, $\varphi(2, 0, 0) = 1$. Значит, в следующий такт после появления точки, на экране будет находиться конфигурация $0, 1, 0$.

Рассмотрим, чему равно значение $\varphi(0, 1, 0)$.

Если $\varphi(0, 1, 0) = 0$, то изображение исчезает, чего не может быть по определению, поскольку после начала движения и до конца движения на экране всегда есть ровно одна метка.

Если $\varphi(0, 1, 0) = 1$, то метка на экране никогда не меняет своего положения.

Если $\varphi(0, 1, 0) = 2$, то на экране реализуется движение изображения только по закону $1(01)^m$, и не реализуются другие законы движения.

Значит, в случае $2 \in L$ наше предположение неверно и $Q(A) > 3$.

2. Пусть теперь $2 \in Q \setminus L$, и пусть изображения строятся с предобработкой, то есть, к моменту появления первой метки конфигурация на экране состоит из элементов множества $Q \setminus L = \{0, 2\}$. Значит существует конфигурация $(1, a, b), a, b \in \{0, 2\}$, т.ч. $\varphi(1, a, b) = 1$. Но это означает, что как только на экране появляется одна метка, появление следующих меток полностью определяется конфигурацией из нулей и двоек на экране, заданной во время предобработки.

Поскольку таких конфигураций конечное число, то и законов движения, реализуемых с их помощью, будет конечное число, а всего законов движения — счетное число. Значит, предположение неверно. ■

Теорема 10. *Если $m \geq 2, m \in \mathbb{N}$, то $Q(I^1, \mathcal{F}^{1,0} \cap \mathcal{F}_m, m) = 4$.*

Доказательство. В лемме 10 было доказано неравенство $Q(I^1, \mathcal{F}_m \cap \mathcal{F}_m, m) > 3$.

Построим экран $S(A, m)$, для которого $Q(A) = 4$. Для элементарного автомата этого экрана множество состояний равно $E_4 = \{0, 1, 2, 3\}$. Функция переходов имеет следующий вид:

$$\varphi(3, 0, 0) = 1,$$

$$\varphi(a, b, c) = a \text{ для любого } a, b \in \{0, 2\}, c \in E_4,$$

$$\varphi(2, 1, 0) = 3,$$

$$\varphi(a, 3, 0) = a, \text{ для любого } a \in \{0, 2\},$$

$$\text{в остальных случаях } \varphi(a, b, c) = b, a, b, c \in E_4.$$

Множество меток положим $L = \{1, 3\}$. Метка 1 остается без изменения до тех пор, пока соседний слева автомат не примет состояние 2. В этом случае она меняется на метку 3. Метка 3 исчезает в следующий такт, одновременно с этим справа от нее появляется метка 1. Сигнал 2 является вспомогательным сигналом, он сам движется со скоростью 1. Фактически этот сигнал дает метке команду в следующий такт передвинуться на одну клетку вправо. Движущий сигнал 2 можно подавать с любой частотой, тем самым двигая точку по любому наперед заданному закону движения $F \in \mathcal{F}^{1,0} \cap \mathcal{F}_m$. ■

На рисунке 3.1 изображен пример движения точки на конечном экране.

Лемма 11. *Существует такой элементарный автомат A с 7 состояниями, что для любого натурального m экран $S(A, m)$ является универсальным для изображения I^1 .*

$$F=(01001001\dots) \quad L=\{1,3\}$$

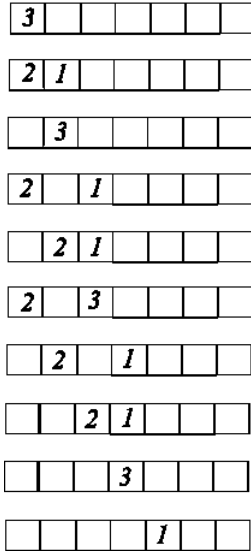


Рис. 3.1: Пример движения точки на экране

Доказательство. Чтобы получить алгоритм, который строит точку, движущуюся по любому наперед заданному закону движения, модифицируем алгоритм из доказательства теоремы 10.

В функции переходов состояний заменим равенство $\varphi(3, 0, 0) = 1$ равенством $\varphi(3, 0, 0) = 3$. Таким образом каждый такт метка будет двигаться на одну клетку вправо.

Введем специальное стоп-состояние s , которое останавливает движение метки 3, если метка оказывается слева от клетки, находящейся в стоп-состоянии. При этом метка 3 исчезает, а стоп-состояние переходит в метку 1 из доказательства теоремы 10.

Таким образом, используя движущие сигналы, мы можем заставить метку двигаться, а используя стоп-сигналы — остановиться.

Остается предварительно расставить стоп-состояния, останавливающие движение со скоростью 1.

Расставим стоп-состояния с помощью следующего алгоритма. Пусть $Q' = \{0, s, 5, 6\}$, где s — стоп-состояние. Зададим функцию переходов:

$$\varphi(l, 6, r) = l \text{ для любого } l, r \in Q',$$

$\varphi(l, 5, r) = l$ для любого $l, r \in Q'$,

$\varphi(5, q, 5) = 5$ для любого $q \in Q'$,

$\varphi(5, q, 6) = 5$ для любого $q \in Q'$,

$\varphi(6, q, r) = 6$ для любого $q \in Q'$,

$\varphi(l, q, r) = q$ в остальных случаях.

Допустим, мы хотим расставить на экране стоп-состояния на те же места, что и в слове i_1, i_2, \dots, i_m , где $i_j \in \{0, s\}$ при $1 \leq j \leq m$. Запустим сначала состояние 6, которое каждый такт будет двигаться вправо на одну клетку. Далее, будем подавать на вход слово $i_m 5 i_{m-1} 5 \dots i_2 5 i_1$. Пока на экране будет оставаться состояние 6, все слово будет двигаться вправо на одну клетку каждый такт. Как только состояние 6 дойдет до границы экрана и исчезнет, i_m и следующая за ней пятерка перестанут двигаться вправо, при этом состояние i_m примет последняя m -я ячейка экрана. В следующий такт $(m - 1)$ -я ячейка примет состояние i_{m-1} и так далее все слово i_1, i_2, \dots, i_m восстановится на экране. На это потребуется $2m$ тактов: m тактов нужно, чтобы состояние 6 дошло до последней ячейки экрана. Далее каждый такт одно состояние i_j доходит до своей позиции и останавливается, то есть, чтобы все значения заняли свои позиции, нужно еще m тактов.

Итак, произвольный закон движения реализуем следующим образом. Сначала расставим стоп-сигналы. Затем запустим метку 3, которая движется вперед (вправо) до тех пор, пока не встретит стоп-сигнал. Когда 3 встречает стоп-сигнал, состояние 3 исчезает, а стоп-сигнал переходит в состояние 1, которое также является меткой. Состояние 1 остается на месте. Чтобы сдвинуть это состояние, пошлем к нему сигнал 2. Когда он достигнет метки 1, метка превратится в метку 3, которая снова будет двигаться до тех пор, пока не встретит стоп-сигнал.

Итак, множество состояний построенного автомата следующее $Q =$

$\{0, 1, 2, 3, s, 5, 6\}$, $L = \{1, 3\}$, то есть, автомат имеет 7 состояний.

Таким образом, существует универсальный экран $S(A, m)$ для изображения I^1 , такой, что $Q(A) = 7$, отсюда следует оценка числа состояний универсального автомата, удовлетворяющего условию леммы. ■

Из лемм 10 и 11 следует теорема.

Теорема 11. *Если $m \in \mathbb{N}$ и $m \geq 2$, то $4 \leq Q(I^1, m) \leq 7$.*

Результаты данного раздела анонсированы в [22] [23] и опубликованы в [14].

2 Движение многоточечных изображений на конечном экране

Теорема 12. *Для любого натурального k существует такой элементарный автомат A с 7^k состояниями, что для любого натурального m экран $S(A, m)$ является универсальным для любого изображения I , состоящего из не более чем k точек.*

Доказательство. Требуемый элементарный автомат можно получить как k -кратное декартово произведение элементарного автомата из леммы 11. При этом каждый из k автоматов будет изображать движение своей точки. ■

Если на изображения наложить некоторые ограничения, то можно построить более дешевые универсальные экраны. В этом разделе приводятся несколько таких примеров, когда изображения состоят из нескольких точек или отрезков.

Лемма 12. *Для любого изображения I и закона $F = 1^m$ выполнено $\mathcal{U}(I, F, m) \neq \emptyset$, причем $Q(I, F, m) = 2$.*

Доказательство. Элементарный автомат A имеет входной и выходной алфавиты, а также множество состояний, равные E_2 . Функция переходов имеет

	1	1^l	2^l	3^l
1	-	-	-	-
1^r	-	-	-	-
2^r	+	+	+	+
3^r	-	-	-	-

Рис. 3.2: Комбинации состояний при движении отрезка

вид $\varphi(a, b, c) = a$ для любых $a, b, c \in E_2$. Очевидно, для любого изображения существует входная последовательность для экрана (совпадающая с кодом этого изображения, записанным начиная с последнего элемента и заканчивая первым), при подаче которой на свободный вход на экране сформируется это изображение и оно будет двигаться по закону $F = (1^m)$. ■

Теорема 13. Если $k, m \geq 2, k, m \in \mathbb{N}$ и $I_{seg}^1 = 1^k$, то

$$Q(I_{seg}^1, \mathcal{F}^{1,0} \cap \mathcal{F}_{m+k-1}, m) \leq 12.$$

Доказательство. Будем считать, что отрезок состоит из трех типов точек — левый край, внутренние точки и правый край. Если отрезок имеет длину 1, считаем, что он состоит из одной правой граничной точки, если длину 2, то из двух граничных точек без внутренних. Граничные точки отрезка будем двигать согласно алгоритму из доказательства теоремы 10.

Обозначим через $1^l, 2^l, 3^l, 0^l$ состояния для левой границы отрезка, соответствующие состояниям 1, 2, 3 и 0 из доказательства теоремы 10, $1^r, 2^r, 3^r, 0^r$ — соответствующие состояния для правой границы отрезка. Положим φ_l и φ_r — соответствующие функции переходов состояний. Положим 1 — состояние внутренней точки отрезка.

Построим множество состояний автомата A как подмножество $\{1^l, 2^l, 3^l, 0^l, 1\} \times \{1^r, 2^r, 3^r, 0^r\}$. Пусть сначала $Q = \emptyset$. Состояния, соответствующие левой границе и внутренним точкам отрезка, поместим в первую координату

нату, а состояния, соответствующие правой границе отрезка — во вторую. Положим $Q = Q \cup \{(0^l, 0^r), (1, 0^r), (1^l, 0^r), (2^l, 0^r), (3^l, 0^r), (0^l, 1^r), (0^l, 2^r), (0^l, 3^r)\}$.

Рассмотрим всевозможные декартовы произведения состояний из множества $\{1^l, 2^l, 3^l, 1\}$ и $\{1^r, 2^r, 3^r, 1\}$ и выберем только те из них, которые могут встретиться при движении отрезка на экране, это: $(1, 2^r), (2^l, 2^r), (3^l, 2^r), (1^l, 2^r)$. На рисунке 3.2 знаком плюс отмечены комбинации, которые могут встретиться на экране, а знаком минус — комбинации, которые не встретятся.

Положим $Q = Q \cup \{(1, 2^r), (2^l, 2^r), (3^l, 2^r), (1^l, 2^r)\}$.

Итак, построено множество состояний автомата A : $Q = \{(0^l, 0^r), (1, 0^r), (1^l, 0^r), (2^l, 0^r), (3^l, 0^r), (0^l, 1^r), (0^l, 2^r), (0^l, 3^r), (1, 2^r), (2^l, 2^r), (3^l, 2^r), (1^l, 2^r)\}$. Положим $L = \{(1, 0^r), (1^l, 0^r), (3^l, 0^r), (0, 1^r), (0, 3^r), (1, 2^r), (3^l, 2^r), (1^l, 2^r)\}$.

Все компоненты с индексами l и r меняются независимо друг от друга по законам, описанным в доказательстве теоремы 10, то есть, $\varphi = (\varphi_l, \varphi_r)$.

Дополним φ следующими правилами:

$$\varphi((3^l, b_1), (1, c), (a_2, b_2)) = (1^l, \varphi_r(b_1, c, b_2)), \text{ для любых } b_1, a_2, b_2, c,$$

$$\varphi((a_1, b_1), (1, c), (a_2, b_2)) = (1, \varphi_r(b_1, c, b_2)), \text{ для } a_1 \in \{1, 1^l\} \text{ и любых } b_1, a_2, b_2, c,$$

$$\varphi((a_1, b_1), (0, c), (a_2, 3^r)) = (1, \varphi_r(b_1, c, 3^r)) \text{ для любых } a_1, b_1, a_2, c.$$

Теперь для любого $F \in \mathcal{F}^{1,0} \cap \mathcal{F}_{m+k-1}$ можно подавать движущие сигналы для левого и правого концов отрезка так, чтобы отрезок двигался по закону F .

Таким образом, построен автомат A , для которого $Q(A) = 12$, и существует управляющий автомат \mathcal{A}_e , такой что для любого $m \in \mathbb{N}$ генератор $\langle \mathcal{A}_e, \mathcal{A}, m \rangle$ формирует на экране изображение, состоящее из одного отрезка, который движется по закону $F \in \mathcal{F}^{1,0} \cap \mathcal{F}_{m+k-1}$. ■

Реализовать движение нескольких точек можно, преобразовав алгоритм из доказательства леммы 11 — для каждой точки посылать сигнал отдельного типа, который влияет на движение только этой точки. Минус такого пре-

образования алгоритма в том, что количество состояний автомата зависит от числа точек в изображении. Ниже приведен универсальный экран с конечным числом состояний для класса изображений \mathcal{I}^k и законов движений $(\mathcal{F}^{1,0} \setminus \mathcal{F}^{1,0,s-1}) \cap \mathcal{F}_{m+|I|-1}$.

Теорема 14. *Для любых натуральных $k, m \geq 2$ и любого изображения I из \mathcal{I}^k , такого, что $s = |I| + k \geq k^2/2 + 3k/2$, выполнено*

$$Q(I, (\mathcal{F}^{1,0} \setminus \mathcal{F}^{1,0,s-1}) \cap \mathcal{F}_{m+|I|-1}, m) \leq 9.$$

Доказательство. Преобразуем алгоритм из доказательства теоремы 10. Для каждой точки изображения будем посылать движущий сигнал. Каждому движущему сигналу добавим след — цепочку автоматов в специальном состоянии. В след сигнала закодируем номер метки, для которой предназначается этот движущий сигнал. Функцию переходов состояний зададим так, что при прохождении через метку длина следа уменьшается на одну клетку. Когда сигнал без следа достигает метки, это означает, что он предназначен этой метке и она передвигается на одну клетку вправо. Таким образом, длина следа сигнала на единицу меньше номера точки, которой предназначен этот сигнал. При прохождении через метку длина следа уменьшается на одну ячейку.

При таком алгоритме расстояние между движущими сигналами должно быть таким, чтобы след сигнала не перекрывался со следующим сигналом и его следом. Кроме того, когда слева от метки появляется движущий сигнал, метке необходим еще один такт, чтобы распознать, есть ли след у этого сигнала. В случае, если следа нет, то в следующий такт метка исчезает, а справа от нее появляется другая метка. Если у движущего сигнала есть след, то в следующий такт движущий сигнал проходит в клетку справа. Таким образом, движущий сигнал проходит через метку с задержкой в один такт. Отсюда ограничение на расстояние между точками отрезка: $i_j - i_{j-1} \geq j + 1$. Также закон движения

должен быть таким, чтобы между любыми двумя движущимися сигналами для k -й точки изображения на экран помещалась последовательность движущихся сигналов для остальных точек. То есть, $F = 10^{s_1}10^{s_2} \dots$ должен быть таким, что $s_i \geq |I| + k = i_k \geq k(k+1)/2 + k = k^2/2 + 3k/2, s_i \in \mathbb{N}$

От такого ограничения на закон движения можно отказаться за счет увеличения числа состояний, добавив новые состояния при наложении следов сигналов друг на друга. При этом снова число состояний будет зависеть от конкретного закона движения.

Построим элементарный автомат, на основе которого получается экран, удовлетворяющий условиям утверждения.

Положим входной и выходной алфавиты и множество состояний равными $Q = \{0, 1, 2, 3, 1^3, 1^{0,3}, 1^{2,3}, 1^2, 1^{3,3}\}$. Множество меток положим равным $L = \{1, 1^3, 1^{0,3}, 1^{2,3}, 1^2, 1^{3,3}\}$.

Функцию переходов зададим следующим образом:

$$\varphi(3, 1, r) = 1^3, \text{ где } r \in Q;$$

$$\varphi(0, 1^3, r) = 1^{0,3}, \text{ где } r \in Q;$$

$$\varphi(l, 1^{0,3}, r) = l, \text{ где } l, r \in Q;$$

$$\varphi(1^{0,3}, q, r) = 1, \text{ где } q, r \in Q;$$

$$\varphi(3, 0, r) = 3, \text{ где } r \in Q;$$

$$\varphi(2, 1^3, r) = 1^{2,3}, \text{ где } r \in Q;$$

$$\varphi(1^{2,3}, 0, r) = 3, \text{ где } r \in Q;$$

$$\varphi(0, 1^{2,3}, r) = 1, \text{ где } r \in Q;$$

$$\varphi(2, 1^{2,3}, r) = 1^2, \text{ где } r \in Q;$$

$$\varphi(0, 1^2, r) = 1, \text{ где } r \in Q;$$

$$\varphi(1^2, 2, r) = 2, \text{ где } r \in Q;$$

$$\varphi(1, 2, r) = 0, \text{ где } r \in Q;$$

$$\varphi(3, 1^{0,3}, r) = 3, \text{ где } r \in Q;$$

	$1l$	$2l$	$3l$	I^3l	$I^{0,3}l$	$I^{2,3}l$	I^2l	$I^{3,3}l$
$1r$	-A	+	+	-	-	B	-	-
$2r$	+	+	+	+	+	+	+	+
$3r$	+	+	+	+	+	+	+	+
I^3r	-	+	+	-	-	-	-	-
$I^{0,3}r$	-	+	+	-	-	-	-	-
$I^{2,3}r$	-	+	+	-	-	-	-	-
I^2r	-	+	+	-	-	-	-	-
$I^{3,3}r$	-	+	+	-	-	-	-	-

Рис. 3.3: Комбинации сигналов для левого и правого концов отрезков

$$\varphi(2, 3, r) = 2, \text{ где } r \in Q;$$

$$\varphi(l, 2, r) = 0, \text{ где } l, r \in Q, l \neq 2, l \neq 1^2;$$

$$\varphi(3, 1^3, r) = 1^{3,3}, \text{ где } r \in Q;$$

$$\varphi(1^{3,3}, 0, r) = 1^3, \text{ где } r \in Q;$$

$$\varphi(2, 1^{3,3}, r) = 2, \text{ где } r \in Q;$$

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

Состояние 3 — это движущий сигнал, 2 — это след сигнала. Индексы у метки 1 позволяют метке сохранять и передавать в следующую клетку информацию о том, что до нее дошел движущий сигнал, а также, есть ли след у этого сигнала.

Используя построенный таким образом автомат можно реализовать движение точек, удовлетворяющее условию утверждения. ■

Теорема 15. Для любых натуральных $k, m \geq 2$ и любого изображения I из \mathcal{I}_{seg}^k , такого, что $s = |I| + k \geq k^2/2 + 3k/2$, выполнено

$$Q(I, (\mathcal{F}^{1,0} \setminus \mathcal{F}^{1,0,s-1}) \cap \mathcal{F}_{m+|I|-1}, m) \leq 54.$$

Доказательство. Для реализации движения нескольких отрезков преобразуем алгоритм из доказательства утверждения 14. Как мы уже делали это ранее, разобьем точки отрезков на три типа: левые концы отрезков, правые концы и внутренние точки отрезков. Как и ранее будем считать, что если отрезок

состоит из одной точки, то он состоит только из правой граничной точки. Движение всех левых концов и всех правых концов отрезков будем реализовывать по построенному в доказательстве утверждения 14 алгоритму.

Обозначим через $0l, 1l, 2l, 3l, 1^3l, 1^{0,3}l, 1^{2,3}l, 1^2l, 1^{3,3}l$ — состояния для левой границы отрезка, соответствующие состояниям $0, 1, 2, 3, 1^3, 1^{0,3}, 1^{2,3}, 1^2, 1^{3,3}$ из доказательства теоремы 10, $0r, 1r, 2r, 3r, 1^3r, 1^{0,3}r, 1^{2,3}r, 1^2r, 1^{3,3}r$ — соответствующие состояния для правой границы отрезка. Положим φ_l и φ_r — соответствующие функции переходов состояний. Положим 1 — состояние внутренней точки отрезка.

Обозначим $Q_l = \{0l, 1l, 2l, 3l, 1^3l, 1^{0,3}l, 1^{2,3}l, 1^2l, 1^{3,3}l\}$ и $Q_r = \{0r, 1r, 2r, 3r, 1^3r, 1^{0,3}r, 1^{2,3}r, 1^2r, 1^{3,3}r\}$ и построим множество состояний автомата A как подмножество $Q_l \times \{0, 1\} \times Q_r$.

Пусть сначала $Q = \emptyset$. Состояния, соответствующие левой границе, поместим в первую координату, состояния, соответствующие внутренним точкам отрезка, поместим во вторую координату, а состояния, соответствующие правой границе отрезка — во третью. Положим $Q = Q \cup Q_l \times \{0\} \times \{0r\} \cup \{0l\} \times \{0\} \times \{Q_r \setminus \{0r\}\}$.

Рассмотрим декартово произведение множеств $Q_l \setminus \{0l\}$, $\{0\}$ и $Q_r \setminus \{0r\}$ и выберем только те комбинации, которые могут встретиться при движении отрезка на экране — это множество состояний обозначим через Q^1 . На рис. 3.3 знаком плюс отмечены комбинации состояний для левого и правого концов отрезка, которые могут встретиться на экране, а знаком минус — комбинации, которые не встретятся.

Положим $Q = Q \cup Q^1$.

Кроме этого, добавим в Q состояния, которые могут встретиться для внутренних точек отрезка: $Q = Q \cup \{0l, 2l, 3, \} \times \{1\} \times \{0r, 2r, 3r\}$.

Итак, построено множество состояний автомата A : $Q = Q_l \times \{0\} \times \{0r\} \cup \{0l\} \times \{0\} \times \{Q_r \setminus \{0r\}\} \cup Q^1 \cup \{0l, 2l, 3, \} \times \{1\} \times \{0r, 2r, 3r\}$. Положим L — множе-

ство таких состояний из Q , для которых хотя бы одна из координат содержит единицу (возможно, с индексом).

Компоненты, содержащие l или r , меняются независимо друга по законам, описанным в доказательстве утверждения 14, то есть, $\varphi = (\varphi_l, \varphi_r)$.

Дополним φ следующими правилами:

$\varphi((a_1, 0, c_1), (q_1, 1, q_3), (a_2, b_2, c_2)) = (\varphi_l(a_1, q_1, a_2), 0, \varphi_r(c_1, q_3, c_2))$, для любых $q_1, q_3, b_2, c_1, c_2, a_1 \in \{1^{0,3l}, 1^{3,3l}\}$;

$\varphi((a_1, b_1, c_1), (q_1, 0, q_3), (a_2, b_2, c_2)) = (\varphi_l(a_1, q_1, a_2), 1, \varphi_r(c_1, q_3, c_2))$, для любых $q_1, q_3, b_2, c_1, c_2, c_2 \in \{1^{0,3r}, 1^{3,3r}\}$.

Теперь для любого $F \in (\mathcal{F}^{1,0} \setminus \mathcal{F}^{1,0,s-1}) \cap \mathcal{F}_{m+|l|-1}$ можно подавать движущие сигналы для левого и правого концов отрезка так, чтобы отрезок двигался по закону F .

Таким образом, построен автомат A , для которого $Q(A) = 9 * 2 - 1 + 28 + 9 = 54$, и существует управляющий автомат \mathcal{A}_e , такой что для любого $m \in \mathbb{N}$ генератор $\langle \mathcal{A}_e, A, m \rangle$ формирует на экране изображение, состоящее из k отрезков, которые движутся по закону $F \in (\mathcal{F}^{1,0} \setminus \mathcal{F}^{1,0,s-1}) \cap \mathcal{F}_{m+|l|-1}$. ■

Результаты раздела были анонсированы в [22] и опубликованы в [14].

3 Движущиеся изображения на бесконечном экране

В данном разделе исследуются способы построения движущихся изображений на бесконечном в одну сторону экране. Найдены законы движения, которые не могут быть реализованы на экране, элементарный автомат которого имеет конечное число состояний. Для некоторых классов законов движений найдены алгоритмы построения изображений, движущихся по этим законам.

Здесь экран представляет из себя бесконечную последовательность из оди-

наковых элементарных автоматов \mathcal{A} , имеющих по два входа. Как и ранее, входы автомата \mathcal{A} называются левым и правым и ими соответственно являются состояния левого и правого соседа. Левый вход первого автомата называется свободным и подключен к выходу управляющего автомата \mathcal{A}_e . При этом функцию переходов состояний автомата \mathcal{A} обозначим через $\varphi(l, q, r)$, где q — текущее состояние автомата, l, r — состояния его левого и правого соседей соответственно. Положим для любого экрана свойство $\varphi(0, 0, 0) = 0$. Бесконечный экран с элементарным автоматом \mathcal{A} будем обозначать через $S(\mathcal{A})$. Пару $G = \langle \mathcal{A}_e, \mathcal{A} \rangle$ будем называть *генератором*. Среди состояний элементарного автомата выделим непустое подмножество L , не содержащее нулевой элемент, и элементы этого множества будем называть *метками*.

Отрезком предобработки экрана в момент t будем называть отрезок подряд идущих ячеек на экране, удовлетворяющий следующим условиям:

- 1) началом отрезка является ячейка, следующая после самой правой метки на экране;
- 2) состояние самой правой ячейки отрезка отлично от 0;
- 3) состояние всех ячеек экрана, следующих за последней (самой правой) ячейкой отрезка предобработки равно 0;
- 4) среди состояний ячеек отрезка нет меток.

Лемма 13. Пусть $S(\mathcal{A})$ — бесконечный экран, и $Q(S) = q$, и на этом экране движется изображение I^1 . Если в момент t^* после начала движения отрезок предобработки экрана имеет длину p и начиная с момента времени t^* точка двигалась вправо $(q - 1)^{p+1} + 1$ тактов подряд, то во все последующие такты точка будет двигаться по закону $F = 1^\infty$.

Доказательство. Пусть Q — множество состояний автомата \mathcal{A} , $L \subset Q$ — множество меток.

Введем несколько обозначений.

Пусть s — отрезок подряд идущих ячеек на экране. Через $C_t(s)$ будем обозначать конфигурацию состояний ячеек этого отрезка в момент времени t . Через $O^{a,b}(s)$ обозначим отрезок экрана, включающий сам отрезок s , а также a ячеек слева и b ячеек справа от отрезка s . $O^{a,b}(s)$ будем называть окрестностью отрезка s . Через $M^i(s)$ будем обозначать отрезок экрана, полученный из s сдвигом вправо на i ячеек, или сдвигом влево на $|i|$ ячеек при отрицательном i .

Обозначим теперь через $s_{t^*}^{p+1}$ такой отрезок ячеек экрана, что в момент времени t^* первой ячейкой отрезка является ячейка, состояние которой является меткой, и отрезок имеет длину $p + 1$.

Рассмотрим последовательность отрезков $s_{t^*}^{p+1}$, $M^i(s_{t^*}^{p+1})$ при $i = 1, \dots, (q - 1)^{p+1} + 1$ и последовательность конфигураций этих отрезков $C_{t^*+i}s_{t^*}^{p+1}$, $C_{t^*+i}(M^i(s_{t^*}^{p+1}))$.

Ясно, что конфигурация отрезка $M^i(s_{t^*}^{p+1})$ в момент времени $t^* + i$ зависит от конфигурации отрезка $O^{1,1}(M^i(s_{t^*}^{p+1}))$ в предыдущий момент времени. Кроме того, поскольку максимальная скорость распространения сигнала равна одной клетке в один такт, то две последние ячейки отрезка $O^{1,1}(M^i(s_{t^*}^{p+1}))$ в момент времени $t^* + i - 1$ находятся в состоянии 0, то есть $C_{t^*+i-1}(O^{1,1}(M^i(s_{t^*}^{p+1}))) = (C_{t^*+i-1}(M^{i-1}(s_{t^*}^{p+1})), 0, 0)$. Это означает, что в приведенной последовательности каждая следующая конфигурация определяется предыдущей.

По условию все конфигурации $C_{t^*+i}s_{t^*}^{p+1}$, $C_{t^*+i}(M^i(s_{t^*}^{p+1}))$ при $i = 1 \dots (q - 1)^{p+1} + 1$ начинаются с меток.

Количество таких конфигураций не превосходит $|L| \cdot |Q \setminus L|^p$. Но поскольку существует хотя бы одно состояние, не являющееся меткой, то $|L| \leq |Q| - 1 = q - 1$, и существует хотя бы одно состояние, являющееся меткой, то $|Q \setminus L| \leq |Q| - 1 = q - 1$. Следовательно, $|L| \cdot |Q \setminus L|^p \leq (q - 1)^{p+1}$.

Значит, среди перечисленных конфигураций найдутся две одинаковые кон-

фигурации. Это означает, что последовательность конфигураций с некоторого момента заикнется, то есть все последующие конфигурации будут начинаться с меток. А это означает, что начиная с момента t^* точка будет двигаться по закону $F = 1^\infty$, что и требовалось доказать. ■

Теорема 16. *Для любого бесконечного экрана $S(\mathcal{A})$ существует такой закон движения $F \in \{0, 1\}^\infty$, что на экране $S(\mathcal{A})$ невозможно реализовать движение изображения I^1 по закону F .*

Доказательство. Пусть $S(\mathcal{A})$ — бесконечный экран и его элементарный автомат \mathcal{A} имеет q состояний. Положим $F_q = 101^{(q-1)^2}01^{(q-1)^4}0\dots 1^{(q-1)^{2(i-1)}}0\dots$. Допустим, что на экране $S(\mathcal{A})$ реализуется движение изображения I^1 по закону F_q с предобработкой p , $p \in \mathbb{N} \cup \{0\}$. Тогда к моменту, когда на экране начинается движение, реализующее i -й отрезок единиц закона движения F_q , отрезок предобработки имеет длину $p + i - 1$. Но тогда существует такое i , что $(q - 1)^{2(i-1)} > (q - 1)^{p+i-1+1}$, а именно любое $i > p + 2$. То есть мы находимся в условии леммы 13, и во все последующие такты точка будет двигаться со скоростью 1. Значит, на экране с q состояниями закон движения F_q одной точки реализовать нельзя. ■

Теорема 16 означает, что для изображения I^1 не существует универсального бесконечного экрана.

Теорема 17. *Существует закон движения $F \in \{0, 1\}^\infty$, движение по которому невозможно реализовать ни на каком экране $S(\mathcal{A})$.*

Доказательство. Положим $F_* = 101^{2^2}01^{3^4}0\dots 1^{2^{2(i-1)}}0\dots$. В этом законе движения i -й отрезок единиц имеет длину $i^{2(i-1)}$.

Допустим, что существует бесконечный экран $S(\mathcal{A})$, на котором реализуется движение изображения I^1 по закону F_* с предобработкой p , $p \in \mathbb{N} \cup \{0\}$. Пусть \mathcal{A} — элементарный автомат и пусть он имеет q состояний.

К моменту, когда на экране начинается движение, реализующее i -й отрезок единиц закона движения F_* , отрезок предобработки имеет длину $p+i-1$. Ясно, что существует такое i , что $i^{2(i-1)} > (q-1)^{p+i-1+1}$, а именно любое $i > \max(q-1, p+2)$. То есть мы находимся в условии леммы 13, и во все последующие такты точка будет двигаться со скоростью 1. Значит, ни для какого q , на экране с q состояниями закон движения F_* одной точки реализовать нельзя. ■

Результаты данного раздела анонсированы в [24] и опубликованы в [14].

4 Движение с ограниченной скоростью на бесконечном экране

В этом разделе будем рассматривать законы движения с ограниченным числом идущих подряд единиц.

В лемме 10 показано, что если для конечного экрана длины $m \geq 2$ выполнено $Q(S) = 2$, то на нем реализуется единственный закон движения $F = 1^m$. Ясно, что на бесконечном экране S , для которого $Q(S) = 2$ также реализуется только один закон движения $F = 1^\infty$, то есть, имеет место утверждение.

Лемма 14. *Если в множестве \mathcal{F} имеется закон движения, содержащий бесконечное число нулей и бесконечное число единиц, и $\mathcal{U}(\mathcal{F}) \neq \emptyset$, то $Q(\mathcal{F}) \geq 3$.*

Используя алгоритм, полученный в доказательстве теоремы 10, можно строить изображение I^1 , движущееся по закону $F \in \mathcal{F}^{1,0}$, то есть, имеет место следующая лемма.

Лемма 15. $\mathcal{U}(\mathcal{F}^{1,0}) \neq \emptyset$ и $Q(\mathcal{F}^{1,0}) \leq 4$.

Рассмотрим класс законов движения $\mathcal{F}^{s,0}$.

Лемма 16. *Если $s \in \mathbb{N}$, $s \geq 2$, то $Q(\mathcal{F}^{s,0}) \leq 2s + 2$.*

Доказательство. Построим универсальный экран S , для которого $Q(S) = 2s + 2$. Для элементарного автомата \mathcal{A} экрана S положим $Q = \{0, d^1, \dots, d^s, l^0, l^1, \dots, l^s\}$. Зададим функцию переходов следующим образом:

$$\varphi(l^0, 0, r) = 0, r \in Q;$$

$$\varphi(l^i, 0, r) = l^{i-1}, \text{ где } 1 \leq i \leq s, r \in Q;$$

$$\varphi(l, d^i, r) = l, \text{ где } l = 0 \text{ или } l = d^i, 1 \leq i \leq s, r \in Q;$$

$$\varphi(d^i, q, 0) = d^i, \text{ где } q = 0 \text{ или } q = l^j, 1 \leq i, j \leq s;$$

$$\varphi(0, l^j, r) = 0, \text{ где } 1 \leq j \leq s, r \in Q;$$

$$\varphi(d^i, l^0, 0) = l^i, \text{ где } 1 \leq i \leq s;$$

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

Положим $L = \{l^i, 0 \leq i \leq s\}$. Группу состояний $d^i, 1 \leq i \leq s$ назовем движущими сигналами.

По построению функции φ видно, что метка l^0 стоит на месте до тех пор, пока ее левым соседом не станет движущий сигнал.

Будем говорить, что остальные метки $l^i, 1 \leq i \leq s$ имеют счетчик, который отсчитывает, сколько тактов точка будет двигаться вперед со скоростью 1. Таким образом, каждая метка $l^i, 1 \leq i \leq s$ остается на экране ровно один такт, причем в следующий такт соседняя справа клетка становится меткой со значением счетчика на 1 меньше, чем имела метка в предыдущий момент. Когда значение счетчика становится равным 0, метка останавливается, пока в соседней слева клетке снова не появится движущий сигнал. При этом уже в следующий такт после остановки метка снова может начать движение.

Таким образом, можно подавать такие движущие сигналы, что на построенном экране можно реализовать любой наперед заданный закон движения, имеющий не более чем s единиц подряд. Поскольку движение метки зависит от состояния самой ячейки и состояния ячейки слева от нее, то состояние правой ячейки не влияет на движение, и движение реализуется без предобработки.

Итак, построен экран, удовлетворяющий условию утверждения, элементарный автомат которого имеет $2s + 2$ состояния, следовательно, получена оценка $Q(\mathcal{F}^{s,0}) \leq 2s + 2$ и лемма доказана. ■

Заметим, что классы $\mathcal{F}^{1,0}$ и $\mathcal{F}^{s,0}$ можно расширить, добавив в начало закона движения произвольный отрезок длины d из нулей и единиц, то есть, можем рассмотреть классы $\mathcal{F}^{1,d}$ и $\mathcal{F}^{s,d}$, $d \in \mathbb{N}$. Для начального отрезка длины d закона движения F из одного из упомянутых классов, сделаем предобработку экрана, расставив на экране стоп-сигналы по аналогии с тем, как это делалось в доказательстве леммы 11. Чтобы реализовать движение изображения в первые d тактов, достаточно не более семи дополнительных состояний. Одно состояние будет двигаться вправо со скоростью одна клетка в два такта и ограничивать длину предобработки. Четыре состояния необходимы для того, чтобы расставить стоп-сигналы. Еще одно состояние-метка будет двигаться на одну клетку вправо каждый такт, пока не встретит стоп-сигнал. Наконец, еще одно состояние будет переводить метку, которая находится в покое, в метку, движущуюся со скоростью 1. Предобработка такого вида займет $2d + 1$ такт. Таким образом, с учетом лемм 14, 15 и 16 имеет место утверждение.

Теорема 18. *Если $s, d \in \mathbb{N}$, то имеют место следующие оценки*

$$3 \leq Q(\mathcal{F}^{s,0}) \leq 2s + 2;$$

$$3 \leq Q(\mathcal{F}^{s,d}) \leq 2s + 9.$$

Результаты данного раздела анонсированы в [24] и опубликованы в [14].

5 Автономно движущиеся изображения на бесконечном экране

Замечание 2. Автономными движениями реализуется счетное число законов движения.

Поскольку конечных автоматов счетное число и преобразований — счетное число, то автономными движениями можно реализовать не более чем счетное число законов движения. Очевидно, что любой периодический закон движения реализуется автономными движениями, а значит, ими реализуется в точности счетное число законов движения.

Лемма 17. Существует непериодический закон движения F , для которого выполнено $v(F) = 1$ и $\mathcal{U}(\{F\}) \neq \emptyset$.

Доказательство. Построим экран $S(A)$, на котором реализуется автономное движение изображения I^1 со скоростью 1. Для этого построим элементарный автомат этого экрана. Положим $Q = \{0, 0^c, 1^c, 2^c, 3, 4\}$ и $L = \{3, 4\}$ и

$$\varphi(0^c, 0, 0) = 1^c;$$

$$\varphi(1^c, 0, 0) = 2^c;$$

$$\varphi(2^c, 0, 0) = 0^c;$$

$$\varphi(0^c, 2^c, r) = 1^c, r \in Q;$$

$$\varphi(1^c, 2^c, r) = 2^c, r \in Q;$$

$$\varphi(0^c, 0^c, r) = 0^c, r \in Q;$$

$$\varphi(1^c, 0^c, r) = 1^c, r \in Q;$$

$$\varphi(2^c, 0^c, r) = 0^c, r \in Q;$$

$$\varphi(0^c, 1^c, r) = 0^c, r \in Q;$$

$$\varphi(1^c, 1^c, r) = 1^c, r \in Q;$$

$$\varphi(2^c, 1^c, r) = 0^c, r \in Q;$$

$t=0$	0^c																				
$t=1$	0^c	1^c																			
$t=2$	3	0^c	2^c																		
$t=3$		3	1^c	0^c																	
$t=4$		4	0^c	1^c	1^c																
$t=5$			3	0^c	1^c	2^c															
$t=6$				3	0^c	2^c	0^c														
$t=7$					3	1^c	0^c	1^c													
$t=8$					4	0^c	1^c	0^c	2^c												
$t=9$						3	0^c	1^c	1^c	0^c											
$t=10$							3	0^c	1^c	1^c	1^c										
$t=11$								3	0^c	1^c	1^c	2^c									

Рис. 3.4: Автономное движение со скоростью 1.

$$\varphi(3, 0^c, r) = 3, r \in Q;$$

$$\varphi(0, 3, 1^c) = 4;$$

$$\varphi(3, 1^c, r) = 0^c, r \in Q;$$

$$\varphi(0, 4, 0^c) = 0;$$

$$\varphi(4, 0^c, r) = 3, r \in Q.$$

Ненулевая часть входной последовательности будет равняться $(0^c, 0^c, 3)$ (элементы последовательности слева направо подаются на свободный вход экрана). Предобработка экрана будет состоять из элементов $0^c, 1^c, 2^c$. Состояния 3 и 4 являются метками. Метка 3 движется на одну клетку вправо каждый такт, пока соседняя справа от нее клетка находится в состоянии 0^c . Сигналом для остановки метки 3 является состояние 1^c в клетке справа от метки. В этом случае метка переходит в состояние 4, в котором, согласно построенной функции переходов, находится один такт. Затем метка 4 исчезает, а клетка справа переходит в состояние 3. Итак, ясно, что скорость движения зависит от того, как часто перед меткой возникает состояние 1^c .

Рассмотрим теперь предобработку экрана. Содержательно, в предобработке экрана записано почти бинарное число из 0^c и 1^c с некоторыми переполненными

разрядами (в них стоит 2^c). Слева у этого числа самый старший разряд, справа — самый младший. Каждый такт вся запись числа сдвигается на одну клетку вправо. Кроме того, каждые два из трех тактов к младшему разряду этого числа прибавляется по единице. Разряд остается переполненным ровно один такт, а в следующий такт он принимает значение 0^c , а значение следующего за ним старшего разряда увеличивается на единицу. Как только переполненным становится самый старший (самый левый) разряд, длина предобработки увеличивается на единицу, и тем самым метка получает сигнал к остановке.

На рисунке 3.4 показаны несколько начальных тактов движения по описанным правилам.

Итак, будем рассматривать конфигурацию предобработки экрана как число, записанное в двоичной системе с некоторыми переполненными разрядами. Посмотрим, как часто длина этого числа будет увеличиваться, а соответственно, метка будет останавливаться на один такт.

Как уже говорилось выше, за 3 такта записанное в предобработке число увеличивается на две единицы. Это означает, что новые двойки образуются в младшем разряде предобработки через три такта. Кроме того, разряд остается переполненным ровно один такт, а на следующий такт он обнуляется, и, возможно, становится переполненным следующий разряд. То есть, двойка в предобработке движется влево со скоростью 1 разряд в 1 такт (и в какой-то момент исчезает совсем). Значит, переполнения разряда больше чем на единицу не произойдет (не будет значений переполненных разрядов, превосходящих 2^c).

Для двоичной записи числа увеличение его длины на один разряд означает, что число увеличилось вдвое. Пусть на каком-то такте наше число в предобработке имеет длину $k \geq 2$, $k \in \mathbb{N}$, а в предыдущий такт имело длину $k - 1$. Ясно, что когда мы прибавим к этому числу 2^{k-1} единиц (на это потребуется $2^{k-1} + 2^{k-2}$ тактов), а также, когда последняя прибавленная единица

дойдет до старшего разряда (на это нужно еще k тактов), произойдет переполнение старшего разряда. То есть, переполнение произойдет не позже, чем через $2^{k-1} + 2^{k-2} + k$ тактов.

Кроме этого, то, что длина записанного в предобработке числа увеличилась с $k - 1$ до k , означает, что 2^{k-1} -я единица, прибавленная к записанному в предобработке числу, успела дойти до старшего разряда, а это заняло $k - 1$ такт. Значит, переполнение старшего разряда в следующий раз произойдет через $2^{k-1} + 2^{k-2} + k - (k - 1) = 2^{k-1} + 2^{k-2} + 1$.

Поскольку каждое переполнение разряда влечет за собой остановку движения метки на 1 такт, значит, когда в предобработке записано число из k разрядов, к этому моменту метка остановилась $k - 1$ раз, то есть, в законе движения встретился $k - 1$ ноль.

Итак, построен экран и на нем построено движение точки закону, который обозначим через F .

Посчитаем скорость движения точки по закону F : $v = \lim_{t \rightarrow \infty} \frac{\nu_t(F)}{t} = \lim_{k \rightarrow \infty} \frac{2^{k-1} + 2^{k-2} + C}{2^{k-1} + 2^{k-2} + (k-1) + C} = 1$, где $0 \leq C \leq 2^{k-1} + 2^{k-2}$.

Итак, построенные экран и автономный закон движения на нем удовлетворяют условиям леммы. ■

В доказательстве леммы 17 в предобработке находилось число в двоичной записи с некоторыми переполненными разрядами. Очевидно, что по аналогии можно использовать в предобработке число в q -ичной записи с переполненными разрядами для любого натурального q . Тем самым мы получим сразу целое множество экранов и законов движения на них, реализующих автономные движения со скоростью 1. Ясно, что с ростом q переполнение разрядов в отрезке предобработки будет происходить реже, чем при меньших q , то есть, движение будет в некотором смысле "быстрее" (при этом скорость движения, определенная выше, будет по-прежнему равняться единице).

Лемма 18. *Существует непериодический закон движения F , для которого выполнено $v(F) = 0$ и $\mathcal{U}(\{F\}) \neq \emptyset$.*

Доказательство. Идея построения экрана и закона движения, удовлетворяющих условиям леммы, та же, что и в доказательстве леммы 17. Модифицируем построенный алгоритм для работы экрана следующим образом. Сначала запустим на экран метку, которая будет оставаться на месте, пока не получит сигнал слева. Кроме того, слева от метки запустим q -ичное растущее число, причем его разряды будут расти слева направо, то есть слева будет самый младший разряд, а справа — самый старший. Число и метка будут отделены нулевым разрядом числа. Как только длина числа будет увеличиваться на единицу, то есть разделяющий разряд между числом и меткой станет равным единице, это будет сигналом для метки, что ей нужно сдвинуться на одну клетку вправо.

Ясно, что при таком алгоритме можно построить элементарный автомат для экрана, и количество состояний автомата будет зависеть от q . При этом, для такого движения справедливы оценки времени, необходимого для переполнения разрядов, полученные в предыдущем утверждении, то есть, скорость движения изображения по описанному алгоритму равна 0. ■

Пусть $0 < a < 1$, $a \in \mathbb{Q}$, причем $a = \frac{p}{q}$, $p, q \in \mathbb{N}$, тогда через F_a будем обозначать закон движения $(1^p 0^{q-p})^\infty$.

Лемма 19. *Если $0 < a < 1$, $a \in \mathbb{Q}$, то существует непериодический закон F , такой что $v(F) = a$ и $\mathcal{U}(\{F\}) \neq \emptyset$.*

Доказательство. Очевидно, что определенный выше закон F_a удовлетворяет условию $v(F) = a$. Также ясно, существует экран S_a , для которого $Q(S_a) = q$, на котором реализуется движение точки по закону F_a . Обозначим через Q_a множество состояний элементарного автомата экрана S_a , L_a — множество меток.

Выше был приведен класс непериодических законов движения со скоростью $v = 0$ и для каждого закона приведена идея построения экрана, на котором

реализуется движение точки по этому закону. Возьмем тот закон, в котором за основание числа в предобработке взято число 2 и обозначим его через F_0 . Экран, на котором реализуется движение по закону F_0 обозначим через S_0, Q_0 — множество состояний элементарного автомата экрана S_0, L_0 — множество меток.

Положим $Q_g = \{0, g\} \ g \notin Q_a \cup Q_0$.

Построим экран, на котором реализуется непериодическое движение точки со скоростью a .

Положим множество состояний элементарного автомата $Q = Q_a \times Q_0 \times Q_g$, множество меток $L = L_a \times Q_0 \times Q_g$.

Итак $(q_1, q_2, q_3) \in Q$ и можем запустить на экране независимо друг от друга в координате q_1 движение метки из L_a по закону F_a и в координате q_2 движение метки L_0 по закону F_0 .

При этом как только метка из L_0 будет двигаться вперед на одну клетку, она будет запускать в координате q_3 сигнал g , который каждый такт двигается вправо на одну клетку. Как только этот сигнал достигнет ячейки, содержащей метку из L_a , он на один такт заморозит ее счетчик, то есть, метка будет оставаться на месте на один такт дольше, чем при обычном периодическом движении. Поскольку в законе движения F_0 интервалы нулей между единицами увеличиваются, то тормозящий сигнал для меток из L_a будет посылатся все реже, а значит, движение меток из L_a будет непериодическим. Поскольку скорость движения меток из L_0 равна нулю, то скорость движения меток из L будет совпадать со скоростью движения меток из L_a и равняться a . ■

Из доказанных лемм следует теорема.

Теорема 19. *Для любого рационального числа $a, 0 \leq a \leq 1$, существует непериодический закон F_a , такой что $v(F_a) = a$ и $\mathcal{U}(\{F_a\}) \neq \emptyset$.*

Теорема 20. *Существует бесконечный экран $S(\mathcal{A})$, на котором реализуется*

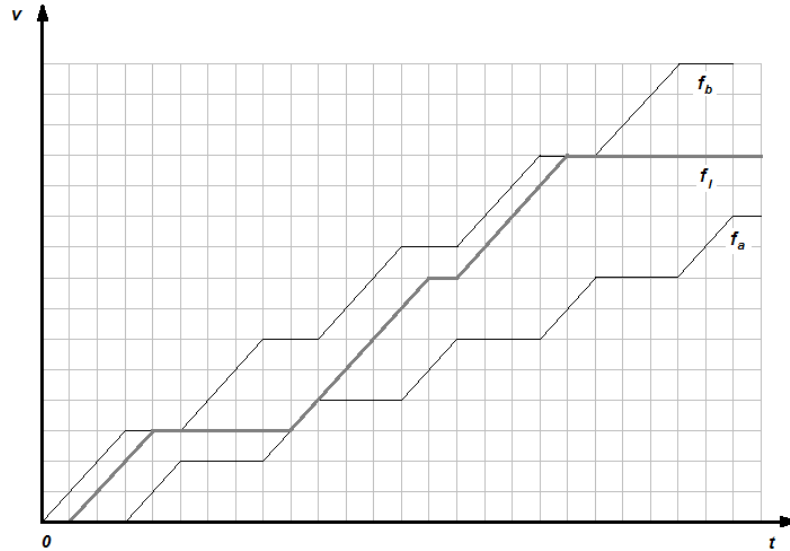


Рис. 3.5: Пример графиков законов движения

такой закон движения F изображения I^1 , что в F реализуются все скорости v из отрезка $[0, 1]$.

Доказательство. Пусть $a = \frac{p}{q}$, $b = \frac{r}{q}$ — рациональные числа и $a < b$. Ясно, что существуют экраны, на которых реализуются законы движения $F_a = (1^p 0^{q-p})^\infty$ и $F_b = (1^r 0^{q-r})^\infty$ изображения I^1 . Построим экран, на котором реализуются оба этих закона движения. Запустим на экране сигнал f_b , и через некоторое время сигнал f_a , движущиеся соответственно по законам F_b и F_a . Третий сигнал f_l запустим так, чтобы он двигался по закону $F_l = (1^s 0^{q-s})^\infty$, где $s > r$ (в частности, можно положить $F_l = 1^\infty$) до тех пор, пока не встретит сигнал f_b . Далее сигнал f_l стоит на месте до тех пор, пока его не догонит сигнал f_a . Таким образом, сигнал f_l всегда находится между сигналами f_a и f_b .

Ясно, что для любой последовательности $\{t_i\} \subseteq \mathbb{N}$ если существует предел $\lim_{t_i \rightarrow \infty} \frac{v_{t_i}(f_l)}{t_i} = v$, то $a \leq v \leq b$.

Изобразим на двумерной плоскости графики зависимости количества единиц от времени в законах движения сигналов f_a , f_b и f_l . На рисунке 3.5 изображен пример таких графиков.

Будем рассматривать точки пересечения графиков, причем, если координаты

точки пересечения оказываются не целыми, будем округлять их до целых.

По аналогии с тем, как строили законы движения f_a и f_b , для $c \in \mathbb{Q}$, $a < c < b$ построим закон f_c и его график зависимости количества единиц от времени. Наконец, в качестве последовательности $\{t_i^c\}$ возьмем моменты времени, когда графики f_l и f_c пересекаются.

Поскольку по условию $c \in (a, b)$, то график закона f_c начиная с некоторого момента будет всегда находиться между графиками f_a и f_b . Кроме того, график закона f_l также находится между графиками f_a и f_b , и он достигает каждого из граничных графиков поочередно бесконечное число раз. Таким образом, график f_c будет пересекаться с f_l также бесконечное число раз.

Ясно, что для построенной последовательности $\{t_i\} = \{t_i^c\}$ $\lim_{t_i \rightarrow \infty} \frac{\nu_{t_i}(f_l)}{t_i} = \lim_{t_i \rightarrow \infty} \frac{\nu_{t_i}(f_c)}{t_i} = c$. Это означает, что все рациональные скорости реализуются в законе движения f_l .

Для $a < c < b$, $c \in \mathbb{R} \setminus \mathbb{Q}$, построим возрастающую последовательность рациональных чисел $\{c_j\}$, сходящуюся к c . Далее, для каждой $\{c_j\}$ построим последовательность $\{t_i^{c_j}\}$.

Зафиксируем $\varepsilon > 0$. Ясно, что для этого ε существует $N > 0$, такое что для любого $n \geq N$ выполнено $|c - c_n| < \varepsilon/2$ и существует $M > 0$, такое что для любого $m \geq M$ выполнено $|\frac{\nu_{t_m}(f_{c_n})}{t_m} - c_n| < \varepsilon/2$.

Далее, для любого $i \in \mathbb{N}$ положим $\varepsilon_i = 1/2^i$. Для каждого $\varepsilon_i = 1/2^i$ найдем соответствующие указанным выше условиям N_i и M_i .

Положим $t_1^c = t_{M_1}^{c_{N_1}}$, $t_i^c = t_{M_i}^{c_{N_i}}$ и так далее.

Очевидно, что $\lim_{t_i^c \rightarrow \infty} \frac{\nu_{t_i}(f_l)}{t_i} = c$. Таким образом, на построенном экране и в построенном законе движения на нем реализуются все скорости c для любого $c \in (a, b)$.

Возьмем теперь в качестве сигнала f_a построенный выше сигнал f_0 , движущийся по непериодическому закону F_0 , для которого $v(F_0) = 0$, в качестве

сигнала f_b возьмем построенный выше сигнал f_1 , движущийся по непериодическому закону F_1 , для которого $v(F_1) = 1$, а в качестве сигнала f_l возьмем сигнал, который двигается по закону $F_l = 1^\infty$ до тех пор, пока не встретит сигнал f_1 . Далее сигнал f_l стоит на месте до тех пор, пока его не догонит сигнал f_0 . Таким образом, сигнал f_l всегда находится между сигналами f_0 и f_1 и все сказанное выше применимо к построенному сигналу f_l . Значит, на этом экране построено движение изображения по закону F_l , в котором реализуются все скорости v , где $0 \leq v \leq 1$. ■

Следствие 1. *Существует автономный бесконечный экран, реализующий для изображения Γ^1 закон движения с растущим числом идущих подряд нулей и растущим числом идущих подряд единиц.*

Результаты данного раздела анонсированы в [24] и опубликованы в [14].

Заключение

В диссертации получены следующие основные результаты.

Получено точное значение числа состояний элементарного автомата минимального универсального экрана. Для экрана, у которого длина хотя бы одной из сторон равна единице, это значение равно двум; для экрана, длина любой стороны которого больше единицы, это значение равно трем.

Показано, что не существует универсального элементарного автомата с двумя состояниями, и существует универсальный элементарный автомат с тремя состояниями. Построен ряд алгоритмов конструирования изображений на плоском экране с разным соотношением числа состояний и времени построения изображений. Доказано, что минимальное время построения любого изображения на единицу больше длины минимальной из сторон экрана.

Получены верхние оценки времени построения любого изображения при растущем числе состояний элементарного автомата. Для экрана с тремя состояниями элементарного автомата время построения изображения не превосходит суммы удвоенной длины меньшей из сторон экрана, длины большей стороны экрана и единицы. Для экрана с четырьмя состояниями элементарного автомата время построения изображения не превосходит суммы удвоенной площади экрана и единицы. Далее, при увеличении числа состояний элементарного экрана на два, оценка времени построения изображений уменьшается на единицу. При этом нижняя оценка времени построения изображений достигается на экране, число состояний элементарного автомата которого на два больше

удвоенной длины минимальной из сторон экрана.

Построен универсальный экран с одним управляющим входом и получена оценка времени построения изображения на этом экране. Если элементарный автомат имеет 8 состояний, то время построения изображения на экране не больше удвоенной площади экрана плюс 4.

Найденные способы построения изображений расширены для случая многомерного экрана, и получены оценки времени построения изображений на многомерном экране в зависимости от числа состояний элементарного автомата.

В работе построены управляющие автоматы, генерирующие входные последовательности для управляющих входов разработанных универсальных экранов. Получены оценки сложности управляющего автомата при условии, что входные последовательности для этого автомата получаются из кода изображения при помощи стандартных элементов для обработки потоков данных, таких как перестановка; устройство, которое собирает отдельные элементы в вектор из элементов; устройство, которое вставляет между элементами последовательности вспомогательные элементы; устройство, которое вставляет в начало последовательности заданное количество вспомогательных элементов. Показано, что в такой интерпретации сложность (число состояний) управляющего автомата равна единице для построенного алгоритма с четырьмя состояниями и с восемью состояниями (для экрана с одним входом). Для универсального автомата с тремя состояниями управляющий автомат имеет два состояния. Наконец, для универсального экрана с минимальным временем построения изображений число состояний управляющего автомата равно длине меньшей из сторон экрана.

Получены следующие результаты для экранов, реализующих движение изображений.

Получена точная оценка минимального числа состояний экрана, универсаль-

ного для одной точки и класса законов движения, в которых точка не двигается более одного такта подряд. При этом элементарный автомат имеет 4 состояния.

Построен универсальный конечный экран, на котором может быть реализовано любое поступательное движение точки. Показано, что минимальное число состояний универсального конечного экрана не может быть меньше четырех и больше семи.

Для любого натурального k существует такой элементарный автомат \mathcal{A} с 7^k состояниями, что для любого натурального m экран $S(\mathcal{A}, m)$ является универсальным для любого изображения I , состоящего из не более чем k точек.

Для многоточечных изображений получен способ построения экрана для движения $k > 1$ точек, для движения отрезка и для подмножества всех изображений, состоящих из k отрезков.

Для случая движения на бесконечных экранах получены следующие результаты.

Для любого бесконечного экрана найден такой закон движения, что на этом экране невозможно реализовать движение точки по этому закону.

Показано, что существует такой закон движения, что движение точки по этому закону невозможно реализовать ни на каком экране.

Для законов движения, в которых не встречается более чем s единиц подряд, минимальное число состояний универсального экрана не превосходит $2s + 2$. Если более чем s единиц подряд не встречается в законе движения с момента d , то минимальное число состояний универсального экрана не превосходит $2s + 9$.

Для автономных движений показано, что для любой рациональной скорости a , $0 \leq a \leq 1$ существует непериодический закон движения со скоростью a и существует экран, на котором реализуется движение точки по этому закону.

Построен бесконечный экран, на котором реализуется такой закон движения одной точки, что в этом законе реализуются все скорости из отрезка $[0, 1]$.

Показано, что существует автономный бесконечный экран, реализующий для одной точки закон движения с растущим числом идущих подряд нулей и растущим числом идущих подряд единиц.

Для автономных движений на бесконечном экране открыт вопрос поиска классов нереализуемых движений. Существует предположение, что описание класса реализуемых законов движения равносильно получению класса легко (быстро) вычисляемых последовательностей.

Литература

- [1] Болотов, А. А. О задачах сводимости и выразимости для однородных структур со входами и выходами / А. А. Болотов // ДАН СССР Т.254. — 1980. — N1 — С. 14.
- [2] Кудрявцев, В. Б. О мощности множеств предполных множеств некоторых функциональных систем, связанных с автоматами / В. Б. Кудрявцев // Проблемы кибернетики. — 1965. — вып. 13 — С. 45.
- [3] Кудрявцев, В. Б. Функциональные системы / В. Б. Кудрявцев. — Москва: Изд-во МГУ, 1982. — 158 с.
- [4] Кудрявцев, В.Б. Введение в теорию автоматов / В. Б. Кудрявцев, С. В. Алешин, А. С. Подколзин. — Москва: "Наука", 1985. — 320 с.
- [5] Кудрявцев, В.Б. Основы теории однородных структур / В. Б. Кудрявцев, А. С. Подколзин, А. А. Болотов. — Москва: "Наука", 1990. — 296 с.
- [6] Мур, Э. Ф. Математические модели самовоспроизведения / Э. Ф. Мур // Математические проблемы в биологии. — 1966. — С. 36
- [7] Нейман Дж., фон Теория самовоспроизводящихся автоматов / Дж. фон Нейман — Москва: Мир, 1971. — 384 с.
- [8] Подколзин, А. С. Об универсальных однородных структурах / А. С. Подколзин // Проблемы кибернетики. — 1975. — С. 199.

- [9] Подколзин, А. С. О сложности моделирования в однородных структурах / А. С. Подколзин // Проблемы кибернетики, вып. 34. — 1978. — С. 109.
- [10] Waksman, Abraham An Optimum Solution to the Firing Squad Synchronization Problem / Abraham Waksman // Information and control 9. — 1966. — С. 66.

Публикации автора по теме диссертации

- [11] Титова, Е. Е. Конструирование изображений клеточными автоматами / Е. Е. Титова // Интеллектуальные системы. — 2008. — т. 12. — С. 105
- [12] Титова, Е. Е. Линейное по времени конструирование изображений клеточными автоматами / Е. Е. Титова // Интеллектуальные системы. — 2012 — т. 16. — С. 215.
- [13] Титова, Е. Е. Сложность конструирования изображений клеточными автоматами / Е. Е. Титова // Интеллектуальные системы. — 2013. — т. 17 — С. 191.
- [14] Титова, Е. Е. Конструирование движущихся изображений клеточными автоматами / Е. Е. Титова // Интеллектуальные системы. — 2014. — т. 18. — С. 153.
- [15] Титова, Е. Е. Конструирование изображений клеточными автоматами / Е. Е. Титова // Материалы международной конференции "Современные проблемы математики, механики и их приложений", посвященная 70-летию академика В.А.Садовниченко. — 2009. — С. 376
- [16] Титова, Е. Е. Зависимость времени конструирования изображений от числа состояний клеточного автомата / Е. Е. Титова // Тезисы докладов

международной конференции студентов, аспирантов и молодых ученых «Ломоносов-2009» — 2009.

- [17] Титова, Е. Е. О времени конструирования изображений клеточными автоматами / Е. Е. Титова // Тезисы докладов I Международная научно-практическая конференция "Интеллектуальные машины" 9-10 апреля 2009 г. — 2009.
- [18] Титова, Е. Е. О линейном по времени конструировании изображений клеточным автоматом с тремя состояниями / Е. Е. Титова // Тезисы докладов международного семинара "Дискретная математика-2010". — 2010.
- [19] Титова, Е. Е. Сложность конструирования изображений клеточными автоматами / Е. Е. Титова // Материалы X международной конференции "Интеллектуальные системы и компьютерные науки". — 2011.
- [20] Титова, Е. Е. О времени конструирования изображений клеточными автоматами / Е. Е. Титова // Тезисы докладов международной конференции студентов, аспирантов и молодых ученых «Ломоносов-2011». — 2011.
- [21] Титова, Е. Е. Сложность управляющего автомата для построения изображений на универсальном экране / Е. Е. Титова // Тезисы докладов международной конференции студентов, аспирантов и молодых ученых «Ломоносов-2012». — 2012.
- [22] Титова, Е. Е. Конструирование движущихся изображений клеточными автоматами / Е. Е. Титова // Материалы XI Международного семинара "Дискретная математика и ее приложения", посвященного 80-летию со дня рождения академика О.Б.Лупанова. — 2012.
- [23] Титова, Е. Е. Конструирование движущихся изображений клеточными ав-

томатами / Е. Е. Титова // Тезисы докладов международной конференции студентов, аспирантов и молодых ученых «Ломоносов-2013». — 2013.

- [24] Титова, Е. Е. Конструирование движущихся изображений на бесконечных экранах / Е. Е. Титова // Тезисы докладов международной конференции студентов, аспирантов и молодых ученых «Ломоносов-2014». — 2014.