

# О представлении плоских конфигураций в однородных структурах\*

В.Е. Владиславлев

Рассматривается вопрос о сложности представления конфигураций в однородных структурах (ОС), то есть, каким минимальным числом состояний должна обладать ОС для того, чтобы из некоторой инициальной (одноклеточной) конфигурации в ней можно было вырастить заданную конфигурацию. Показано, что сложность представления почти любой плоской бинарной конфигурации  $\mathcal{K}$  асимптотически равна  $l^{-1}(2^{d^2})$ , где  $d$  — диаметр  $\mathcal{K}$  в метрике  $l_{\max}$ , а  $l(x) = x^{x^5}$ .

## 1. Введение

В этой работе мы рассматриваем понятие *представления* изображений в однородных структурах (ОС). Представимость изображения в некоторой ОС  $\sigma$  — это способность  $\sigma$  вырастить это изображение, точнее конфигурацию, соответствующую этому изображению из некоторой одноклеточной конфигурации. Под сложностью такого представления мы будем понимать число состояний ОС  $\sigma$ , необходимых для этого выращивания.

Для получения каких-либо оценок сложности представления нам необходимо приписать каждой конфигурации некоторую числовую характеристику. В качестве такой характеристики был выбран диаметр в  $l_{\max}$ , то есть длина стороны минимального квадрата, содержа-

---

\*Работа выполнена при частичной поддержке РФФИ, грант № 02-01-00162.

шего эту конфигурацию. Теперь, мы можем рассмотреть класс конфигураций фиксированного диаметра  $d$  и ввести шеннонову сложность  $L(d)$  представления выбранного класса.

Для указанного функционала найдена асимптотика (с точностью до аддитивной константы) и показано, что почти все конфигурации имеют такую сложность представления.

## 2. Необходимые понятия и постановка задачи

*Однородной структурой* (ОС)  $\sigma$  будем называть набор  $\sigma = \langle \mathbb{Z}^k, E_n, V, \varphi \rangle$ , где  $\mathbb{Z}^k$  — множество  $k$ -мерных векторов с целыми координатами,  $E_n = \{0, 1, \dots, n-1\}$ ,  $V = \{\alpha_1, \alpha_2, \dots, \alpha_{h-1}\}$  — упорядоченный набор различных векторов из  $\mathbb{Z}^k$ ,  $\varphi: (E_n)^h \rightarrow E_n$ , причем  $\varphi(0, 0, \dots, 0) = 0$ . Элементы множества  $\mathbb{Z}^k$  называются ячейками ОС  $\sigma$ , элементы  $E_n$  — состояниями ячеек; состояние 0 мы также будем называть состоянием покоя. Набор  $V$  называется шаблоном соседства ОС  $\sigma$  и для всякой ячейки  $\alpha$  однородной структуры определяет ее окрестность:  $V(\alpha) = \{\alpha, \alpha + \alpha_1, \dots, \alpha + \alpha_{h-1}\}$ . Функция  $\varphi$  называется локальной функцией переходов ОС  $\sigma$ : если  $x_0, x_1, \dots, x_{h-1}$  — соответствующие состояния ячеек из  $V(\alpha)$  в некоторый момент времени  $t$ , то  $\varphi(x_0, x_1, \dots, x_{h-1})$  — состояние ячейки  $\alpha$  в момент времени  $t + 1$ .

Приписывая каждой ячейке ОС  $\sigma$  состояния из множества  $E_n$ , получаем *состояние однородной структуры*. Другими словами, состоянием однородной структуры  $\sigma$  называется функция  $f: \mathbb{Z}^k \rightarrow E_n$ .

*Основной функцией переходов* ОС  $\sigma$  называется функция  $\Phi$ , определенная на множестве всех состояний ОС, принимающая значения в этом же множестве и удовлетворяющая следующему соотношению

$$g = \Phi(f) \Leftrightarrow g(\alpha) = \varphi(f(\alpha), f(\alpha + \alpha_1), \dots, f(\alpha + \alpha_{h-1})), \forall \alpha \in \mathbb{Z}^k,$$

где  $\varphi$  — локальная функция переходов ОС  $\sigma$ .

*Функционированием* ОС называется последовательность  $f_0, f_1, \dots, f_n, \dots$  ее состояний, таких, что  $f_i = \Phi(f_{i-1}), \forall i \in \mathbb{N}$ , причем  $f_0$  — интерпретируется как начальное (заданное извне) состояние ОС.

Состояние  $f$  ОС  $\sigma$ , для которого соотношение  $f(\alpha) \neq 0$  выполнено не более, чем для конечного числа ячеек  $\alpha \in \mathbb{Z}^k$ , называется *конфигурацией* ОС  $\sigma$ . Для удобства под *конфигурацией* мы будем также понимать конечное множество  $\mathcal{K}$  ячеек ОС, находящихся в ненулевом состоянии,  $\mathcal{K} = \{\alpha \in \mathbb{Z}^k \mid f(\alpha) \neq 0\}$ .

Пусть  $\Phi$  — основная функция переходов ОС  $\sigma$ . Если существует такое  $t \in \mathbb{N}$ , что  $\Phi^t(f) = \Phi^{t+1}(f) = g$ , то говорят, что ОС  $\sigma$  *выращивает* конфигурацию  $g$  из конфигурации  $f$  (отметим, что равенство здесь понимается как повекторное совпадение, а не существование параллельного переноса). Минимальное  $t$ , удовлетворяющее указанному соотношению, называется временем выращивания  $g$  из  $f$  в ОС  $\sigma$ .

Если в определении однородной структуры  $k = 2$ , то такая однородная структура называется *плоской*. Если для любой ячейки  $\alpha$ , для которой  $f(\alpha) \neq 0$ , верно, что  $f(\alpha) = 1$ , то конфигурация  $f$  называется *бинарной*. Если конфигурация  $f$  такова, что  $f(\alpha) \neq 0$ , верно лишь для единственной ячейки  $\alpha$ , то  $f$  называется *инициальной*. Если шаблон соседства плоской ОС имеет вид  $V = \{(1, 0), (0, 1), (-1, 0), (0, -1)\}$ , то он называется *шаблоном соседства Мура*; будем обозначать его через  $V_+$ . Класс всех плоских однородных структур с шаблоном соседства Мура и алфавитом состояний, содержащим не более  $n$  элементов, обозначим через

$$H_+(n) = \{\sigma \mid \sigma = \langle \mathbb{Z}^2, E_m, V_+, \varphi \rangle, m \leq n\}.$$

Далее мы будем рассматривать только ОС из указанного класса.

Конфигурация  $\mathcal{K}$  называется *представимой в ОС  $\sigma$  с помощью  $q_0 \in E_m$* , если  $\sigma$  выращивает  $\mathcal{K}$  из инициальной конфигурации, у которой единственная ненулевая ячейка имеет состояние  $q_0 \in E_m$ . Обозначим это как  $(\sigma, q_0) \rightsquigarrow \mathcal{K}$ . *Сложностью* такого представления конфигурации  $\mathcal{K}$  назовем число состояний ОС  $\sigma$  и будем обозначать через  $L_{(\sigma, q_0)}(\mathcal{K})$ .

Минимальную сложность представления  $\mathcal{K}$  по всем парам  $(\sigma, q_0)$  назовем *сложностью представления конфигурации  $\mathcal{K}$*  и обозначим через

$$L(\mathcal{K}) = \min_{(\sigma, q_0) \rightsquigarrow \mathcal{K}} \{L_{(\sigma, q_0)}(\mathcal{K})\}.$$

Пусть теперь  $C(d)$  — некоторый конечный класс конфигураций,

который характеризуется числом  $d \in \mathbb{N}$ . Сложностью представления класса  $C(d)$  назовем максимальную из сложностей представления его элементов и обозначим как  $L_C(d)$ .

$$L_C(d) = \max_{\mathcal{K} \in C(d)} \{L(\mathcal{K})\}.$$

Введем еще несколько обозначений. Через  $\text{diam}(\mathcal{K})$  будем обозначать диаметр конфигурации  $\mathcal{K}$  в метрике  $l_{\max}(\alpha, \beta) = \max(|\alpha_1 - \beta_1|, |\alpha_2 - \beta_2|)$ , где  $\alpha = (\alpha_1, \alpha_2)$ ,  $\beta = (\beta_1, \beta_2)$ . Множество всех плоских бинарных конфигураций диаметра  $d$  обозначим через  $D(d) = \{\mathcal{K} | \text{diam}(\mathcal{K}) = d\}$ . Через  $l(x)$  обозначим число всех функции  $x$ -значной логики от 5-и переменных  $l(x) = x^{x^5}$ .

Предложенное выше описание прозвояет нам рассматривать однородные структуры как аналог управляющей системы. Таким образом, введенная нами функция  $L_C(d)$  как сложность представления класса конфигураций  $C(d)$  является аналогом шенноновой сложности для управляющих систем. Первым вопросом, возникающим в связи с введенными определениями, является вопрос об асимптотическом поведении выбранного функционала сложности  $L_C(d)$ . Если при изучении какой-либо управляющей системы класс функций алгебры логики, который мы выбираем для представления, составляют функции, зависящие от  $d$  переменных, то для представления конфигураций мы выбрали класс  $D(d)$ .

Следующая часть работы посвящена получению нижней оценки сложности представления произвольного конечного класса конфигураций.

### 3. Нижняя оценка сложности представления

Для начала определим мощность класса плоских бинарных конфигураций диаметра  $d$ . Верно следующее утверждение.

**Лемма 1.** При  $d \rightarrow \infty$  справедливо соотношение  $|D(d)| \sim 2^{d^2}$ .

**Доказательство.** Из определения  $\text{diam}(\mathcal{K})$  следует, что всякая конфигурация диаметра  $d$  содержится в некотором квадрате со сторо-

ной  $d$ . Так как в теории однородных структур конфигурации рассматриваются с точностью до сдвига, то можно считать, что все конфигурации лежат в одном и том же квадрате  $Q$ . Отсюда следует, что мощность  $D(d)$  не больше, чем число (бинарных) состояний квадрата  $Q$ , то есть  $|D(d)| \leq 2^{d^2}$ .

Так как нас интересует асимптотическое поведение  $|D(d)|$ , будем считать, что  $d \geq 2$ . Разделим квадрат  $Q$  на 5 непересекающихся частей: центральная часть — квадрат со стороной  $d - 2$  в середине исходного квадрата, и четыре периферийных части — полосы единичной ширины и длины  $d - 1$ , которые составляют границу квадрата  $Q$ . Рассмотрим такие конфигурации в  $Q$ , ограничение каждой из которых на любую периферийную часть содержит не менее одной ячейки в единичном состоянии. Ограничение этих конфигураций на центральную часть может быть любым. Все эти конфигурации будут различны, то есть никакая из них не может быть получена из другой с помощью параллельного переноса, и диаметр каждой будет равен  $d$ . Оценим снизу мощность  $D(d)$  числом конфигураций описанного вида. Каждая периферийная часть может находиться в любом состоянии, кроме нулевого. Число таких состояний равно  $2^{d-1} - 1$ . Центральная часть может находиться в любом состоянии; их число есть  $2^{(d-2)^2}$ . Перемножив число состояний каждой из пяти областей, получим нижнюю оценку для мощности  $D(d)$ :

$$|D(d)| \geq 2^{(d-2)^2} \cdot (2^{d-1} - 1)^4 \sim 2^{d^2} + o(2^{d^2}), \quad d \rightarrow \infty,$$

откуда, в совокупности с верхней оценкой, следует утверждение леммы.

Прежде, чем перейти к доказательству нижней оценки, сделаем следующее предложение.

**Предложение 1.** *Если конфигурация  $\mathcal{K}$  представима в некоторой ОС  $\sigma$  с помощью одного из своих состояний, а число всех ее состояний равно  $n_0$ , то для всякого  $n$ , большего  $n_0$ , существует ОС  $\sigma'$  с числом состояний  $n$ , которая также представляет  $\mathcal{K}$  с помощью некоторого своего состояния, то есть*

$$\begin{aligned} ((\sigma \in H_+(n_0)) \& ((\sigma, q_0) \rightsquigarrow \mathcal{K})) \Rightarrow \\ \Rightarrow (\forall n \geq n_0 (\exists \sigma' \in H_+(n)) \& ((\sigma', q_0) \rightsquigarrow \mathcal{K})). \end{aligned}$$

**Доказательство.** В самом деле, при  $n \geq n_0$  рассмотрим такую ОС  $\sigma' \in H_+(n)$ , локальная функция переходов которой на любом наборе состояний  $(\alpha_1, \dots, \alpha_5) \in E_{n_0}^5$  совпадает с функцией переходов ОС  $\sigma$ , а на остальных наборах принимает произвольное значение. Инициальную конфигурацию для представления  $\mathcal{K}$  в  $\sigma'$  возьмем ту же, что и для представления  $\mathcal{K}$  в  $\sigma$ . Очевидно, что выбранная таким образом ОС  $\sigma'$  будет представлять заданную конфигурацию  $\mathcal{K}$ . Предложение установлено.

Теперь докажем мощностную нижнюю оценку для функционала сложности представления произвольного конечного класса плоских конфигураций.

**Лемма 2.** *Сложность представления  $L_C$  произвольного конечного класса  $C$  плоских конфигураций не меньше, чем  $l^{-1}(|C|)$ .*

**Доказательство.** Так как сложность представления класса  $C$  является, по определению, максимальной сложностью представления его конфигураций (обозначим ее через  $n$ ), то, в силу сделанного выше предложения, можно считать, что все конфигурации из  $C$  представимы однородными структурами из  $H_+(n)$ . Каждая ОС из  $H_+(n)$  может представлять не более  $(n - 1)$  различных конфигураций; ровно столько существует различных инициальных состояний. Так как ОС из  $H_+(n)$  однозначно задаются локальными функциями переходов, мощность множества  $H_+(n)$  равна числу всех функций  $n$ -значной логики от пяти переменных, сохраняющих 0; тем самым  $|H_+(n)| = n^{(n-1)^5}$ . Следовательно, для представления класса  $C$  необходимо выбрать такое  $n$ , чтобы выполнялось неравенство

$$|C| \leq (n - 1) \cdot |H_+(n)| \leq l(n).$$

Так как функция  $l$  монотонно возрастает, обратная к ней  $l^{-1}$  также монотонно возрастает. Взяв от обеих частей неравенства  $l^{-1}$ , получаем:

$$l^{-1}(|C|) \leq n.$$

**Следствие 1.** *Так как доказанная оценка является необходимым условием представления класса  $C$ , то справедливо, что*

$$n - 1 \leq l^{-1}(|C|).$$

**Лемма 3.** Пусть, теперь, рассматриваемый класс конфигураций  $C$  может быть параметризован натуральным числом  $d$ , то есть  $C = C(d)$ . Причем  $|C(d)| = g(d)$ , где  $g(d)$  — монотонно возрастающая, неограниченная функция. И пусть  $C'(d) = \{\mathcal{K} \in C(d) | L(\mathcal{K}) \leq L_C(d) - 2\}$ , состоит из тех конфигураций из  $C(d)$ , сложность представления которых меньше сложности представления всего класса  $C(d)$  хотя бы на два. Тогда верно следующее утверждение

$$\lim_{d \rightarrow \infty} \frac{|C'(d)|}{|C(d)|} = 0.$$

**Доказательство.** Очевидно, что  $0 \leq \frac{|C'(d)|}{|C(d)|}$ . Оценим это отношение сверху. Обозначим через  $n$  сложность представления класса  $C(d)$ . В силу доказанной леммы верно, что

$$n \geq l^{-1}(|C(d)|) = l^{-1}(g(d)).$$

Из монотонности и неограниченности функций  $g$  и  $l^{-1}$  следует, что  $n \rightarrow \infty$  при  $d \rightarrow \infty$ . Из определения класса  $C'(d)$  следует, что  $|C'(d)| \leq l(n - 2)$ . Тогда

$$\begin{aligned} \frac{|C'(d)|}{|C(d)|} &\leq \frac{l(n - 2)}{g(d)} = \frac{l(n - 2)}{l(l^{-1}(g(d)))} \leq \\ &\leq \frac{l(n - 2)}{l(n - 1)} = \left( \left( 1 - \frac{1}{n - 1} \right)^{n-1} \frac{1}{n - 2} \right)^5. \end{aligned}$$

Предел первого сомножителя в скобках есть  $e^{-1}$ , а второго — 0. Следовательно,  $\frac{|C'(d)|}{|C(d)|} \rightarrow 0$ , при  $d \rightarrow \infty$ .

Из доказанных лемм вытекает, что почти любая плоская бинарная конфигурация  $\mathcal{K}$  имеет сложность представления не меньше, чем  $l^{-1}(2^{d^2}) - 1$ , где  $d = \text{diam}(\mathcal{K})$ .

## 4. Асимптотически оптимальное представление плоских бинарных конфигураций

Эта часть работы посвящена описанию функционирования однородной структуры  $\sigma(\mathcal{K}) = \langle \mathbb{Z}^2, E_\sigma, V_+, \varphi_\sigma \rangle$ , которая построена по

некоторой (произвольной) плоской бинарной конфигурации  $\mathcal{K}$ . Эта ОС обладает тем свойством, что она представляет  $\mathcal{K}$ , и число ее состояний асимптотически равно  $l^{-1}(2^{d^2})$ , где  $d = \text{diam}(\mathcal{K})$ .

Пусть конфигурация  $\mathcal{K}$  такова, что минимальный прямоугольник, ее содержащий, имеет размер  $n_1 \times n_2$  ячеек. Из определения диаметра конфигурации следует, что  $\max(n_1, n_2) = d$ .

Начнем с описания структуры алфавита состояний  $E_\sigma$ . Разобьем  $E_\sigma$  на несколько непересекающихся подмножеств. Первое подмножество — множество *вспомогательных состояний*

$$Q = \{(x, y, t) \in \mathbb{N}_0^3 : x \leq 2, y \leq 3, t \leq 5\}.$$

Второе подмножество — цифры  $(m+1)$ -ричной системы исчисления

$$M = \{0_{m+1}, 1_{m+1}, \dots, m_{m+1}\}.$$

Третье подмножество

$$X = \{\#, \#', *, 0', 0'', \tilde{0}, 1', 1'', \tilde{1}, -1, 0'_{m+1}, \Delta, \tilde{\Delta}, \omega_0, \omega_1, \rightarrow, \downarrow, \delta_0, \delta_1\}$$

содержит так называемые *служебные состояния*. Наконец, последним подмножеством является множество  $E_2 = \{0, 1\}$  — это именно те состояния, которые принимают ячейки бинарной конфигурации  $\mathcal{K}$ .

Разобьем описание процесса выращивания конфигурации  $\mathcal{K}$  на несколько этапов.

#### 4.1. Построение полигона

*Полигоном* называется конфигурация, имеющая форму прямоугольника размером 4 клетки на 6, структура которой будет описана ниже. Пусть  $(0, 0, 5) \in Q$  — состояние единственной ненулевой ячейки инициальной конфигурации, из которой будет выращиваться финальная конфигурация  $\mathcal{K}$ . Последнюю компоненту векторов из  $Q$  будем называть *временем* и интерпретировать как число тактов, оставшееся до завершения очередного шага в процессе выращивания. На каждом такте эта компонента у всех ячеек, находящихся в состояниях из  $Q$ , будет совпадать (то есть ячейки во вспомогательных состояниях синхронизированы). Первые две компоненты состояний

из  $Q$  назовем «*координатами*». Они показывают расстояния по каждой из соответствующих осей координат от ячейки, находящейся в этом состоянии до ячейки в состоянии с нулевыми «координатами» (такая ячейка будет единственной). Заметим, что «координаты» не определяют ячейку однозначно. Например, ячеек в состоянии вида  $(1, 2, t)$  будет четыре. Но в совокупности с информацией о состоянии хотя бы одного (любого) соседа во вспомогательном состоянии ячейка определяется однозначно.

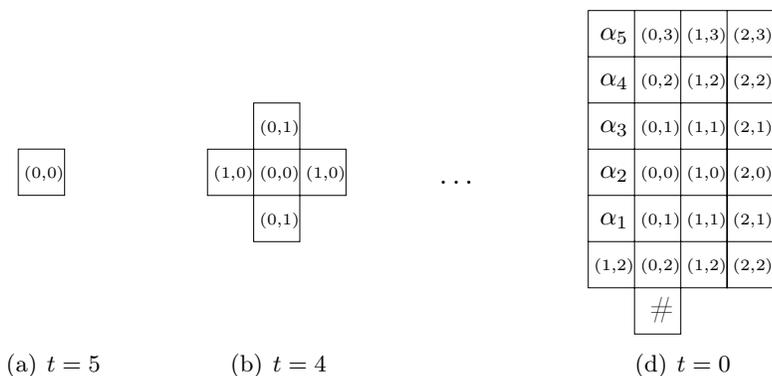


Рис. 1. Выращивание полигона

Всякая ячейка в нулевом состоянии, имеющая соседа (соседей) во вспомогательном состоянии, сама переходит во вспомогательное состояние (с соответствующими «координатами» и временем), если это не приводит к выходу за границы полигона; иначе она остается в нулевом состоянии. Ячейки во вспомогательных состояниях на каждом такте просто уменьшают свое время на единицу. На последнем (шестом) такте этого этапа некоторые ячейки вместо того, чтобы перейти в очередное вспомогательное состояние (с нулевым временем), переходят в состояния  $\alpha_i^0 \in M$ , где  $i \in \mathbb{N}, 1 \leq i \leq 5$ . Последний такт этого этапа показан на рисунке 1.

Ячейку с «координатами»  $(1, 2)$ , у которой только нижний сосед находится в нулевом состоянии, будем называть *выходом из полигона*. На последнем такте построения полигона, ячейка в нулевом состоянии, верхний сосед которой находится во вспомогательном состоянии

с «координатами»  $(0, 2)$ , переходит в служебное состояние  $\#$ . Ячейки в состояниях  $\alpha_i^0, i = 1, \dots, 5$  в совокупности представляют пятизначное число  $A^0$  в  $(m+1)$ -ричной системе исчисления, где  $\alpha_1^0$  младший разряд  $A^0$ , а  $\alpha_5^0$  — старший. Совокупность этих пяти ячеек далее будем называть *числовым блоком*. Левую нижнюю ячейку полигона назовем *генератором тактов*. В каждый следующий момент времени состояние этой ячейки отличается от ее предыдущего состояния только временем:  $t_{i+1} = t_i - 1 \pmod{6}$ . Когда время генератора тактов становится равно 0 — это является сигналом к началу очередного шага процесса (вычитания единицы).

## 4.2. Вычитание единицы

Опишем вспомогательный процесс *вычитания единицы* в числовом блоке, то есть выращивание в ОС  $\sigma_{\mathcal{X}}$  в числовом блоке из конфигурации, представляющей натуральное число  $A^i = (\alpha_5^i \alpha_4^i \alpha_3^i \alpha_2^i \alpha_1^i)_{m+1}$ , конфигурации, представляющей число  $A^{i+1} = A^i - 1 = (\alpha_5^{i+1} \alpha_4^{i+1} \alpha_3^{i+1} \alpha_2^{i+1} \alpha_1^{i+1})_{m+1}$ . Для этого нам понадобятся два служебных состояния:  $-1$  для обозначения заимствования из следующего разряда и  $0'_{m+1}$  для обозначения старших, но уже не значащих (обнуленных) разрядов.

Рассмотрим ячейку числового блока, находящуюся в состоянии  $\alpha$  из множества  $M$ , нижний сосед которой либо является генератором тактов и находится в состоянии с нулевым временем (то есть рассматриваемая ячейка представляет младший разряд числа  $A^i$ ), либо находится в служебном состоянии  $-1$  (то есть рассматриваемая ячейка представляет не младший разряд  $A^i$  и все меньшие разряды оказались равными  $0_{m+1}$ ). Тогда, если  $\alpha \notin \{0_{m+1}, 1_{m+1}\}$ , то на следующем такте ячейка переходит в состояние  $\alpha - 1 \in M$ .

Если же состояние рассматриваемой ячейки равно  $1_{m+1}$ , то ее состояние в следующем такте будет зависеть еще и от состояния ее верхнего соседа. Если сосед сверху находится либо в нулевом состоянии (то есть рассматриваемая ячейка представляет старший разряд (все еще пятиразрядного) числа  $A^i$ ), либо в служебном состоянии  $0'_{m+1}$  (то есть рассматриваемая ячейка представляет старший разряд  $A^i$ , разрядность которого уже меньше пяти), то на следующем

такте ячейка перейдет в служебное состояние  $0'_{m+1}$ . В противном случае ячейка примет состояние  $0_{m+1} \in M$ .

Наконец, если рассматриваемая ячейка находится в состоянии  $0_{m+1}$ , то на следующем такте ячейка примет служебное состояние  $-1$  (то есть будет осуществлено заимствование из следующего разряда). Ячейка, находящаяся в служебном состоянии  $-1$  на следующем такте перейдет в состояние  $m_{m+1}$  — максимальная цифра в  $(m+1)$ -ричной системе исчисления.

Если у ячейки-генератора тактов время равно нулю, а верхний сосед находится в служебном состоянии  $0'_{m+1}$  (то есть числовой блок представляет число 0), то генератор тактов переходит в служебное состояние  $\Delta$ . Более подробное описание того, что происходит с ячейками, у которых среди соседей есть ячейки в состоянии  $\Delta$ , будет дано ниже.

Заметим, что процесс вычитания единицы всегда занимает по времени не более шести тактов. Используя этот процесс как вспомогательный, опишем, как функционирует полигон.

### 4.3. Формирование новой цифры

Каждые 6 тактов в числовом блоке будет возникать новая конфигурация, кодирующая в  $(m+1)$ -ричной системе число, на единицу меньшее предыдущего. Так будет осуществлен перебор всех целых чисел от  $A^0$  до нуля. За эти 6 тактов все цифры текущего числа из числового блока будут перенесены в область генерации очередной цифры, как это показано на рисунке 2 (область генерации очередной цифры отмечена серым цветом). Когда ячейка и все ее соседи будут находиться в состояниях из множества  $M$ , однородная структура  $\sigma_{\mathcal{X}}$  переведет эту ячейку в новое состояние из множества  $M$ .

Через  $\beta' \in M$  на рисунке обозначена цифра, которая была получена на предыдущей итерации, а через  $\beta$  — цифра, полученная на текущей итерации. Вынос очередной сформированной цифры за пределы полигона требует 8 или 9 тактов, в зависимости от того, идет ли сейчас формирование границы  $\mathcal{X}$  или ее кода (об этом речь пойдет ниже). Но, как показано на рисунке, нехватяющие такты можно заимствовать у следующей итерации так, чтобы это не мешало про-

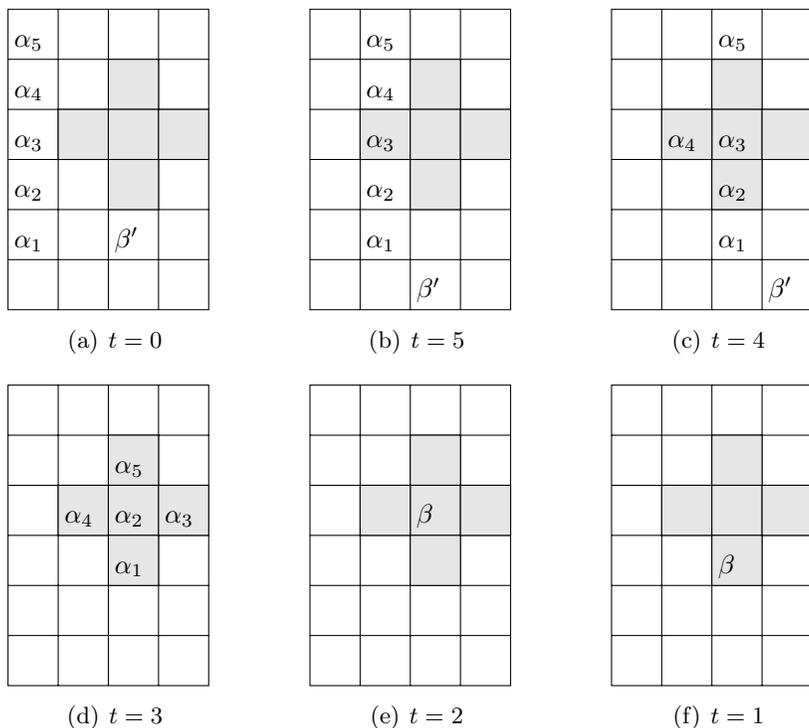


Рис. 2. Формирование новой цифры

цессу формирования очередной цифры.

Из приведенного рисунка видно, что на каждом такте у всякой ячейки в состоянии из множества  $M$ , за исключением той, в которой формируется новая цифра, есть сосед в служебном состоянии. Это позволяет ячейкам в состояниях из множества  $M$  корректно переходить обратно во вспомогательное состояние с текущим значением времени и имевшимися у них ранее «координатами».

#### 4.4. Формирование границ и кода конфигурации $\mathcal{K}$

Из приведенного выше описания функционирования ОС  $\sigma_{\mathcal{K}}$  следует, что ее локальная функция переходов  $\varphi_{\sigma(\mathcal{K})}$  сохраняет множество

М. Пусть  $\varphi_m = \varphi_{\sigma(K)} \Big|_{M^5}$  — ограничение функции  $\varphi_{\sigma(K)}$  на множество цифр.

Свяжем с функцией  $\varphi_m$  числовую функцию  $f_m$  (которую назовем *функцией формирования кода*), определенную на целых числах из полуинтервала  $[0, (m + 1)^{(m+1)^5})$  и принимающую целые значения из отрезка  $[0, m]$ :

$$f_m(x) = y \Leftrightarrow \varphi_m(x_1, x_2, \dots, x_5) = y,$$

где  $x = \overline{x_5x_4x_3x_2x_1}$  — запись числа в  $(m + 1)$ -ричной системе исчисления, в которой все разряды обязательно присутствуют (старшие, правда, могут быть нулевыми). Таким образом, задание функции  $f_m(x)$  влечет задание и функции  $\varphi_m$ .

Для кодирования представляемой конфигурации  $\mathcal{K}$  будем использовать  $m^5$ -значное число в  $m$ -ричной системе исчисления. Число таких кодов равно  $m^{m^5}$ . Так как начальное число  $A^0$ , представляемое числовым блоком в начальный момент времени не превышает  $(m + 1)^5 - 1$ , полигон может породить код длиной  $(m + 1)^5$ .

Положим

$$m = \lceil l^{-1}(2^{d^2}) \rceil.$$

Тогда верна следующая цепочка неравенств:

$$l^{-1}(2^{d^2}) \leq m \Rightarrow 2^{d^2} \leq m^{m^5} \Rightarrow d^2 < m^5 \cdot \log m \Rightarrow d^2 < m^6 \Rightarrow d < m^3.$$

А так как  $n_1 + n_2 + m^5 \leq 2 \cdot d + m^5 \leq 2m^3 + m^5 < (m + 1)^5$ , то полигон может выдать две последовательности длин  $n_1$  и  $n_2$ , которые кодируют длины сторон прямоугольника, содержащего  $\mathcal{K}$ , и последовательность длины  $m^5$ , которая некоторым образом кодирует саму конфигурацию  $\mathcal{K}$ .

Из сделанной оценки следует, что начальное значение  $A^0$  числового блока можно положить равным  $m^5 + n_1 + n_2 + 1$ .

Зададим функцию  $f_m$  следующим образом:

$$f_m(x) = \begin{cases} 1, & x \in [m^5 + n_1, m^5 + n_1 + n_2 + 1], \\ 0, & x \in (m^5, m^5 + n_1 - 1], \\ m, & x = m^5, \\ \neq m, & x \in [0, m^5 - 1]. \end{cases}$$

Последняя строка в определении  $f_m$  означает, что для указанных значений аргумента функция может принимать любое значение, за исключением  $m$ .

Теперь перейдем к рассмотрению того, как описанный механизм применяется для выращивания конфигурации  $\mathcal{K}$ . Сначала полигон  $n_2 + 2$  раза подряд формирует цифру  $1_{m+1}$ . Покинув пределы полигона, эта цифра будет двигаться вертикально вниз вдоль *стены* из ячеек в состоянии  $\#$  (напомним, что первая ячейка этой стены была сформирована сразу же по окончании выращивания полигона). Формально это можно описать так: ячейка в нулевом состоянии, у которой левый сосед находится в состоянии  $\#$ , а верхний — в состоянии  $1_{m+1}$ , на следующем такте сама перейдет в состояние  $1_{m+1}$ ; ячейка в состоянии  $1_{m+1}$ , у которой левый сосед имеет состояние  $\#$ , на следующем такте переходит в нулевое состояние. Ячейка, у которой верхний сосед имеет состояние  $1_{m+1}$ , а все остальные находятся в состоянии покоя, на следующем такте переходит в состояние  $\#$ . Ячейка, у которой верхний и правый соседи имеют состояние  $\#$ , сама на следующем такте примет состояние  $\#$ . А ячейка в состоянии  $\#$ , у которой левый сосед имеет такое же состояние, а все остальные находятся в состоянии покоя, сама переходит в состояние покоя. Таким образом каждая порожденная полигоном цифра  $1_{m+1}$  увеличивает длину стены на одну ячейку.

После этого полигон сформирует  $n_1 - 1$  раз цифры  $0_{m+1}$ , которые будут передаваться вниз вдоль построенной стены. Когда появится ячейка в нулевом состоянии, у которой лишь верхний сосед имеет ненулевое состояние  $0_{m+1}$ , то на следующем такте она примет состояние  $\#'$ . Далее, порождаемые полигоном цифры  $0_{m+1}$ , будут двигаться сначала вдоль вертикальной стены из ячеек в состоянии  $\#$ , а затем вдоль горизонтальной стены из ячеек в состоянии  $\#'$ , и каждая новая цифра  $0_{m+1}$  будет удлинять горизонтальную стену на одну ячейку, подобно тому, как это было с цифрой  $1_{m+1}$  в случае горизонтальной стены.

Затем полигон породит единственную цифру  $m_{m+1}$ . После того, как эта цифра выйдет за границу полигона, и расстояние между ней и полигоном станет равно 2, возникнет ячейка в нулевом состоянии, у которой левый сосед имеет состояние  $\#$ , нижний — состояние  $m_{m+1}$ ,

а верхний — является выходом из полигона, ячейка перейдет в состояние  $*$ . Это означает, что выход из полигона заблокирован. На следующем такте правый сосед этой ячейки (то есть ячейка, расположенная под правым нижним углом полигона) тоже перейдет в состояние  $*$ . Между двумя ячейками в состоянии  $*$  будет располагаться код конфигурации  $\mathcal{K}$ . Сама же цифра  $m_{m+1}$  проследует вниз до горизонтальной стены. Когда появится ячейка в нулевом состоянии, у которой левый сосед имеет состояние  $\#$ , нижний —  $\#'$ , а верхний —  $m_{m+1}$ , ячейка перейдет в состояние  $0_{m+1}$ . Цифра  $0_{m+1}$  проследует вдоль горизонтальной стены и увеличит ее длину еще на одну (последнюю) ячейку. На этом формирование границы выращаемой конфигурации будет закончено.

Перейдем к фазе формирования кода конфигурации  $\mathcal{K}$ . Теперь, когда выход из полигона перекрыт снизу ячейкой в состоянии  $*$ , цифры, порождаемые полигоном, будут выстраиваться правее этой ячейки. Опишем этот процесс детально. Ячейка, у которой левый сосед имеет состояние цифры, нижний — либо состояние  $*$ , либо тоже состояние цифры, а верхний — либо состояние покоя, либо вспомогательное состояние с «координатами»  $(2, 1)$ , переходит в состояние этой цифры (что соответствует перемещению новой цифры вдоль уже сформированной части кода). Ячейка в состоянии  $*$ , у которой слева либо цифра, либо  $*$ , а сверху цифра, принимает состояние верхнего соседа (что соответствует добавлению новой цифры в конец кода). Ячейка в нулевом состоянии, у которой правый сосед находится в состоянии цифры, а все остальные имеют нулевые состояния, переходит в состояние  $*$ .

Таким образом, у нас есть последовательность ячеек, чьи состояния представляют цифры  $m$ -ричной системы исчисления, которые в совокупности образуют число, являющееся кодом конфигурации  $\mathcal{K}$ . Если перевести это число в двоичную систему исчисления и уложить в прямоугольник с границей, заданной ячейками в состояниях  $\#$  и  $\#'$ , то получится конфигурация  $\mathcal{K}$ .

#### 4.5. Деление пополам

Для перевода полученного кода в двоичное представление, реализуем в ОС  $\sigma(\mathcal{K})$  процесс деления пополам. Остаток на каждом шаге

и будет очередной цифрой искомого двоичного представления.

Как упоминалось выше, в момент, когда числовой блок полигона представляет число 0, а к нему вновь применяется процесс вычитания единицы, то генератор тактов полигона переходит в состояние  $\Delta$ , которое является сигналом к началу процесса самоуничтожения полигона. Более точно: все ячейки во вспомогательных состояниях и в состоянии  $0'_{m+1}$ , имея соседа в состоянии  $\Delta$ , сами переходят в состояние  $\Delta$ . А из состояния  $\Delta$  ячейка переходит в нулевое состояние. Из этого правила есть два исключения. Первое: ячейка, чей левый сосед находится в состоянии  $*$ , не переходит в состояние  $\Delta$  (эта ячейка уже не принадлежит полигону, а представляет младший разряд кода конфигурации  $\mathcal{K}$ ). Второе исключение: ячейка, чей нижний сосед находится в состоянии  $\#$ . После того, как все соседи этой ячейки, кроме нижнего, станут нулевыми, она перейдет во вспомогательное состояние  $(2, 3, 0)$  и станет играть роль *тактового генератора*. Состояния этой ячейки будут меняться по следующему правилу: первая «координата» будет принимать по очереди значение 1 и 2. Каждый раз, когда значение первой «координаты» равно 1, время уменьшается на единицу (по модулю 6). (Заметим, что переиспользование вспомогательных состояний для создания такого генератора тактов не приведет к тому, что начнется новый рост части полигона.)

Итак, ячейка будет принимать состояние с нулевым временем каждые 12 тактов. Когда время этой ячейки равно нулю, ее правый сосед принимает служебное состояние  $\rightarrow$ , которое передается над кодом горизонтально вправо. Когда у ячейки в состоянии  $*$ , верхний сосед оказывается в состоянии  $\rightarrow$  (то есть сигнал дошел до правого края кода), а правый — в нулевом состоянии, она переходит в состояние  $\downarrow$ , которое на следующем такте передается нижнему соседу. Сама же ячейка на следующем такте возвращается в состояние  $*$ . Ячейка в состоянии  $\downarrow$ , у которой левый сосед имеет нулевое состояние, на следующем такте сама перейдет в нулевое состояние, а ее левый сосед из нулевого состояния перейдет в служебное состояние  $\delta_0$ .

Итак, после огибания стрелками  $m$ -ричного кода конфигурации  $\mathcal{K}$ , под старшим разрядом этого кода появляется ячейка в состоянии  $\delta_0$ . Это служит сигналом к началу очередного деления пополам. Опишем его.

Если верхний сосед ячейки, которая находится в состоянии  $\delta_0$ , содержит четную цифру, то ячейка переходит в служебное состояние  $\omega_0$  (это соответствует тому, что очередной разряд поделился на два). Левый сосед ячейки в состоянии  $\omega_0$  переходит в состояние  $\delta_0$ , а сама ячейка переходит в нулевое состояние. Если цифра над ячейкой в состоянии  $\delta_0$  окажется нечетной, то ячейка перейдет в состояние  $\omega_1$ . А ее левый сосед на следующем такте перейдет в состояние  $\delta_1$  (сама же ячейка из состояния  $\omega_1$  перейдет на следующем такте в нулевое состояние). Если ячейка находится в состоянии  $\alpha \in M$ , а ее нижний сосед находится в состоянии  $\delta_0$ , то ячейка переходит в состояние  $[\frac{\alpha}{2}] \in M$ . А если у ячейки в состоянии  $\alpha$  нижний сосед имеет состояние  $\delta_1$ , то ячейка переходит в состояние  $[\frac{\alpha+m}{2}] \in M$ . Это соответствует тому, что предыдущий (более старший) разряд кода  $\mathcal{K}$  не поделился нацело на 2.

Осталось рассмотреть граничные случаи этого деления. Первый — это уменьшение разрядности числа. Когда ячейка, представляющая старший разряд кода (та, у которой правый сосед имеет состояние \*), переходит в состояние  $0_{m+1}$ , то на следующем такте она принимает состояние \*, а ее правый сосед принимает нулевое состояние. То есть граничная ячейка кода сдвигается влево — навстречу очередному сигналу о начале деления. Но 12 тактов между последовательными сигналами хватает, чтобы это сокращение не привело к наложению сигналов.

Формирование  $\mathcal{K}$  происходит по слоям, начиная с нижнего. Заполнение каждого слоя в отдельности происходит справа налево.

Второй граничный случай — получение остатка от деления (который как раз и формирует состояние очередной ячейки выращаемой конфигурации  $\mathcal{K}$ ). Ячейка в нулевом состоянии, у которой верхний сосед находится в состоянии \*, левый — в состоянии  $\#$ , а правый — в состоянии  $\omega_0$  или  $\omega_1$ , принимает на следующем такте состояние правого соседа. Далее это состояние передается вниз вдоль вертикальной границы  $\mathcal{K}$ , а затем вдоль горизонтальной «стены» из ячеек либо в состоянии  $\#'$ , либо в состояниях  $0'$  и  $1'$ , если идет формирование не самого нижнего слоя конфигурации  $\mathcal{K}$ . Формирование самого нижнего слоя происходит следующим образом. Крайняя правая ячейка «стены», когда ее верхний сосед принимает состояние  $\omega_0$  или  $\omega_1$ , пе-

переходит с состояние  $0''$  или  $1''$ , соответственно. Остальные ячейки «стены», когда придет их время (то есть правый сосед будет находиться в одном из состояний  $\{0'', 1'', 0', 1'\}$ , а верхний —  $\omega_0$  или  $\omega_1$ ), переходят в состояния  $0'$  и  $1'$ , соответственно. Таким образом, произойдет замещение состояний всех ячеек в «стене», причем крайняя правая ячейка будет иметь состояние  $i''$ , а остальные  $j'$ . Формирование последующих слоев происходит так. Состояние  $\omega_i$  перемещается вдоль предыдущего слоя, пока нижняя ячейка не станет содержать состояние  $0''$  или  $1''$ . Тогда ячейка перейдет в состояние  $i''$ . Последующие состояния перемещаются вправо до тех пор, пока не встретят ячейку в ненулевом состоянии, а встретив, переходят в состояние  $i'$ .

Последним шагом процесса станет деление на два числа 1. Остатком этого деления, очевидно, станет цифра 1, которая сформирует последнюю ненулевую ячейку последнего слоя конфигурации  $\mathcal{K}$ . Этот самый верхний слой может оказаться заполненным не до конца (то есть оставшиеся ячейки в конфигурации  $\mathcal{K}$  имеют состояние 0). После того, как последний шаг сделан, две ячейки в состоянии \*, которые ограничивали  $m$ -ричный код, окажутся рядом. Появление очередного сигнала  $\rightarrow$  приведет к тому, что левая из двух ячеек в состоянии \* перейдет в состояние  $\tilde{\Delta}$ . Это сигнал к началу заключительного этапа представления  $\mathcal{K}$ .

## 4.6. Выравнивание

Конфигурацию, которая получилась к этому моменту, обозначим через  $\mathcal{K}'$ . Заметим, что она является связной. Построение  $\mathcal{K}'$  осуществлялось таким образом, что в состояниях  $1'$  и  $1''$  находятся те и только те ячейки, которые будучи переведенными в состояние 1 составят конфигурацию  $\mathcal{K}$ . Опишем процесс выращивания  $\mathcal{K}$  из  $\mathcal{K}'$ .

Если у ячейки в состоянии  $1'$  или  $1''$  появляется сосед в состоянии  $\tilde{1}$ ,  $\tilde{0}$  или  $\tilde{\Delta}$  (назовем эти состояния *выравнивающими*), то ячейка переходит в состояние  $\tilde{1}$ . Если ячейка находится либо в состоянии  $0'$ , либо в состоянии  $0''$  и хотя бы один из ее соседей имеет выравнивающее состояние, то ячейка переходит в состояние  $\tilde{0}$ . Из этих состояний ( $\tilde{0}$  и  $\tilde{1}$ ) ячейки переходят в состояния 0 и 1 соответственно, и более их не меняют. Если ячейка имеет ненулевое состояние, не упомяну-

тое выше и хотя бы один из ее соседей находится в выравнивающем состоянии, то ячейка переходит в состояние  $\tilde{0}$ .

Из того, что  $\mathcal{K}'$  является связной и из правил, описанных выше, следует, что любая ячейка  $\mathcal{K}'$  побывает в выравнивающем состоянии ровно один такт и после перейдет в состояние из  $E_2$ . Таким образом, будет сформирована конфигурация  $\mathcal{K}$ .

### 4.7. Сложность представления $\mathcal{K}$

Число состояний ОС  $|E_\sigma| = m + \text{const}$ , где  $\text{const} \leq 100$ . Из того, что мы положили  $m = \lceil l^{-1}(2^{d^2}) \rceil$  следует, что  $L_\sigma(\mathcal{K}) \leq l^{-1}(2^{d^2}) + \text{const}$  для всех  $\mathcal{K} \in D(d)$ . В совокупности с результатом предыдущей части работы получаем, что верна следующая

**Теорема 1.** *Для почти любой плоской бинарной конфигурации  $\mathcal{K}$*

$$L(\mathcal{K}) \sim l^{-1}(2^{\text{diam}^2(\mathcal{K})}), \text{ при } \text{diam}(\mathcal{K}) \rightarrow \infty.$$

### 4.8. О времени представления конфигурации

Обозначим через  $T_{(\sigma, q_0)}(\mathcal{K})$  время представления конфигурации  $\mathcal{K}$  в ОС  $\sigma$  с помощью состояния  $q_0$ . Оценим эту величину сверху для построенной нами ОС  $\sigma(\mathcal{K})$  и инициального состояния  $(0, 0, 5)$ . Для этого пройдемся по этапам выращивания  $\mathcal{K}$  и оценим время, необходимое каждому из них.

Построение полигона требует 6 тактов. Для формирования границы необходимо  $6(n_1 + n_2 + 2)$  тактов; формирование собственно  $m$ -ричного кода  $\mathcal{K}$  занимает  $6m^5$  тактов. Число шагов деления на этапе перевода  $m$ -ричного кода  $\mathcal{K}$  в двоичный можно оценить длиной получающегося двоичного кода:  $m^5 \cdot \log_2(m)$ . Следовательно, число тактов этого этапа не превосходит  $12m^5 \cdot \log_2(m)$ . Наконец, выравнивание потребует  $n_1 + n_2 + C$  тактов, где  $C$  — некоторая константа.

Так как  $\max(n_1, n_2) \leq d$ , где  $d = \text{diam}(\mathcal{K})$ , а  $m \sim l^{-1}(2^{d^2})$  и, следовательно,  $m^5 \cdot \log_2(m) \sim d^2$ , то получаем следующую оценку:

$$T_{(\sigma(\mathcal{K}), (0,0,5))}(\mathcal{K}) \leq 14d + 6m^5 + 12m^5 \cdot \log_2(m) \leq 12d^2.$$

В заключении, автор хочет выразить признательность своему научному руководителю — профессору Кудрявцеву Валерию Борисовичу, которому принадлежит постановка рассматриваемой задачи, и благодарит его за помощь в работе над статьей.

## Список литературы

- [1] Кудрявцев В.Б., Подколзин А.С., Болотов А.А. Основы теории однородных структур. М.: Наука, 1990.
- [2] Думов А.С. О времени существования конфигураций в однородных струтурах // Фундаментальная и прикладная математика. Т. 6. Вып. 1. 2001. С. 137–149.