

Об обходе лабиринтов автоматами, оставляющими след в вершинах лабиринта

А.В. Голованов

1 Введение

Впервые проблему обхода лабиринтов автоматами поставил К. Шеннон в начале шестидесятых годов. Будахом [1] было показано, что не существует автомата, обходящего произвольный лабиринт. Для более узких классов лабиринтов задача обхода автоматами рассмотрена в [2], [3]. Что же касается произвольных лабиринтов, то для решения проблемы обхода необходимо рассматривать автоматы, обладающие дополнительными возможностями. В работе [4] рассмотрен автомат, расставляющий (по своему выбору) в вершинах лабиринта нестираемые отметки.

Интерес представляет случай, когда автомат "принудительно" оставляет отметки в посещенных вершинах лабиринта. На содержательном уровне это можно проиллюстрировать как след, оставляемый на поверхности планеты аппаратом или как запах, оставляемый животным.

Однако, решение задачи обхода для автоматов, обладающих таким свойством, встречает затруднения, и в данной работе будет рассмотрено естественное обобщение этой задачи.

2 Основные понятия. Формулировка полученного результата

Основные определения, касающиеся автоматов и лабиринтов, взяты из [5].

Пусть X_α - некоторое индексированное семейство множеств, множество индексов A - конечно. Через α будем обозначать отображение проектирования произведения $\prod_{\beta \in A} X_\beta$ на его α -й сомножитель X_α .

Пусть $L = (V, \Gamma)$ - связный ориентированный граф, V - множество вершин, Γ - множество дуг, эти множества будем обозначать $V(L)$ и $\Gamma(L)$. Пусть $\gamma = (u, v) \in \Gamma$. Обозначим γ^{-1} дугу (v, u) . Если в графе L вместе

с дугой (u, v) содержится дуга (v, u) , то эту пару дуг будем называть ребром и обозначать $\langle u, v \rangle$.

Граф L называется симметрическим, если $\forall (u, v) \in \Gamma \langle u, v \rangle \subseteq \Gamma$.

Пусть $v \in V$. Обозначим $\Gamma_v = \{\gamma \in \Gamma | 1(\gamma) = v\}$ - множество дуг, "выходящих" из вершины v .

Система вращения графа L есть множество $\{\eta_v | v \in V\}$, где η_v - циклическая перестановка множества Γ_v .

Стороной графа называется множество дуг $\gamma_1, \dots, \gamma_n$; $\gamma_i = (u_i, v_i)$ такое, что:

- 1) $u_i = v_{i-1}, i = 2, \dots, n; u_1 = v_n$
- 2) $\gamma_i = \eta_{u_i}(\gamma_{i-1}^{-1}), i = 2, \dots, n; \gamma_1 = \eta_{u_1}(\gamma_n^{-1})$
- 3) все дуги $\gamma_1, \dots, \gamma_n$ - различные.

Понятие стороны, таким образом, связано с системой вращения графа.

Замечание. 1) Каждая дуга симметрического графа (с некоторой системой вращения) принадлежит одной и только одной стороне.

2) У конечного графа число сторон конечно.

Пусть $M \in \mathbf{R}^2, N \in \mathbf{R}^2, M \neq N, \overline{MN} = ai + bj, a, b \in \mathbf{R}, i, j$ - единичные координатные векторы декартовой системы координат на плоскости. Будем говорить, что $\overline{uv}, u, v \in V(L)$ идет в направлении n , если $a = 0, b > 0$; в направлении e , если $a > 0, b = 0$; в направлении s , если $a = 0, b < 0$; в направлении w , если $a < 0, b = 0$.

Обозначим $D = \{n, e, s, w\}$.

Пусть Ω и Σ - алфавиты букв ω и σ , которые берутся в качестве отметок вершин и дуг графа соответственно, $\Omega \cap \Sigma = \emptyset$.

Лабиринтом назовем связный симметрический граф, не имеющий петель и кратных дуг, такой, что разным дугам, выходящим из одной вершины, приписаны разные отметки. Класс лабиринтов, имеющих в качестве отметок вершин и дуг алфавиты Ω и Σ , обозначим как $\mathbf{L}(\Omega, \Sigma)$.

Отметку дуги γ и вершины v будем обозначать $|\gamma|$ и $|v|$, степень вершины v , то есть количество выходящих из нее дуг, будем обозначать $\deg(v)$.

Множество T отрезков на плоскости назовем конфигурацией, если любые два различных отрезка из T могут иметь не более одной общей точки, и, если таковая имеется, то она обязательно является концевой для обоих отрезков.

Лабиринт L называется прямоугольным, если

- 1) $\Sigma = D$.
- 2) $\forall u, v \in V$, если $(u, v) \in \Gamma$, то отрезок \overline{uv} идет в направлении $|(u, v)|$.
- 3) множество отрезков $T = \{\overline{uv} | u, v \in V\}$ является конфигурацией.

Прямоугольный лабиринт с выделенной начальной вершиной v_0 будем называть инициальным лабиринтом и обозначать L_{v_0} .

Пусть τ - циклическая перестановка множества $\{n, e, s, w\}$. Определим отображение η "поворота по часовой стрелке" в вершине v : $\eta(v, r) = \tau_\alpha(r)$, где $\alpha = \min\{m \in \mathbf{N} \mid \exists \gamma \in \Gamma_v : |\gamma| = r\}$, $r \in D$.

$$\eta^i(v, r) = \eta(v, \eta^{i-1}(v, r)), i \geq 2; \eta^{-1}(v, r) = r' : \eta(v, r') = r.$$

Дугу, выходящую из вершины v в направлении $r \in D$ обозначим $\gamma(v, r)$.

Будем считать в дальнейшем, что множества Γ_v упорядочены следующим образом:

$$\Gamma_v = \{\eta(v, r), \dots, \eta^{\deg(v)}(v, r)\}.$$

Тогда с графом связана система вращения "по часовой стрелке". При этом каждая сторона H может быть обойдена по "правилу правой руки" то есть, если переходить, начиная с некоторой дуги $\gamma_i = (u_i, v_i)$, $\gamma_i \in H$ к дуге $\gamma(v_i, \eta(v_i, |\gamma_i^{-1}|))$, и так далее, то все дуги данной стороны будут обойдены.

Множество $\mathbf{R}^2 \setminus L$ является открытым и, в общем случае несвязным. Если L - прямоугольный лабиринт, то число компонент связности $\mathbf{R}^2 \setminus L$ равняется числу сторон. Если это число равно k , то лабиринт называют k -связным.

Пусть $A_{q_0} = (A, Q, B, \varphi, \psi, q_0)$ - инициальный автомат. A_{q_0} будем называть допустимым для лабиринтов из $\mathbf{L}(\Omega, \Sigma)$, если его входной алфавит состоит из букв a вида $(\omega, \{\sigma_1, \dots, \sigma_n\})$, где $\omega \in \Omega$ и $\{\sigma_1, \dots, \sigma_n\} \subseteq \Sigma$, и выходной алфавит есть $\Omega \times (\Sigma \cup \{\Lambda\})$, $\Lambda \notin \Sigma$, при этом всегда ${}_2(\psi(a_n, q_n)) \in {}_2(a_n) \cup \{\Lambda\}$.

Всюду в дальнейшем будем рассматривать только допустимые для того или иного класса лабиринтов автоматы.

Будем считать, что в качестве алфавитов, которыми могут быть нагружены вершины некоторого лабиринта L , взяты конечные отрезки натурального ряда: $E_M = \{0, 1, \dots, M-1\}$, т.е. $L \in \mathbf{L}(E_M, D)$. Пусть $K_M = \{1, 2, \dots, M-1\}$, $D^* = \{\emptyset \neq d \subseteq D\}$.

В дальнейшем, для краткости будем писать $\psi_i(a_t, q_t)$ вместо ${}_i(\psi(a_t, q_t))$, $i = 1, 2$.

Функционирование A_{q_0} в L_{v_0} понимаем следующим образом: в начальный момент отметки всех вершин лабиринта равны 0 (не окрашены) и автомат, находящийся в состоянии q_0 , помещается в начальную вершину v_0 . Если в некоторый момент времени t автомат находится в вершине v_t , то входной буквой a_t является пара, состоящая из отметки вершины и множества отметок дуг, выходящих из данной вершины. В следующий момент, если $\psi_2(a_t, q_t) = \Lambda$, автомат остается в вершине v_t , иначе перемещается в вершину ${}_2(\gamma(v_t, \psi_2(a_t, q_t)))$, заменяя при этом отметку вершины v_t на $\psi_1(a_t, q_t)$, и в любом случае переходя в состояние $\varphi(a_t, q_t)$.

Автоматом, оставляющим след, будем называть инициальный автомат $A_{q_0} = (A, Q, B, \varphi, \psi, q_0)$ такой, что $A = E_2 \times D^*$, $B = 1 \times \{D \cup \Lambda\}$. Таким образом, автомат A_{q_0} окрашивает в "отметку 1" каждую вершину, в которой он побывал в процессе функционирования. Будем называть отметку 1 следом.

Рассмотрим следующее обобщение этого автомата.

Автоматом с M красками будем называть инициальный автомат $\tilde{A}_{q_0} = (A, Q, B, \varphi, \psi, q_0)$ такой, что $A = E_{M+1} \times D^*$, $B = K_{M+1} \times \{D \cup \Lambda\}$, причем выполнены условия: $\mu_n \leq \mu_{n+1} \leq M \forall n \in \mathbf{N}$, $\psi_2(a_n, q_n) \in {}_2(a_n)$, μ_n - номер краски в момент времени n , $\mu_0 = 1$, $\mu_{n+1} = \mu_n + \psi_1(a_n, q_n)$. В силу определения автомата \tilde{A}_{q_0} , $\mu_n \leq M \forall n \in \mathbf{N}$ (номер краски не убывает и не превосходит максимального номера краски из множества K_M). \tilde{A}_{q_0} является допустимым для лабиринтов из класса $\mathbf{L}(E_{M+1}, D)$. Класс таких автоматов будем обозначать \tilde{A}_M .

Класс автоматов, оставляющих след, таким образом, будет совпадать с \tilde{A}_1 .

Поведением A_{q_0} в L_{v_0} называется последовательность пар $\pi = \pi_i(A_{q_0}, L_{v_0})$, $\pi_i(A_{q_0}, L_{v_0}) = (q_i, v_i)$, $i \in \mathbf{N}_0$. Если для $v \in V(L) \exists n \in \mathbf{N}_0 : v = {}_2(\pi_n(A_{q_0}, L_{v_0}))$, то говорим, что A_{q_0} обходит вершину v .

Обозначим $\text{Int}(A_{q_0}, L_{v_0})$ - множество вершин лабиринта, которые обходятся автоматом. Если $\text{Int}(A_{q_0}, L_{v_0}) = V(L)$, то говорим, что A_{q_0} обходит лабиринт L_{v_0} , если A_{q_0} обходит лабиринт $L_v \forall v \in V(L)$, то говорим, что A_{q_0} сильно обходит лабиринт L .

Пусть \mathbf{L}_k - класс k -связных лабиринтов, для которых допустимы автоматы из рассматриваемого класса.

Теорема 1 $\forall k \in \mathbf{N} \exists \tilde{A}_{q_0} \in \tilde{A}_M : \forall L \in \mathbf{L}_k, M = M(k), \tilde{A}_{q_0}$ сильно обходит L , причем $M \leq 198k$.

3 Обход лабиринтов с использованием одного алгоритма обхода

Рассмотрим алгоритм **A** обхода графов, являющихся прямоугольными лабиринтами. У таких графов $1 \leq \deg(v) \leq 4 \forall v \in V(L)$. Пусть лабиринт L - k -связный (следовательно, L имеет k сторон), n_i - число дуг в стороне H_i . Пусть $n_{\max} = \max\{n_i\}$.

Алгоритм **A** работает в дискретном времени и, просматривая вершину v_t в момент времени t , формирует две таблицы: основную $W(L)$ размера $n_{\max} \times k$ и вспомогательную $Z(L)$ размера $2 \times (k - 1)$.

1) инициализация: v_0 - начальная вершина, $|\gamma_0|$ - начальное направление (где $\gamma_0 = \min(\Gamma_{v_0})$). Дуга γ_0 принадлежит некоторой стороне, которую мы назовем первой. Записываем в первую строку первым элементом вершину v_0 , переходим к п.2.

Обозначения (см. рис. 1): пусть в вершину v мы пришли из вершины u . Обозначим направления, а также вершины, в которые из v ведут дуги в этих направлениях, как $C^{-1}(v) = |(v, u)|$, $C^i(v) = \eta^i(v, |(v, u)|)$, $i = 1, 2, \dots, \deg(v)$.

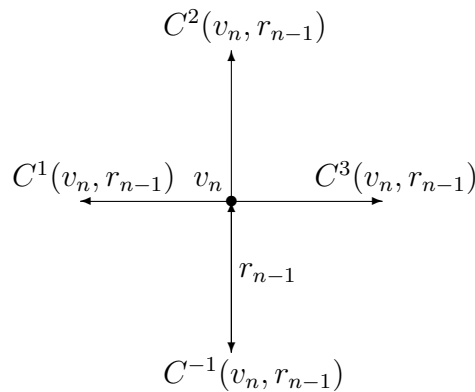


Рис. 1.

Пусть совершается обход i -й стороны, текущую вершину будем обозначать v_t .

2) 1-обход: совершаем обход стороны по правилу правой руки, т.е., пока обход не закончен, переходим от вершины v_t к вершине $v_{t+1} = C^1(v_t)$, вершину v_{t+1} записываем очередным элементом в i -ю строку, снова выполняем п.2. Обход стороны будет завершен, если $v_t = w_{i_0}$ и $C^1(v_t) = w_{i_1}$, в этом случае переходим к п.3.

Замечание. В i -й строке к этому моменту записаны все вершины, принадлежащие i -й стороне.

3) 2-обход: двигаемся по вершинам i -й стороны в порядке их записи в таблицу W . Для вершин степени больше двух проверяем по порядку вершины $C^2(v_t)$ и $C^3(v_t)$, записаны они в таблицу W или нет.

3.1) Пусть проверяем j -й элемент i -й строки и одна из вершин $C^2(v_t)$, $C^3(v_t)$, например $C^2(v_t)$, отсутствует в таблице. Это означает, что $C^2(v_t)$ находится на новой стороне, отличной от всех сторон, записанных в таблицу W ранее. Пусть на этот момент в таблицу записаны m строк, тогда новую сторону назовем $(m+1)$ -й, запишем в m -ю строку вспомогательной таблицы Z два натуральных числа: номер строки i (т.е. номер стороны) и номер элемента строки j , в которых была обнаружена новая не обойденная вершина. Обнаруженная вершина объявляется первой вершиной $(m+1)$ -й стороны, записывается первым элементом в новую, $(m+1)$ -ю строку

таблицы W . Направление $|(w_{ij}, C^2(w_{ij}))|$ считаем начальным для $(m+1)$ -й стороны. Будем говорить при этом, что из j -й вершины i -й стороны произошло углубление в $(m+1)$ -ю сторону.

3.2) Если при проверке w_{ij} все вершины множества $\{C^2(w_{ij}), \dots, C^n(w_{ij})\}$, $n = \deg(w_{ij})$ записаны в таблицу и вершина w_{ij} не последняя в i -й строке (т.е. $j < n_i$), переходим к следующей вершине строки и выполняем п.3. Если вершина w_j - последняя, и вершины $\{C^2(w_{ij}), \dots, C^n(w_{ij})\}$ записаны в таблицу, то 2-обход i -й стороны закончен. Если $i \neq 1$, то в $(i-1)$ -й строке вспомогательной таблицы записаны номер строки и вершины этой строки, из которых произошло углубление в i -ю сторону. 1 Переходим к продолжению 2-обхода h -й стороны, $h = z_{i-1,1}$, т.е. к п.3, начиная с g -го элемента, $g = z_{i-1,2}$. При этом говорим, что произошло возвращение из i -й стороны в h -ю.

Алгоритм заканчивает работу, когда завершается 2-обход первой стороны.

Пример графа L с занумерованными вершинами и матриц $W(L)$ и $Z(L)$, полученных в результате работы алгоритма **A**, показан на рис. 2 и 3, здесь начальная вершина - 1, начальное направление $|(1, 2)|$.

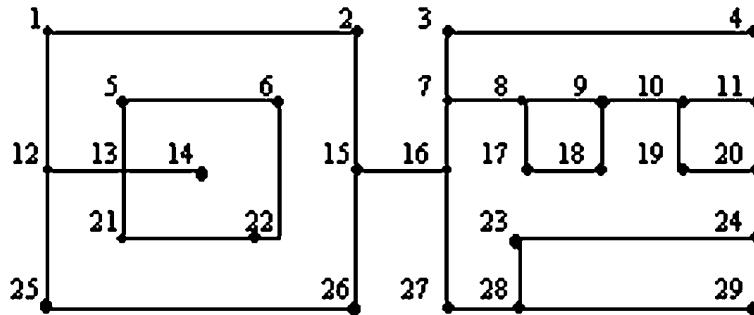


Рис. 2.

№	1	2	15	16	<u>7</u>	3	4	11	20	<u>24</u>	29	28	27	16	15	26	25	<u>12</u>	1
2	<u>8</u>	9	<u>10</u>	11	4	3	7	8	-	-	-	-	-	-	-	-	-	-	-
3	17	18	9	8	17	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	19	20	11	10	19	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	23	28	29	24	23	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	<u>13</u>	5	6	22	21	13	12	25	26	15	2	1	12	13	-	-	-	-	-
7	14	13	21	22	6	5	13	14	-	-	-	-	-	-	-	-	-	-	-

1	5
2	1
2	3
1	10
1	18
6	1

Рис. 3.

Будем говорить, что алгоритм обходит вершину v лабиринта, если она заносится в таблицу. Алгоритм обходит лабиринт, если он обходит все его вершины.

Для доказательства некоторых утверждений об алгоритме **A** рассмотрим вспомогательный граф $R(L)$, который будем строить по ходу выполнения алгоритма.

Вершины графа R соответствуют сторонам лабиринта L , а из вершины r_1 в вершину r_2 ведет дуга в том и только в том случае, когда из стороны r_1 происходит углубление в сторону r_2 , причем дуги, выходящие из одной вершины, упорядочены по времени углубления.

Существование цикла в графе R невозможно, так как если из вершины r_1 на некотором шаге выходит дуга, то сторона r_1 лабиринта уже полностью 1-обойдена, следовательно, все ее вершины записаны в таблицу W и углубления в r_1 -ю сторону не происходит. Таким образом, поскольку число вершин R конечно, R является деревом, корнем которого является первая сторона, при углублении алгоритм переходит в вершину-потомок, при возвращении - в вершину-родитель. Нетрудно видеть, что при выполнении алгоритма происходит обход дерева R в глубину, следовательно, из каждой вершины R , не являющейся корнем произойдет возвращение, значит, будет закончен ее 2-обход (пример R -дерева для рассмотренного графа показан на рис. 4).

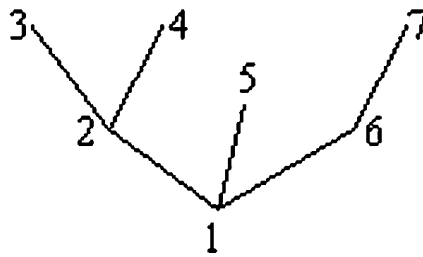


Рис. 4.

Но после того, как будут 2-обойдены все вершины, в которые ведут дуги, выходящие из корня, будет закончен и 2-обход корневой вершины, таким образом, справедливо

Утверждение 1 *Если был начат 2-обход некоторой стороны, то он будет закончен за конечное время.*

Таким образом, алгоритм заканчивает работу за конечное время. Назовем моментом времени $\mathbf{T} = \mathbf{T}(L)$ момент, когда заканчивается 2-обход первой стороны. Будем иногда употреблять для вершин лабиринта термин "окрашена" вместо термина "обойдена" поскольку для определенных выше автоматов с красками эти понятия совпадают. На момент \mathbf{T} справедливы следующие утверждения.

Утверждение 2 Если вершина $v \in V(L)$ 1-обойдена, то она и 2-обойдена (Очевидно из алгоритма).

Другими словами, если вершина окрашена, то она была 2-обойдена в некоторый момент времени.

Утверждение 3 Не $\exists(u, v) \in \Gamma(L) : (|u| \neq 0) \wedge (|v| = 0)$.

Доказательство. Пусть такая дуга существует. Тогда, по Утверждению 2, происходил 2-обход вершины u . Сразу заметим, что степень вершины u не менее трех, иначе вершина v была бы обойдена как соседняя для вершины u уже при 1-обходе u .

Пусть $\deg(u) = 3$, и при выполнении 2-обхода в вершину u мы пришли из вершины $C^{-1}(u)$. Тогда вершины $C^{-1}(u)$ и $C^1(u)$ были 1-обойдены, следовательно, неокрашенной может быть только вершина $C^2(u)$, но ее мы в этом случае обнаружим следующим же шагом 2-обхода, углубимся в нее и окрасим. Аналогично, если степень u равна четырем, доказываем, что v не может быть вершиной $C^2(u)$ и вершиной $C^3(u)$ при окрашенной $C^2(u)$. Остается рассмотреть случай, когда и $C^2(u)$, и $C^3(u)$ не окрашены. Тогда происходит углубление в сторону, которой принадлежит $C^2(u)$. Пусть номер этой стороны равен m .

Если $C^3(u)$ также принадлежит стороне m , то $C^3(u)$ будет обойдена при 1-обходе, если не принадлежит, то $C^3(u)$ может остаться не окрашенной только до момента 2-обхода вершины $C^2(u)$, стороны m . В этот момент, если $C^3(u)$ осталась не окрашенной, алгоритм произведет углубление в сторону, которой принадлежит $C^3(u)$, и окрасит ее. Таким образом, оказываются окрашенными все соседние вершины для u , и мы приходим к противоречию.

Утверждение 4 Алгоритм обходит любой лабиринт за время \mathbf{T} .

Доказательство. По крайней мере одну вершину (начальную) алгоритм обходит, $|v_0| \neq 0$.

Пусть существует не обойденная при $t = \mathbf{T}$ вершина w , $|w| = 0$. Тогда, в силу связности лабиринта, существует путь $S \subseteq \Gamma$, соединяющий вершины v_0 и w . Но тогда существует дуга $(u, v) \in S : (|u| \neq 0) \wedge (|v| = 0)$, что противоречит Утверждению 3.

Доказательство теоремы 1

Нам нужно построить автомат с красками, реализующий алгоритм **A**, без использования таблиц W и Z , совершающий 1- и 2-обходы, углубление

и возвращение. Для этой цели будем расставлять в вершинах лабиринта метки т.е. краски, по которым автомат будет определять, в какой ситуации он находится. Будем считать пока, что число красок сколь угодно велико.

В начале 1-обхода ставится метка, соответствующая направлению $|(C^{-1}(v_t), v_t)|$, то есть направлению, в котором мы пришли в v_t . Автомат определяет, что 1-обход закончен, если он попадает в вершину с меткой начала обхода, и направление $|(C^{-1}(v_t), v_t)|$ соответствует направлению, закодированному меткой. Аналогично автомат определяет окончание 2-обхода.

В течение 2-обхода алгоритм проверяет, обойдены ли вершины множества $\{C^2(v_t), \dots, C^{\deg(v_t)}(v_t)\}$, в соответствии с этим автомат, находясь в вершине v_t , совершает "вылазки" в вершины $C^2(v_t), \dots, C^n(v_t)$, возвращаясь, если вершина окрашена, и совершая углубление, если вершина не окрашена. При углублении, в соответствии с алгоритмом, начинается 1-обход новой стороны (с той вершины, которая была обнаружена неокрашенной при вылазке), при этом начальным направлением обхода считается направление $|(v_t, C^1(v_t))|$.

Будем различать случаи углубления в вершины $C^2(v_t)$ и $C^3(v_t)$, поскольку при окончании 2-обхода (при возвращении в предыдущую сторону) необходимо вернуться именно к тому направлению 2-обхода предшествующей стороны, в котором он производился до углубления. Если 2-обход закончен в направлении $r \in D$, то для того, чтобы вернуться из C^2 -углубления в предшествующую сторону, переходим в нее в направлении $\eta^{-1}(r^{-1})$, в случае C^3 -углубления (которое возможно только в вершине степени 4) - в направлении $\eta^{-2}(r^{-1})$.

Замечание 1. Метка начала 2-обхода каждой стороны будет "затирать" метку начала 1-обхода этой стороны и находиться в той же вершине.

Замечание 2. Метки начал 2-обходов разных сторон находятся в разных вершинах и поэтому не затирают друг друга (поскольку новую метку мы всегда ставим в неокрашенную вершину). Таким образом, для реализации алгоритма необходимо 9 типов красок:

метки, соответствующие 4-м направлениям начала обхода в случае C^2 -углубления,

метки, соответствующие 4-м направлениям начала обхода в случае C^3 -углубления,

один тип краски, закрашивающий все остальные вершины, в которых не стоит меток ("непомеченные").

Заметим, что по определению автомата с красками, при попадании на некоторую вершину автомат закрашивает ее своей текущей краской (обязательно оставляет отметку). Поэтому, для того, чтобы сохранять информацию об отметке начала обхода, будем поступать следующим образом.

Разобьем множество красок K_m автомата \tilde{A}_{q_0} на отрезки длины 9 и не будем различать краски множества K_m , номера которых равны по модулю 9. На каждом отрезке естественным образом вводится нумерация, также по модулю 9:

1,2,3,4 - метки начал обходов, соответствующие направлениям n, e, s, w , случай C^2 -углубления.

5,6,7,8 - метки начал обходов, соответствующие направлениям n, e, s, w , случай C^3 -углубления.

0 - основная краска (натуральное число, кратное 9).

Таким образом, с точки зрения автомата \tilde{A}_{q_0} каждая вершина лабиринта либо не окрашена, либо в ней стоит одна из 8-ми меток, либо она окрашена основной краской.

Теперь, если автомат попадает в вершину, отметка которой отличается по модулю 9 от номера текущей краски, причем эту отметку необходимо сохранить, автомат подбирает наименьший номер n , превосходящий номер текущей краски, такой, чтобы n совпадал по модулю 9 с отметкой вершины, и двигается дальше. Тогда отметка вершины будет нести ту же информацию, что и раньше. Если лабиринт L - k -связный, то в его вершинах автоматом \tilde{A}_{q_0} расставляется не более k меток, следовательно, главным образом вершины окрашиваются основной краской (поскольку вершин у лабиринта значительно больше, чем сторон). Поэтому типичным является случай, когда основная краска является текущей и метка окружена вершинами неокрашенными или окрашенными основной краской. В этом случае при прохождении через помеченную вершину, отметку которой необходимо сохранить, автомат "затрачивает" 9 красок: если отметка равна $n \pmod{9}$, то к номеру текущей краски прибавляется n , автомат окрашивает данную вершину, сохраняя отметку, затем требуется еще $(9 - n)$ красок, чтобы текущая краска снова стала основной.

В каждый момент времени t и в каждом состоянии $q \in Q(\tilde{A}_{q_0})$, кроме начального, требуется знать направление $|(C^{-1}(v_t), v_t)|$. Поэтому состояний q_i , соответствующих каждому шагу алгоритма - четыре, по одному на каждое направление. В соответствии с этим закодируем состояния как $\langle q_i, r \rangle, r \in D$. Будем считать, что на входе \tilde{A}_{q_0} имеется также остаток от деления текущего цвета на 9, то есть автомат каким-то образом знает, какое число нужно прибавлять к текущему номеру краски, чтобы сохранить отметку помеченной вершины. Это сделано для упрощения изложения, но, в принципе, без этой входной информации можно было бы обойтись, храня информацию о текущей краске в состояниях автомата.

Пусть ω_n - отметка вершины v_n ; $\chi(n) = 1, \chi(e) = 2, \chi(s) = 3, \chi(w) = 4$. $\mu_{n+1} = \mu_n + \psi_1(a_n, q_n)$ - наименьшее из возможных, равное по модулю 9

указанному в таблице.

Задаем автомат функциями φ и ψ (см. таблицу 1).

Таким образом, построенный автомат реализует алгоритм **A**.

Определим момент времени T как момент, в который будет 2-обойдена автоматом \tilde{A}_{q_0} первая сторона лабиринта L (сохраняя терминологию, относящуюся к алгоритму **A**).

Замечание. К моменту времени T лабиринт L , согласно Утверждению 4, будет обойден, и нам не важно, как поведет себя автомат при $t > T$.

Лемма 1 Если лабиринт L - k -связный, то к моменту времени T автомат \tilde{A}_{q_0} затратит не более $198k$ красок.

Доказательство. Каждой стороне соответствует одна метка начала обхода этой стороны, таким образом, в вершинах лабиринта ставится не более k меток.

Состояние	$r_n = \psi_2(a_n, q_n)$	$\mu_{n+1} \bmod 9$	$\varphi(a_n, q_n)$	Примечание
$\langle \mathbf{q}_0 \rangle$	первое из Γ_{v_0}	0	$\langle q_1, r_0 \rangle$	начальное
$\langle \mathbf{q}_1, \mathbf{r}_{n-1} \rangle$ $(\omega_n \neq \chi(r_{n-1})) \wedge$ $(\omega_n \neq \chi(r_{n-1}) + 4)$ иначе	$C^1(v_n, r_{n-1})$ Λ	ω_n ω_n	$\langle q_1, r_n \rangle$ $\langle q_2, r_{n-1} \rangle$	1-обход
$\langle \mathbf{q}_2, \mathbf{r}_{n-1} \rangle$ $(\omega_n \neq \chi(r_{n-1})) \wedge$ $(\omega_n \neq \chi(r_{n-1}) + 4)$ $\deg(v_n) \leq 2$ $\deg(v_n) > 2$ $\omega_n = \chi(r_{n-1})$ $\omega_n = \chi(r_{n-1}) + 4$	$C^1(v_n, r_{n-1})$ $C^2(v_n, r_{n-1})$ $C^{-1}(v_n, r_{n-1})$ $C^{-1}(v_n, r_{n-1})$	ω_n ω_n 0 0	$\langle q_2, r_n \rangle$ $\langle q_3, r_{n-1} \rangle$ $\langle q_7, \eta^{-1}(v_n, r_{n-1}) \rangle$ $\langle q_7, \eta^{-2}(v_n, r_{n-1}) \rangle$	2-обход
$\langle \mathbf{q}_3, \mathbf{r}_{n-1} \rangle$ $\omega_n = 0$ $\omega_n \neq 0$	$C^1(v_n, r_{n-1})$ $C^{-1}(v_n, r_{n-1})$	$\chi(r_{n-1})$ ω_n	$\langle q_1, r_n \rangle$ $\langle q_4, r_{n-1} \rangle$	вылазка в C^2 -вершину
$\langle \mathbf{q}_4, \mathbf{r}_{n-1} \rangle$ $\deg(v_n) = 3$ $\deg(v_n) = 4$	$C^1(v_n, r_{n-1})$ $C^3(v_n, r_{n-1})$	ω_n ω_n	$\langle q_2, r_n \rangle$ $\langle q_5, r_n \rangle$	возврат из C^3 -вылазки
$\langle \mathbf{q}_5, \mathbf{r}_{n-1} \rangle$ $\omega_n = 0$ $\omega_n \neq 0$	$C^1(v_n, r_{n-1})$ $C^{-1}(v_n, r_{n-1})$	$\chi(r_{n-1}) + 4$ ω_n	$\langle q_1, r_n \rangle$ $\langle q_6, r_{n-1} \rangle$	вылазка в C^3 -вершину
$\langle \mathbf{q}_6, \mathbf{r}_{n-1} \rangle$	$C^1(v_n, r_{n-1})$	ω_n		переход к следующей вершине при 2-обходе
$\langle \mathbf{q}_7, \mathbf{r}_{n-1} \rangle$	$C^1(v_n, r_{n-1})$	ω_n	$\langle q_2, r_{n-1} \rangle$	возвращение к соответствующей стороне

Таблица 1.

Пусть v - одна из помеченных вершин. При $t \leq T \tilde{A}_{q_0}$ может побывать в этой вершине самое большое:

3 раза при 1-обходах сторон,

12 раз при 2-обходах сторон, которым v принадлежит (если степень v равна 4),

7 раз при 2-обходах вершин, соседних с v (если все эти вершины степени 4). Четвертое попадание при 1-обходе нами не учтено, так как при этом сразу начинается 2-обход и это попадание по сути совпадает с первым для 2-обхода, которое мы и учитываем. Таким образом, на метку уходит не более $(3 + 12 + 7) \times 9 = 198$ красок, всего на лабиринт - не более $198k$ красок.

Следствие 1 (Теорема 1) $\forall k \in \mathbf{N} \exists \tilde{A}_{q_0} \in \tilde{A}_M : \forall L \in \mathbf{L}_k, M = M(k), \tilde{A}_{q_0}$ сильно обходит L , причем $M \leq 198k$.

Замечание. Количество красок, затраченных на данную вершину, зависит от ее расположения, степени и степеней ее соседей, наибольшее количество красок для данного алгоритма равно 198. Можно показать, что в самом лучшем случае на помеченную вершину затрачивается не менее двух красок, так что $2k \leq M \leq 198k$.

4 Дополнение

Назовем двусвязным лабиринт, у которого число сторон k равно двум (см. рис. 5).

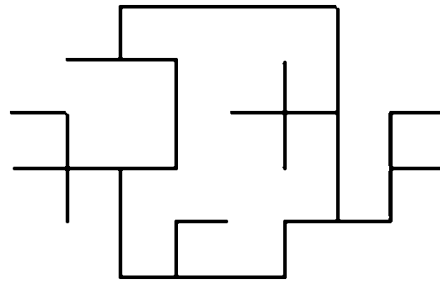


Рис. 5.

Предложение 1 В случае $k = 2 \exists A_{q_0} \in \tilde{A}_1$ такой, что $\forall L \in \mathbf{L}_2 A_{q_0}$ сильно обходит L .

Автомат с одной краской обязательно окрашивает вершину, из которой выходит (в краску 1), таким образом, каждая вершина лабиринта либо окрашена в 0 (при этом автомат в ней не был, вершину будем называть неокрашенной), либо окрашена в 1 (будем называть ее просто окрашенной).

Пусть H_1 - одна сторона двусвязного лабиринта L , H_2 - его вторая сторона.

Пусть $V_i = \{v \in V(L) \mid \exists \gamma \in H_i : {}_1(\gamma) = v\}$, $i = 1, 2$ - множество вершин стороны.

Непустое множество вершин, общих для обеих сторон $V_c = V_1 \cap V_2$ назовем циклом лабиринта L , а множество $V_i^0 = V_i \setminus V_c$, $i = 1, 2$ - внутренними точками стороны H_i .

Если множество V_i^0 несвязно, то, очевидно, каждая из компонент связности V_i^0 , которые мы будем называть отростками, является деревом. Корнем отростка назовем вершину цикла, соседнюю с данным отростком.

Пусть в вершину u мы пришли из v .

Введем обозначения (см. рис. 6): $L(u) = C^1(u)$, $R(u) = C^{\deg(u)-1}(u)$, $B(u) = C^{-1}(u) = v$, $LL(u) = L(L(u))$, $RR(u) = R(R(u))$.

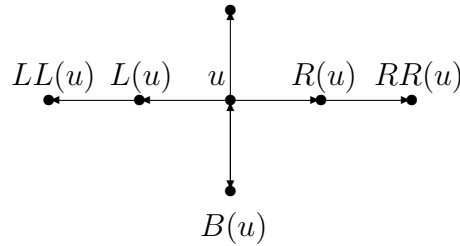


Рис. 6.

Пусть v_0 - начальная вершина, $r \in D$ - начальное направление, т.е. то направление, по которому производится переход в соседнюю с начальной вершину. Сторону, которой принадлежит дуга $\gamma(v_0, r)$, назовем первой, без ограничения общности считаем, что это H_1 . Оставшуюся сторону назовем второй. Если начальная вершина принадлежит некоторому отростку, назовем его первым.

Ситуацией $1 \rightarrow 2$ в вершине u назовем положение, когда $|u| = 1$, $|L(u)| = 1$, $|R(u)| = 0$.

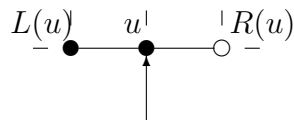


Рис. 7.

Ситуацией $2 \rightarrow 3$ - положение, когда $|u| = 1$, $|L(u)| = 1$, $|LL(u)| = 1$, $|R(u)| = 0$ или когда $|u| = 1$, $|L(u)| = 1$, $|LL(u)| = 1$, $|R(u)| = 1$, $|RR(u)| = 0$.

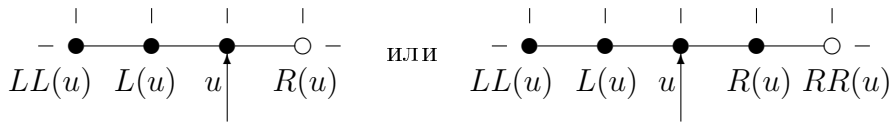


Рис. 8.

Рассмотрим алгоритм \mathbf{A}' обхода двусвязного лабиринта L :

1) 1-й этап обхода первой стороны: если ситуация $1 \rightarrow 2$, то переход в вершину $L(u)$ и ко 2-му этапу. Иначе, обход по правилу правой руки.

2) 2-й этап: если ситуация $2 \rightarrow 3$, то переход в неокрашенную в ситуации $2 \rightarrow 3$ вершину и к 3-му этапу. Иначе, обход по правилу правой руки.

3) 3-й этап: обход второй стороны по правилу правой руки.

Утверждение 5 *Ситуации $1 \rightarrow 2$ на 1-м этапе и $2 \rightarrow 3$ на 2-м этапе не могут произойти в вершине отростка.*

Доказательство. Пусть на 1-м этапе алгоритма ситуация $1 \rightarrow 2$ произошла в некоторой вершине u отростка, т.е. n – момент времени, $|u| = 1$, $|L(u)| = 1$, $|R(u)| = 0$, $v_n = u$, $v_{n-1} = B(u)$. Пусть $L(u) = u_L$, $R(u) = u_R$, $B(u) = u_B$ – фиксированные вершины лабиринта.

$u \in V_i^0 \Rightarrow \{(u, L(u)), (L(u), u), (u, R(u)), (B(u), u)\} \subseteq H_1$. Поскольку окрашены вершины u и u_L , то существует момент времени $k < n$ такой, что либо алгоритм перешел от вершины u_L к вершине u , либо наоборот. И в том, и в другом случае, поскольку в момент времени $n - 1$ алгоритм находился в вершине u_B , существует момент t , $k < t < n$, в который алгоритм перешел из u в u_B . Но так как на первом этапе происходит обход по правилу правой руки, то в момент времени $t - 1$ алгоритм перешел из u_R в u , следовательно, u_R окрашена. Противоречие.

Аналогично доказывается, что на втором этапе алгоритма не может произойти ситуация $2 \rightarrow 3$.

Утверждение 6 *1) На 1-м этапе происходит обход всей первой стороны, за исключением, быть может, некоторой части первого отростка. 1-й этап заканчивается, когда:*

1. *Окрашен весь цикл.*

2. *Окрашена одна вершина отростка второй стороны, соседняя с корнем этого отростка.*

2) *На 2-м этапе заканчивается обход первой стороны, 2-й этап заканчивается, когда найден отросток второй стороны, по крайней мере одна вершина которого не окрашена.*

3) *На 3-м этапе обходим вторую сторону.*

Доказательство. 1) Из Утверждения 5 следует, что ситуация $1 \rightarrow 2$ может произойти только на цикле лабиринта. Пусть цикл не окрашен полностью и мы обходим вершину v цикла.

Возможны три случая:

1. v обходится впервые
2. v является корнем отростка, который тогда должен обходиться впервые, при этом
 - 2.1. вершина $C^1(v)$ принадлежит циклу, тогда она не обойдена.
 - 2.2. вершина $C^1(v)$ принадлежит второму отростку, корнем которого является v , тогда $C^1(v)$ не была еще обойдена.

Во всех случаях ситуация $1 \rightarrow 2$ невозможна по определению.

Таким образом, на этапе 1 производится обход цикла и всех отростков, за исключением, может быть части первого отростка, причем это произойдет только если корень первого отростка является одновременно корнем некоторого отростка второй стороны. Ту вершину, принадлежащую отростку второй стороны, которая была не окрашена при ситуации $1 \rightarrow 2$, обозначим v^* .

Состояние	$r_n = \psi_2(a_n, q_n)$	$\varphi(a_n, q_n)$	Примечание
$\langle \mathbf{q}_0 \rangle$	L	$\langle q, r_n \rangle$	начальное
$\langle \mathbf{q}_1, \mathbf{r}_{n-1} \rangle$			1-й этап
$\omega_n = 0$	L	$\langle q_1, r_n \rangle$	
$\omega_n = 1$	L	$\langle q_2, r_n \rangle$	
$\langle \mathbf{q}_2, \mathbf{r}_{n-1} \rangle$			
$\omega_n = 0$	L	$\langle q_1, r_n \rangle$	
$\omega_n = 1$	B	$\langle q_3, r_n \rangle$	
$\langle \mathbf{q}_3, \mathbf{r}_{n-1} \rangle$	R^2	$\langle q_4, r_n \rangle$	
$\langle \mathbf{q}_4, \mathbf{r}_{n-1} \rangle$			
$\omega_n = 0$	B	$\langle q_5, r_n \rangle$	
$\omega_n = 1$	B	$\langle q_6, r_n \rangle$	
$\langle \mathbf{q}_5, \mathbf{r}_{n-1} \rangle$	L^2	$\langle q_1, r_n \rangle$	
$\langle \mathbf{q}_6, \mathbf{r}_{n-1} \rangle$	L^2	$\langle q_7, r_n \rangle$	
$\langle \mathbf{q}_7, \mathbf{r}_{n-1} \rangle$			2-й этап
$\omega_n = 0$	L	$\langle q_7, r_n \rangle$	
$\omega_n = 1$	L	$\langle q_8, r_n \rangle$	
$\langle \mathbf{q}_8, \mathbf{r}_{n-1} \rangle$			
$\omega_n = 0$	L	$\langle q_7, r_n \rangle$	
$\omega_n = 1$	L	$\langle q_9, r_n \rangle$	
$\langle \mathbf{q}_9, \mathbf{r}_{n-1} \rangle$			
$\omega_n = 0$	L	$\langle q_7, r_n \rangle$	
$\omega_n = 1$	B	$\langle q_{10}, r_n \rangle$	
$\langle \mathbf{q}_{10}, \mathbf{r}_{n-1} \rangle$	R	$\langle q_{11}, r_n \rangle$	
$\langle \mathbf{q}_{11}, \mathbf{r}_{n-1} \rangle$	R^2	$\langle q_{12}, r_n \rangle$	
$\langle \mathbf{q}_{12}, \mathbf{r}_{n-1} \rangle$			
$\omega_n = 0$	L	$\langle q_{16}, r_n \rangle$	
$\omega_n = 1$	R	$\langle q_{13}, r_n \rangle$	
$\langle \mathbf{q}_{13}, \mathbf{r}_{n-1} \rangle$			
$\omega_n = 0$	L	$\langle q_{16}, r_n \rangle$	
$\omega_n = 1$	B	$\langle q_{14}, r_n \rangle$	
$\langle \mathbf{q}_{14}, \mathbf{r}_{n-1} \rangle$	L	$\langle q_{15}, r_n \rangle$	
$\langle \mathbf{q}_{15}, \mathbf{r}_{n-1} \rangle$	L^2	$\langle q_7, r_n \rangle$	
$\langle \mathbf{q}_{16}, \mathbf{r}_{n-1} \rangle$	L	$\langle q_{16}, r_n \rangle$	3-й этап

Таблица 2.

2) Как было сказано при доказательстве п.1), первый отросток не обходится полностью на первом этапе алгоритма, только если вершина v^* - соседняя с корнем первого отростка, а значит, в этом случае мы переходим к выполнению этапа 2 в корне первого отростка. Из алгоритма следует, что следующим шагом мы переходим в первый отросток, следовательно, он будет обойден до возникновения ситуации $2 \rightarrow 3$.

3) Поскольку начальным для этапа 3 является направление из цикла внутрь отростка, то соответствующая этому направлению дуга принадлежит второй стороне. Поэтому обход по правилу правой руки будет обходом именно второй стороны.

Утверждение 7 Алгоритм A' сильно обходит любой двусвязный лабиринт.

Доказательство. Если алгоритм не перейдет ко второму этапу, то V_2^0 пусто, тем самым, лабиринт будет обойден на первом этапе. Если алгоритм перейдет ко второму, но не перейдет к третьему этапу, то V_2^0 состоит из одной вершины v^* , которая была окрашена во время ситуации $1 \rightarrow 2$, и лабиринт также будет обойден. Наконец, если переход к третьему этапу осуществился, то очевидно, лабиринт будет обойден.

Построение автомата с одной краской. Будем называть направлениями L, L^2, R, R^2, B направления $C^1(v_n), C^2(v_n), C^{\deg(v_n)-1}(v_n), C^{\deg(v_n)-2}(v_n), C^{-1}(v_n)$. В таблице 2 заданы функции φ и ψ автомата, ω_n - отметка текущей (на момент n) вершины.

Список литературы

- [1] Budach L. Automata and Labyrinths, Math. Nachrichten 86, 1978, p. 195-282.
- [2] Зыричев А.Н. О синтезе автомата, обходящего плоские лабиринты с ограниченными дырами. Дискретная математика, 1991, т. 3, вып. 1, с. 105-113.
- [3] Золотых А.А. Обход лабиринтов с ограниченными в фиксированном направлении дырами. Дискретная математика, 1993, т. 5, вып. 1, с. 59-69.
- [4] Насыров А.З. Об обходе лабиринтов автоматами, оставляющими нестираемые отметки. Дискретная математика, 1997, т. 9, вып. 1.
- [5] Кудрявцев В.Б., Ушчумлич Ш., Килибарда Г. О поведении автоматов в лабиринтах. Дискретная математика, 1992, т. 4, вып. 3, с. 3-28.