Моделирование поведения автоматов в лабиринтах

Ю.А. Курепа

В последние десятилетия большое внимание уделяется изучению проблемы поведения автоматов в лабиринтах [2]. Наблюдение за движением автоматов в лабиринтах даже в случае, когда входной алфавит, выходной алфавит и алфавит состояний небольшие по объему, является сложным, поэтому естественным образом возникает потребность моделирования поведения автоматов в лабиринтах при помощи компьютера. Для этой цели разработана программа АвтЛаб [4]. Эта программа позволяет «вводить» автоматы, поведение которых рассматривается, с помощью небольшого числа операторов, в краткой и компактной форме. Работая в автономном режиме, программа позволяет установить корректность такого описания автомата, то есть существование соответствующего автомата с данной записью, а также позволяет устранить некорректность описания (в соответствующем диалоге с пользователем). Программный код, который при этом порождается, моделирует поведение данного автомата в любом лабиринте, предоставленном программе. Для произвольно заданного лабиринта обеспечивается визуальное наблюдение за поведением данного автомата в нем. При этом в интерактивном режиме допускаются изменения лабиринтов, так что программа может быть использована в качестве удобного «вспомогательного инструмента» для решения проблем, связанных с поведением автоматов в лабиринтах. Так, например, АвтЛаб может быть очень полезен в процессе поиска универсальных автоматов для заданного класса лабиринтов. В случае, когда такие автоматы не существуют, она обеспечивает возможность путем экспериментирования построить лабиринт-ловушку для рассматриваемого автомата из заданного класса лабиринтов. В настоя-

щей работе рассматривается важный аспект моделирования поведения автоматов в лабиринтах, а именно, система записи автоматов в программе АвтЛаб.

1. Основные понятия и обозначения

Пусть $\overline{z_1, z_2} = \{z \in \mathbb{Z} \,|\, z_1 \leqslant z \leqslant z_2\}$ для любых $z_1, z_2 \in \mathbb{Z}$, $z_1 \leqslant z_2$. Для любого множества X обозначим его булеан через $\mathfrak{B}(X)$, а также положим, что $\mathfrak{B}^{\leqslant n}(X) = \{Y \subseteq X \,|\, |Y| \leqslant n\}$ для любого $n \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Пусть X_1 и X_2 — произвольные множества. Обозначим, через $\mathbf{pr}_i$, $i \in \overline{1,2}$, отображение проектирования произведения $X_1 \times X_2$ на X_i правилом $\mathbf{pr}_i(x_1, x_2) = x_i$ для любой пары $(x_1, x_2) \in X_1 \times X_2$.

Пусть G — орграф и $p = (u, v)$ — некоторая его дуга. Множество вершин орграфа G обозначим через VG, а множество его дуг — через EG. Вершины u и v обозначим соответственно через $v_1(p)$ и $v_2(p)$.

Пусть $D = \{w, n, e, s\}$. Положим, что $e^{-1} = w$, $s^{-1} = n$, $w^{-1} = e$, $n^{-1} = s$. Если $v = (a, b) \in \mathbb{R}^2$ — произвольная точка плоскости, то через vw, vn, ve и vs обозначим точки $(a-1, b)$, $(a, b+1)$, $(a+1, b)$ и $(a, b-1)$, соответственно.

Фиксируем некоторое $r \in \mathbb{N}$. Связный конечный орграф G без кратных дуг и петель, у которого любой дуге p сопоставлена пара $(|p|, < p >) \in D \times \overline{0, r}$ называется (*односторонним*) r*-лабиринтом* [3], если:

1) для любых $p_1, p_2 \in EG$, таких, что $p_1 \neq p_2$ и $v_1(p_1) = v_1(p_2)$, выполнено, что $|p_1| \neq |p_2|$;

2) $VG \subseteq \mathbb{Z}^2$ и $p = (v_1(p), v_1(p)|p|)$ для любой дуги $p \in EG$.

3) в орграфе G выделены вершины v' и v'', $v' \neq v''$; первую из них назовем *начальной*, вторую — *заключительной* вершиной;

4) если $p = (v, u) \in EG$, то $\overline{p} = (u, v) \in EG$ и $|\overline{p}| = |p|^{-1}$.

Символы $|p|$ и $< p >$ называем *направлением* и *отметкой* ребра p, соответственно. Если орграф G кроме выше данных условий удовлетворяет еще и условию, что $< p >=< \overline{p} >$ для любой дуги $p \in EG$, то называем его *двухсторонним* r*-лабиринтом*.

Пусть G — r-лабиринт. Введем обозначения: $[v] = [v]_G = \{p \in EG | \, v_1(p) = v\}$ и $\delta(v, G) = \{(|p|, < p >) \,|\, p \in [v]_G\}$.

Под инициальным конечным автоматом понимаем набор $V = (A, Q, B, \varphi, \psi, q_0)$, где A, Q и B — конечные множества, которые соответственно называем входным алфавитом, алфавитом состояний и выходным алфавитом; $\varphi : Q \times A \to Q$ — функция переходов; $\psi : Q \times A \to B$ — функция выходов, q_0 — начальное состояние автомата.

Инициальный конечный автомат $V = (A, Q, B, \varphi, \psi, q_0)$ называется *r-автоматом*, если удовлетворяет следующим условиям:

1) $A = \mathfrak{B}^{\leqslant 4}(D \times \overline{0, r})\backslash\{\varnothing\}$ и $B = D \times \overline{1, r}$;

2) $|\mathbf{pr}_1(a)| = |a|$ для любого $a \in A$;

3) $\mathbf{pr}_1(\psi(q, a)) \in \mathrm{pr}_1(a)$ для любых $q \in Q$ и $a \in A$.

Через M_r обозначим множество всех r-автоматов, а через $M_{r,p}$ — множество всех автоматов из M_r, у которых p состояний.

Поведением r-автомата $\mathfrak{A} = (A, Q, B, \varphi, \psi, q_0) \in M_r$ в r-лабиринте G называем последовательность $(v_0, q_0, G_0), (v_1, q_1, G_1), \ldots$, такую, что v_0 — начальная вершина G; $G_0 = G$; $v_{i+1} = v_i \mathbf{pr}_1(\psi(q_i, \delta(v_i, G_i)))$ и $q_{i+1} = \varphi(q_i, \delta(v_i, G_i))$; G_{i+1} является r-лабиринтом, который получается из лабиринта G_i заменой отметки дуги, которая идет из v_i в v_{i+1}, на $\mathbf{pr}_2(\psi(q_i, \delta(v_i, G_i)))$. Последовательность вершин $v_0, v_1, \ldots$ назовем *V-траекторией*, последовательность состояний $q_0, q_1, \ldots$ — *Q-траекторией*, последовательность r-лабиринтов $G_0, G_1, \ldots$ — *G-траекторией*, а последовательность пар $(v_0, q_0), (v_1, q_1), \ldots$ — *VG-траекторией* r-автомата $\mathfrak{A}$. Если V-траектория r-автомата $\mathfrak{A}$ содержит все вершины множества VG, то говорим, что $\mathfrak{A}$ *обходит* лабиринт G, в противном случае G является *ловушкой* для $\mathfrak{A}$.

2. Описание r-автоматов с помощью операторов программы АвтЛаб

Внутреннее представление r-автоматов в программе АвтЛаб осуществляется в виде соответствующих массивов. Такое представление отвечает стандартному представлению автоматов с помощью прямоугольных таблиц с двумя входами.

Пусть $V = (A, Q, B, \varphi, \psi, q_0)$ автомат из M_r. Предположим, что элементы множества A (Q) упорядочены в некоторую конечную последовательность и обозначим через $n(a)$ ($n(q)$) порядковый номер

элемента $a \in A$ ($q \in Q$) в этой последовательности.

Также введем следующее обозначение. Пусть $a \in A$ и $\omega \in D$. Тогда, если существует $k \in \overline{0,r}$, удовлетворяющее условию $(\omega, k) \in a$, то положим $a|_\omega = k$, а если такого k не существует, то $a|_\omega = -1$.

Значение r хранится в целочисленной переменной `r`. Множество A в программе АвтЛаб задается двумерным массивом `A`, у которого в качестве типа первого индекса выступает ограниченный тип $1\,..\,(r+2)^4-1$, а в качестве типа второго индекса — перечислимый тип `(e,s,w,n)`. Элементами массива `A` являются значения типа $0\,..\,r+1$. Любое $a \in A$ представлено элементами `A[`$n(a)$`,e]`, `A[`$n(a)$`,s]`, `A[`$n(a)$`,w]` и `A[`$n(a)$`,n]` массива `A` так, что `A[`$n(a)$`,j]` $= a|_\omega$, где $\omega = \iota$(`j`), для любого значения второго индекса `j`; здесь ι(`e`) $= e$, ι(`s`) $= s$, ι(`w`) $= w$ и ι(`n`) $= n$. Максимальное значение, которое может принять `r` в программе АвтЛаб, есть 8, поэтому под кодом буквы $a \in A$ можно понимать десятичное число $\Delta(a) = \overline{\texttt{A[n(a),e]A[n(a),s]A[n(a),w]A[n(a),n]}}$, а также можно предположить, что $\Delta(a_1) < \Delta(a_2)$, если $n(a_1) < n(a_2)$ для любых $a_1, a_2 \in A$. Массив `A` назовем *входным* массивом автомата $\mathfrak{A}$; как только определено значение `r`, программа сама заполняет этот массив.

Любое состояние $q \in Q$ кодируется программой числом $n(q)$.

Функции φ и ψ в программе АвтЛаб задаются вместе одним двумерным массивом `B`, элементами которого являются значения комбинированного типа

`record dr:(e,s,w,n); lb:1..`r`; st:1..`$|Q|$ `end;`

в качестве типа первого индекса этого двумерного массива выступает ограниченный тип $1\,..\,(r+2)^4-1$, а в качестве типа второго индекса — ограниченный тип $1\,..\,|Q|$. Смысл индексов следующий: первый индекс массива принимает значение порядкового номера входа, а второй — порядкового номера состояния автомата. Массив `B` называется *выходным* массивом автомата $\mathfrak{A}$, и он *определяет* автомат $\mathfrak{A}$, если

`B[`$n(a)$`,`$n(q)$`].dr` $= \iota^{-1}(\mathbf{pr}_1(\psi(q,a)))$,
`B[`$n(a)$`,`$n(q)$`].lb` $= \mathbf{pr}_2(\psi(q,a))$,

и

`B[`$n(a)$`,`$n(q)$`].st` $= n(\varphi(q,a))$

для любых $a \in A$ и $q \in Q$. В ниже приведенной таблице даны соответствующие массивы для автомата V из $M_{0,4}$, который обходит все 0-деревья, по правилу «левой руки».

		$n(q)=1$	2	3	4
$n(a)=1$	$\Delta(a)=0001$	n04	n04	n04	n04
2	0010	w03	w03	w03	w03
3	0011	n04	w03	w03	w03
4	0100	s02	s02	s02	s02
5	0101	n04	s02	s02	n04
6	0110	s02	s02	w03	w03
7	0111	n04	s02	s02	w03
8	1000	e01	e01	e01	e01
9	1001	n04	e01	n04	n04
10	1010	e01	e01	w03	w03
11	1011	n04	e01	w03	e01
12	1100	e01	e01	s02	e01
13	1101	n04	e01	s02	n04
14	1110	e01	e01	s02	w03
15	1111	n04	e01	s02	w03

Система АвтЛаб позволяет заполнение массива B поэлементно, но этот прием является трудоемким, поэтому в программе АвтЛаб существует и другой, более удобный способ определения автоматов с помощью так называемых АвтЛаб операторов. Их несколько — это два оператора направления, оператор состояния и два оператора отметки. Ниже дадим описание этих операторов, но сначала введем некоторые необходимые обозначения.

Пусть $a \in A$, $\omega \in \mathbf{pr}_1(a)$ и $k \in \overline{0,r}$. Обозначим через σ — циклическую перестановку (e,s,w,n). Тогда, $\omega^+ = \omega^+(a) = \sigma^{i_0}(\omega)$, где $i_0 = \min\{i \in \overline{1,4} \,|\, \sigma^i(\omega) \in \mathbf{pr}_1(a)\}$, и $\omega^- = \omega^-(a) = \sigma^{-i_1}(\omega)$, где $i_1 = \min\{i \in \overline{1,4} \,|\, \sigma^{-i}(\omega) \in \mathbf{pr}_1(a)\}$. Также, если $k \in \mathbf{pr}_2(a)$, то пусть $\omega_k^+ = \omega_k^+(a) = \sigma^{i_0}(\omega)$, где $i_0 = \min\{i \in \overline{1,4} \,|\, \sigma^i(\omega) \in \mathbf{pr}_1(a) \wedge (\sigma^i(\omega),k) \in a\}$, и $\omega_k^- = \omega_k^-(a) = \sigma^{-i_1}(\omega)$, где $i_1 = \min\{i \in \overline{1,4} \,|\, \sigma^{-i}(\omega) \in \mathbf{pr}_1(a) \wedge (\sigma^{-i}(\omega),k) \in a\}$; если $k \notin \mathbf{pr}_2(a)$, то значения ω_k^+ и ω_k^- не определены.

Синтаксис операторов программы АвтЛаб опишем с помощью формальной системы обозначений Бэкуса [1] (здесь используем вариант этой системы).

Оператор направления `p` имеет следующий вид:

<оператор направления p> ::= p(<сп_условий>)<сп_действий>
<сп_условий> ::= <сп_условий1>|<сп_условий2>
<сп_условий1> ::= <условие1>|<условие1><сп_условий1>
<сп_условий2> ::= <условие2>|<условие2><сп_условий1>
<условие1> ::= <напр><знак><отношение><список_отм>
<условие2> ::= x<отношение><сп_отметок>
<напр> ::= e|s|w|n
<знак> ::= −| + |<пусто>
<отношение> ::= <| > | <= | >= | = | <>
<сп_отметок> ::= список i, $1 \leqslant i \leqslant r$, различных значений типа $0\,..\,r$ (отметки), разделенных запятыми
<пусто> ::=
<сп_действий> ::= <сп_действий1>|<н_о_с1>
<сп_действий1> ::= <н_о_с>|<н_о_с><сп_действий1>
<н_о_с> ::= <сост><напр><знак><отм><сост1>
<н_о_с1> ::= x<отм><сост1>
<отм> ::= (<отм1>)|<пусто>
<сост1> ::= [<сост2>]|<пусто>
<сост2> ::= значение типа $1\,..\,|Q|$
<отм1> ::= значение типа $1\,..\,r$

Пусть $a \in A$, $\omega \in D$, $k_1, \ldots, k_t$ — произвольные числа множества $\overline{-1, r}$ такие, что $t \in \overline{1, r+2}$ и $k_i \neq k_j$ для любых $i, j \in \overline{1, t}$, $i \neq j$, и $*$ — одно из отношений $\leqslant$, $<$, $\geqslant$, $>$, $\neq$ и $=$. Тогда через $\omega * k_1, \ldots, k_t$ обозначаем формулу $(\exists k \in \overline{-1, r})(\omega, k) \in a \wedge (k * k_1 \vee \cdots \vee k * k_t)$; если она выполнена, то говорим, что a удовлетворяет условию $\omega * k_1, \ldots, k_t$; здесь через $(\omega, -1) \in a$ обозначено условие $\omega \notin a$. Также через $\omega^+ * k_1, \ldots, k_t$ $(\omega^- * k_1, \ldots, k_t)$ обозначаем формулу $k_0 * k_1 \vee \cdots \vee k_0 * k_t$, где k_0 такое, что $(\omega^+, k_0) \in a$ $((\omega^-, k_0) \in a)$; если она выполнена, то говорим, что a удовлетворяет условию $\omega^+ * k_1, \ldots, k_t$ $(\omega^- * k_1, \ldots, k_t)$. Наконец, через $x * k_1, \ldots, k_t$ обозначаем формулу $(\forall k \in \overline{0, r})(\forall \omega \in D)(\omega, k) \in a \to k * k_1 \vee \cdots \vee k * k_t$.

Теперь ясно, каким способом с помощью синтаксических категорий <сп_условий1> и <сп_условий2> задаем конъюнкцию только что описанных условий и таким образом определяем некоторый предикат P на множестве A; обозначим область истинности этого предиката через P_A. Теперь объясним «функционирование» оператора `p`. При этом примем следующее соглашение: повсюду ниже, где в выражениях появляется имя некоторой синтаксической категории, на самом деле, имеется в виду некоторый терминальный символ, являющийся возможным значением этой категории.

Смысл «элементарного действия», которое задается синтаксической категорией <н_о_с>, состоит в следующем. Для любого $a \in \mathrm{P}_A$ программа должна выполнить следующую процедуру. Пусть `i` $:=$ $n(a)$ и `j` $:=$ <сост>. Если $a|_{\text{<напр>}} = -1$, то программа не предпринимает никаких действий, а если $a|_{\text{<напр>}} \geqslant 0$, то:

а) `B(i,j).dr:=`<напр>;

б) `B(i,j).lb:=`<отм>, если <отм> $\neq$ <пусто>,
и `B(i,j).lb:=A[i,`<напр>`]` в противном случае;

в) `B(i,j).st:=`<сост1>, если <сост1> $\neq$ <пусто>, и `B(i,j).st` не определяется в противном случае.

Пусть $\omega_1, \ldots, \omega_m$ — список всех букв из множества $\{\mathtt{e}, \mathtt{s}, \mathtt{w}, \mathtt{n}\}$, которые в таком порядке появляются в строке, определяющей синтаксическую категорию <сп_условий>. Также, пусть $m(a) = \max\{t \in \overline{1,m} \,|\, (\exists k \in \overline{0,r})(\omega_t, k) \in a\}$ и $k(a) \in \overline{0,r}$ такое, что $(\omega_{m(a)}, k(a)) \in a$. Тогда смысл «элементарного действия», которое задается синтаксической категорией <н_о_с1>, следующий. Для любого $a \in \mathrm{P}_A$ и любого значения индекса `j` программа определяет `i` $:= n(a)$ и выполняет следующие действия:

а) `B(i,j).dr` $:= \iota^{-1}(\omega_{m(a)})$, если существует $\omega_{m(a)}$, и `B(i,j).dr` не определяется в противном случае;

б) `B(i,j).lb` $:=$ <отметка>, если $\omega_{m(a)}$ и <отметка> существуют, и `B(i,j).lb` $:= k(a)$ в противном случае;

в) `B(i,j).st:=`<сост1>, если $\omega_{m(a)}$ и <сост1> существуют, и `B(i,j).st` не определяется в противном случае.

Дадим несколько примеров.

1) `p(w=2,w+=0)1w+(1)2w`. Тогда $\mathrm{P}(a) = ((w,2) \in a) \wedge ((w^+,0) \in a)$, и программа выполняет следующие действия: `B[`$n(a)$`,1].dr:=`$\mathtt{w}^+$,

B[$n(a)$,1].lb:=1, B[$n(a)$,2].dr:=w и B[$n(a)$,2].lb:=2.

2) p(x=1,e=>0)1e. Тогда $\mathrm{P}(a) = (\mathbf{pr}_2(a) = \{1\}) \wedge (e \in \mathbf{pr}_1(a))$, и программа выполняет следующие действия: B[$n(a)$,1].dr:=e и B[$n(a)$,1].lb:=1.

3) p(x=>1,n≠1,e≠1,s≠1,s+=1)x. Тогда $\mathrm{P}(a) = \mathbf{pr}_2(a) \subseteq \overline{1,r} \wedge (n,1) \notin a \wedge (e,1) \notin a \wedge (s,1) \notin a \wedge (s^+,1) \in a$, и программа выполняет следующие действия: B[$n(a)$,1].dr:=w и B[$n(a)$,1].lb:=1.

4) p(e=12,w≠2)x(3). Тогда $\mathrm{P}(a) = ((e,1) \in a \vee (e,2) \in a) \wedge (w,2) \notin a$, и программа для любого значения индекса j выполняет следующие действия: B[$n(a)$,j].dr:=w, если $w \in \mathbf{pr}_1(a)$, или B[$n(a)$,j].dr:=e, если $w \notin \mathbf{pr}_1(a)$, и B[$n(a)$,j].lb:=3.

5) p(e>0,n≠1)1e+(1)[2]2s-(2). Тогда $\mathrm{P}(a) = a|_e > 0 \wedge (n,1) \notin a$, и программа выполняет следующие действия: B[$n(a)$,1].dr := $\iota^{-1}(e^+)$, B[$n(a)$,1].lb:=1, B[$n(a)$,1].st:=2, B[$n(a)$,2].dr := $\iota^{-1}(s^-)$ и B[$n(a)$,2].lb:=2.

Из определения оператора p непосредственно следует, что только с помощью него можно описать любой r-автомат.

Оператор направления P имеет следующий вид:

<оператор направления P> ::= P(<напр3><зн>, <напр4><зн>)
<напр3> ::= e|s|w|n
<напр4> ::= e|s|w|n|<пусто>
<зн> ::= −|+.

Функционирование оператора P объясним на следующих двух примерах:

1) P(e+,w-). Для любого $a \in A$ сначала определяется значение $k(a) = a|_{e^+}$, потом определяется значение $\omega(a) = w^-_{k(a)}$. Программа выполняет B[$n(a)$,j].dr := $\iota^{-1}(\omega(a))$ для всех значений индекса j.

2) P(n-,-). Для любого $a \in A$ сначала определяется $k(a) = a|_{n^-}$, потом определяется значение $\omega(a) = (n^-)^-_{k(a)}$. Программа выполняет B[$n(a)$,j].dr := $\iota^{-1}(\omega(a))$ для всех значений индекса j.

Оператор состояния S имеет вид:

<оператор состояния S> ::= S(<стр>)
<стр> ::= <напр>, <зона_отм>, <зона_сост>|x, <отм>, <сост>
<напр> ::= e|s|w|n

<зона_отм> ::= <отм>|u(<отм1><отм2>)
<отм>, <отм1>, <отм2> ::= $1\,..\,r$
<зона_сост> ::= <сост>|<сост1>, <сост2>
<сост>, <сост1>, <сост2> ::= *|$1\,..\,|Q|$

Оператором S заполняется массив B таким способом, что определяется поле B[i,j].st для тех записей B[i,j], у которых первое и второе поля уже определены. Делается это следующим образом. Примем следующее соглашение: если пишем, что B[i,j].st = <состояние>, где <состояние> — одна из категорий <сост>, <сост1> или <сост2>, и $m \in \overline{1,|Q|} \cup \{*\}$ — значение категории <состояние>, то, на самом деле, полагаем, что B[i,j].st $= m$, если $m \in \overline{1,|Q|}$, и B[i,j].st=j, если $m = *$. Тогда:

а) если <стр> = x,<отм>,<сост>, то для любых значений индексов i и j таких, что B[i,j].lb=<отм>, программа выполняет B[i,j].st := <сост>.

б) если <стр> = <напр>,<отм>,<сост> и <напр> принадлежит множеству {e,s,w,n}, то для любых значений индексов i и j таких, что B[i,j].dr=<напр> и B[i,j].lb=<отм>, программа выполняет B[i,j].st := <сост>.

в) если <стр> = <напр>, u(<отм1><отм2>),<сост1>,<сост2> и B[$n(a)$,j].dr=<напр>, то в случае, когда B[$n(a)$,j].lb=<отм1>, $a|_\omega$ = <отм2> для любого $\omega \in \mathbf{pr}_1(a)\backslash\{$<напр>$\}$ и $|\mathbf{pr}_1(a)\backslash\{$<напр>$\}| \geqslant 1$, программа выполняет B[i,j].st := <сост1>, и выполняет B[i,j].st := <сост2> в противном случае.

Например, если $\Delta(a) = 0211$, B($n(a)$,1).dr=s и B($n(a)$,1).lb=1, то оператор S(s,u(10),=2) определяет действие B($n(a)$,1).st:=1.

Оператор отметки o имеет вид:

<оператор отметки o> ::= o(<отм1>,<отм2>)
<отм1>, <отм2> ::= $1\,..\,r$

Функционирование оператора o состоит в следующем: для всех $a \in A$ и всех значений индекса j, у которых $a|_\omega$ = <отм1>, где ω = B[$n(a)$,j].dr, программа выполняет B[$n(a)$,j].lb := <отм2>.

Оператор отметки O имеет следующий вид:

<оператор отметки O> ::= O(<вар1>)|O(<вар2>)
<вар1> ::= O(<сп_н_о>, -, <отм1>)

<вар2> ::= O(<сп_н_о>, o(<отм1>), <отм2><отм3>)
<сп_н_о> ::= <напр1><сп_отм1>|
<напр1><сп_отм1><напр2><сп_отм2>
<напр1>, <напр2> ::= e|s|w|n
<сп_отм1>, <сп_отм2> ::= список i, $1 \leqslant i \leqslant r$, различных значений типа $0\,..\,r$ (отметки), разделенных запятыми
<отм1> ::= значение типа $0\,..\,r$
<отм2>, <отм3> ::= $*$|значение типа $0\,..\,r$

Функционирование оператора O состоит в следующем: для всех $a \in A$ и всех значений индекса j, у которых a удовлетворяет условию <сп_н_о> (в смысле, приведенном при описании оператора p, при этом считаем, что между <напр1> и <сп_отм1>, а также между <напр2> и <сп_отм2>, находится знак равенства) и у которых B[$n(a)$,j].dr = <напр1>, программа проделывает следующую процедуру:

а) если <условие> = -, то программа выполняет B[$n(a)$,j].lb := <отм2>, если <отм2> $\neq *$, и B[$n(a)$,j].lb:=$a|_{\text{<напр1>}}$ в противном случае;

б) если <условие> = o(<отм1>), то программа выполняет B[$n(a)$,j].lb:=<отм2>, если <отм1>$\in \mathbf{pr}_2(a\backslash\{(\text{<напр1>}, a|_{\text{<напр1>}})\})$, и B[$n(a)$,j].lb:=<отм3> в противном случае.

Например, оператор O(e123n1,-,=) работает следующим способом: для всех a, удовлетворяющих условиям $e = 123$ и $n = 1$, и для всех значений индекса j таких, что B[$n(a)$,j].dr $= e$, программа выполняет B[$n(a)$,j].lb:=$a|_e$. Оператор O(s1w2,o(1),12) работает следующим способом: для всех a, удовлетворяющих условиям $s = 1$ и $w = 2$, и для всех значений индекса j таких, что B[$n(a)$,j].dr $= e$, программа выполняет B[$n(a)$,j].lb := 1, если $1 \in \mathbf{pr}_2(a\backslash\{(s,1)\})$, и B[$n(a)$,j].lb := 2, если $1 \notin \mathbf{pr}_2(a\backslash\{(s,1)\})$.

Использование вышеописанных операторов может привести к переопределению выходного массива: программа предпринимает попытку присвоить новое значение некоторому из его элементов. Эта проблема в программе АвтЛаб преодолена за счет использования трех возможностей, которые предоставляются пользователю в диалоговом режиме и состоят в следующем:

• не вписывать новый выходной код, то есть оставить переменную выход в том виде, в каком она была до дублирования;

• вписать новый выходной код, то есть приписать переменной выход появившееся сочетание;

• вписать полностью новый выходной код.

3. Моделирование поведения автомата в программе АвтЛаб

Моделирование поведения r-автоматов в r-лабиринтах осуществляется в четыре этапа. На первом этапе задается автомат с помощью АвтЛаб операторов, на основании которых программа формирует соответствующие массивы (внутреннее описание автомата), то есть соответствующее АвтЛаб-описание данного автомата, или загружается уже имеющееся описание из памяти компьютера. На втором этапе задается произвольный (односторонний или двухсторонний) r-лабиринт или соответствующее его описание загружается из заранее подготовленного файла. На третьем этапе для заданных r-автомата и r-лабиринта даются начальные условия, то есть дается начальная вершина лабиринта и начальное состояние автомата. На четвертом этапе программа АвтЛаб позволяет с помощью удобного интерфейса на мониторе компьютера следить за движением данного автомата в заданном лабиринте. Работая в автономном режиме, она отвечает на вопрос, обходит ли заданный автомат заданный лабиринт или нет (если не обходит, то на каком шагу происходит зацикливание; если обходит, то выдает время обхода, и т.п.). Проиллюстрируем сказанное следующими двумя примерами:

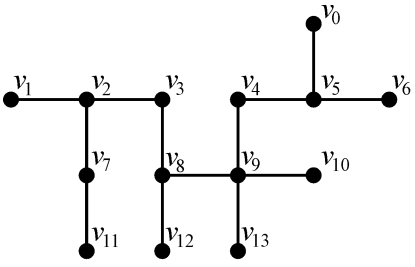

Рис. 1.

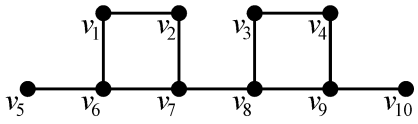

Рис. 2.

1) Пусть $\mathfrak{A}$ — автомат из $M_{0,4}$, заданный в программе АвтЛаб следующим способом:

```
p(w+=0)1w+2n+3e+4s+; s(e,0,1); s(s,0,2); s(w,0,3); s(n,0,4).
```

Если, например, автомат $\mathfrak{A}$ в начальный момент находится в состоянии q_1 и в вершине v_8 лабиринта G, представленного на рис. 1, то его VQ-траекторией является последовательность (v_8, q_1), (v_3, q_4), (v_2, q_3), (v_7, q_2), (v_{11}, q_2), (v_7, q_4), (v_2, q_4), (v_1, q_3), (v_2, q_1), (v_3, q_1), (v_8, q_2), (v_9, q_1), (v_4, q_4), (v_5, q_1), (v_0, q_4), (v_5, q_2), (v_6, q_1), (v_5, q_3), (v_4, q_3), (v_9, q_2), (v_{10}, q_1), (v_9, q_3), (v_{13}, q_2), (v_9, q_4), (v_8, q_3), (v_{12}, q_2).

2) Пусть $\mathfrak{A}$ — автомат из $M_{1,2}$, который в программе АвтЛаб задается следующим образом:

```
P(w-,+); o(01,2); o(2,1); s(x,0,1); s(x,1,1); s(x,2,1).
```

Нетрудно удостовериться, что этот автомат является универсальным для класса всех 1-лабиринтов. Так, например, если автомат $\mathfrak{A}$ в начальный момент находится в состоянии q_1 и в вершине v_5 лабиринта G, представленного на рис. 2, то его V-траекторией является последовательность v_5, v_6, v_5, v_6, v_1, v_2, v_1, v_6, v_7, v_6, v_5, v_6, v_1, v_2, v_7, v_2, v_1, v_6, v_7, v_8, v_7, v_6, v_5, v_6, v_1, v_2, v_7, v_2, v_1, v_6, v_7, v_8, v_3, v_4, v_3, v_8, v_9, v_8, v_7, v_6, v_5, v_6, v_1, v_2, v_7, v_2, v_1, v_6, v_7, v_8, v_3, v_4, v_9, v_4, v_3, v_8, v_9, v_{10}.

Список литературы

[1] Backus J.W. The History of Fortran I, II and III // History of Programming Languages. New York, 1981.

[2] Kudryavtsev V.B., Ushchumlich Sh., Kilibarda G. On behavior of automata in labyrinths // Discrete Math. Appl. Vol. 3. No. 1. P. 1–28. 1993.

[3] Кудрявцев Г.Ю. О времени решения лабиринтных задач конечными автоматами. Дисс. канд. физ.-мат. наук. Саратов, 1990.

[4] Kurepa J.A. Modeliranje ponašanja automata u jednoj klasi lavirinata. Beograd: Magistarski rad, 2000.