

Майер Р.В. АВТОМАТЫ И ИХ ОБУЧЕНИЕ

Нас окружают *автоматы* — дискретные устройства, выполняющие заданную последовательность действий, в результате чего происходит преобразование информации, материальных объектов или энергии. К ним относятся светофор, телефонный аппарат, калькулятор, микроволновая печь, стиральная машина, лифт, телевизор с пультом дистанционного управления, компьютер и т.д. Для анализа работы этих устройств используют математическую модель — абстрактный автомат, имеющий *множество внутренних состояний* $Q = \{q_1, q_2, \dots, q_K\}$ и преобразующий *входные сигналы (символы)* $X = \{x_1, x_2, \dots, x_N\}$ в *выходные сигналы (символы)* $Y = \{y_1, y_2, \dots, y_M\}$ в соответствии с определенными правилами. Эффективный метод исследования работы автоматов предполагает создание его компьютерной модели на ПЭВМ.

Автоматы без памяти и с памятью. Множество внутренних состояний Q образует *внутреннюю память* автомата. Если автомат имеет только одно внутреннее состояние, то он называется *автоматом без памяти*. Такие автоматы не меняют своего поведения: выходной сигнал (символ) определяется только входным и не зависит от ранее поступивших сигналов (символов) в предыдущие моменты времени: $y_j^t = \theta(x_i^t)$. К автоматам без памяти относятся: 1) цепь из источника, переключателя и лампочек; 2) комбинационная схема, состоящая из логических элементов И, ИЛИ, НЕ и не содержащая триггеров; 3) кодер, осуществляющий побуквенный перевод входного сообщения и т.д.

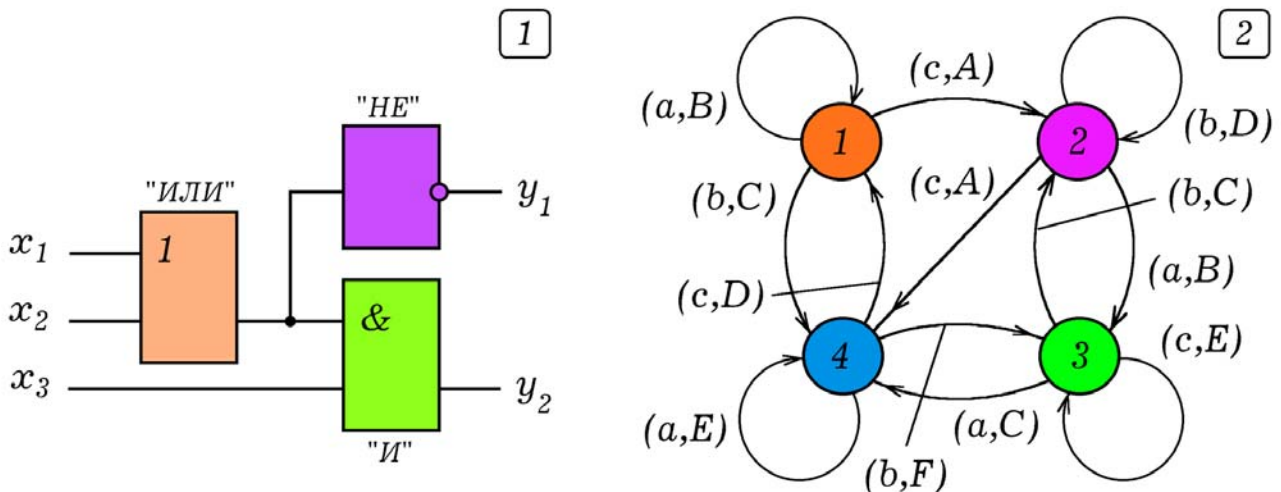


Рис. 1.

Рассмотрим комбинационную схему с тремя входами и двумя выходами (рис. 1.1). Состояния ее выходов, описываются логическими функциями: $y_1 = \text{not}(x_1 \text{ or } x_2)$, $y_2 = (x_1 \text{ or } x_2) \text{ and } x_3$. Чтобы исследовать работу этой цифровой схемы, наберем в среде Borland Pascal 7.0 программу ПР – 1. В ней

перебираются всевозможные состояния входов x_1 , x_2 , x_3 и определяются состояния выходов y_1 и y_2 . На экране монитора появляется *таблица истинности*.

```
===== ПР-1 =====
uses crt;                                     {Borland Pascal 7.0}
var x1,x2,x3,y1,y2: boolean;
BEGIN
For x1:=false to true do
  For x2:=false to true do
    For x3:=false to true do
      begin
        y1:=not((x1)or(x2));
        y2:=((x1)or(x2))and(x3);
        writeln(x1,' ',x2,' ',x3,' ',y1,' ',y2);
      end; ReadKey;
END.
```

Если автомат имеет два или более внутренних состояния, то он называется *автоматом с памятью*. Рассмотрим автомат с четырьмя внутренними состояниями 1, 2, 3, 4, входным алфавитом $X = \{a, b, c\}$ и выходным алфавитом $Y = \{A, B, C, D, E, F\}$. Для задания правил переключения автомата будем использовать *диаграмму Мура*, представляющую собой ориентированный граф (рис. 1.2). Вершины графа изображают состояния автомата $Q = \{1, 2, 3, 4\}$, а ребра — переходы из одного состояния в другое. Каждой команде $q_i x_r \Rightarrow q_j x_s$ соответствует ребро, идущее из q_i в q_j и называемое (x_r, x_s) . Команда “2c \Rightarrow 4A” означает, что если автомат находится в состоянии 2 и на его вход приходит символ “c”, то он переходит в состояние 4 и на его выходе появляется символ “A”.

Промоделировать работу этого автомата на компьютере можно с помощью программы ПР – 2. Пусть на вход поступает последовательность символов “babacaabcbabcbcbbaaccbcbabcbccaacbb”. Программа последовательно перебирает входные символы, каждый раз определяя новое состояние автомата и символ на его выходе. Результаты выводятся на экран.

```
===== ПР-2 =====
uses crt;                                     {Borland Pascal 7.0}
var i: integer; x,y,S,a,b,c: string;
    q: integer; label m;
BEGIN S:='babacaabcbabcbcbbaaccbcbabcbccaacbb';
      writeln('S=',S);q:=1;
      For i:=1 to length(S) do begin x:=copy(S,i,1);
        If (x='a')and(q=1) then begin q:=1; y:='B'; goto m; end;
        If (x='c')and(q=1) then begin q:=2; y:='A'; goto m; end;
        If (x='b')and(q=1) then begin q:=4; y:='C'; goto m; end;
        If (x='b')and(q=2) then begin q:=2; y:='D'; goto m; end;
        If (x='a')and(q=2) then begin q:=3; y:='B'; goto m; end;
        If (x='c')and(q=2) then begin q:=4; y:='A'; goto m; end;
        If (x='a')and(q=3) then begin q:=4; y:='C'; goto m; end;
        If (x='c')and(q=3) then begin q:=3; y:='E'; goto m; end;
```

```

If (x='b')and(q=3) then begin q:=2; y:='C'; goto m; end;
If (x='a')and(q=4) then begin q:=4; y:='E'; goto m; end;
If (x='b')and(q=4) then begin q:=3; y:='F'; goto m; end;
If (x='c')and(q=4) then begin q:=1; y:='D'; goto m; end;
m: write(q, ' ', y, ' | ');
end; ReadKey;
END.

```

Вероятностные автоматы. Все сказанное выше относится к *детерминированным автоматам*, у которых внутреннее состояние и выходной символ в следующий момент времени $t + 1$ однозначно связаны с состоянием и входным символом в предыдущий момент t . Также существуют *недетерминированные* или *вероятностные автоматы*; для них переход из одного состояния в другое и появление символа на выходе — случайный процесс, происходящий с некоторой вероятностью. Вероятностный автомат работает так: если он находится в состоянии 3 и на его вход поступает символ “b”, то с вероятностью 0,2 автомат переходит в состояние 1 и выдает символ “A” на выходе; с вероятностью 0,3 переходит в состояние 2 и выдает символ “C”; с вероятностью 0,5 автомат переходит в состояние 4, на выходе появляется символ “B”.

Вероятностный автомат плохо предсказуем и в этом смысле похож на живой организм. Поэтому часто говорят о *поведении* вероятностного автомата, входные сигналы называют *стимулами*, а выходные — *реакциями*. Если входной алфавит состоит из N символов (стимулов), а выходной — из M символов (реакций), то поведение вероятностного автомата с одним внутренним состоянием определяется *матрицей вероятностей* размером n на m . Ее элементами являются вероятности p_{ij} появления j – ой реакции на i – ый стимул. Если множество $Y = \{y_1, y_2, \dots, y_M\}$ содержит полный набор всех реакций, то сумма соответствующих им вероятностей на каждый стимул (входной сигнал) x_i равна 1. То есть при любом i должно выполняться равенство: $p_{i1} + p_{i2} + \dots + p_{iM} = 1$.

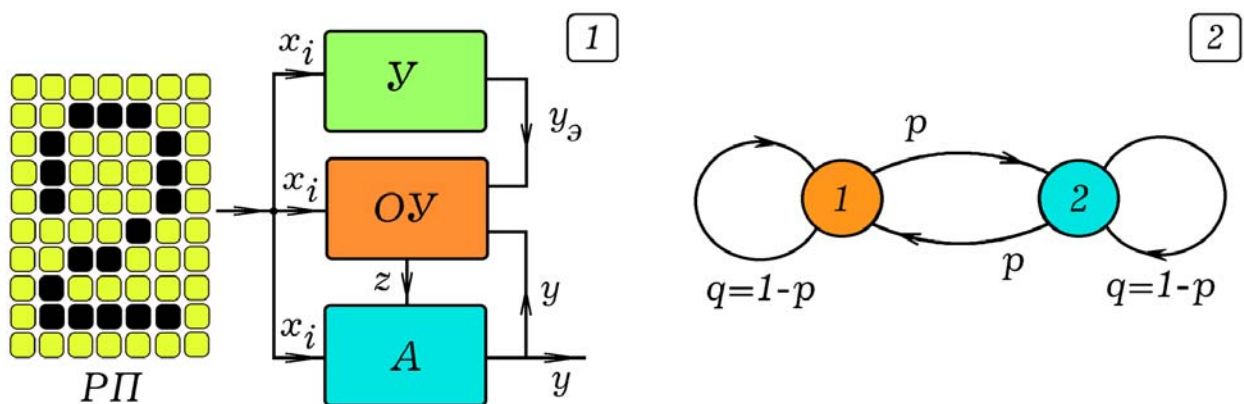


Рис. 2.

Обучение распознаванию образов. Вероятностный автомат можно научить *распознавать образы объектов*. Пусть имеется множество O из S

объектов: $O = \{o_1, o_2, \dots, o_S\}$, принадлежащее к небольшому числу классов K_1, K_2, \dots, K_R . Автомат (например, ЭВМ) правильно решает задачу распознавания, если он относит объект o_i к соответствующему классу K_j .

Рассмотрим обучение распознаванию изображений. Допустим имеется *рецепторное поле* в виде матрицы фотодиодов, на которую можно спроецировать изображение объекта (цифры или буквы). С целью обучения вероятностного автомата A на матрицу фотодиодов проецируются k изображений, случайно выбранных из множества O (рис. 2.1). *Учитель* $У$, роль которого играет человек или эталонный автомат, дает правильный ответ, показывая к какому классу относится каждый из k предъявленных объектов. *Обучающее устройство* $ОУ$ сравнивает реакции $у_{э}$ учителя $У$ с реакциями $у$ *обучаемого автомата* A и изменяет его матрицу вероятностей p_{ij} таким образом, чтобы реакции автомата A возможно чаще совпадали с реакциями учителя $У$. Затем автоматам предъявляется *экзаменационная последовательность объектов*. Если число ошибок автомата не превышает допустимого уровня, обучение закончено.

Обучение вероятностного автомата во многом похоже на обучение животного. При правильной реакции на стимул его "*поощряют*", а в случае ошибки — "*наказывают*". В результате изменяется матрица вероятностей — автомат "*запоминает*" урок.

Формирование навыка. Как происходит обучение человека? Допустим, ученик должен научиться выполнять определенную последовательность операций: операция 1, затем операция 2, после этого операция 3, затем снова операция 1 и т.д. Формирование навыка у человека (животного) происходит аналогично обучению автомата. Чтобы исследовать этот процесс, заменим человека *абстрактной моделью ученика* (АМУ), то есть некоторым воображаемым вероятностным автоматом с тремя состояниями, которые соответствуют операциям 1, 2 и 3. До обучения АМУ случайным образом выполняет различные операции, совершая при этом ошибки. Алгоритм работы АМУ можно задать в виде стохастического (вероятностного) графа — совокупности вершин, соединенных стрелками, которые соответствуют переходам от одной операции к другой (как на рис. 2). Вероятности p_{ij} перехода от i -ой операции к j -ой образуют двумерную матрицу вероятностей. Если автомат не обучен, то вероятности всех переходов равны:

$$p_{ij} = \begin{pmatrix} 0,33 & 0,33 & 0,33 \\ 0,33 & 0,33 & 0,33 \\ 0,33 & 0,33 & 0,33 \end{pmatrix},$$

то есть он выбирает каждую следующую операцию совершенно произвольно.

Учитель использует "*метод кнута и пряника*". Если АМУ совершила правильное действие, то его "поощряют" высокой оценкой 1, в результате чего вероятность повторения этого действия увеличивается, а остальных —

уменьшается. В случае ошибки АМУ "наказывают" оценкой 0, что приводит к уменьшению вероятности ее повторения. В результате обучения вероятности правильных переходов p_{12} , p_{23} , p_{31} возрастают, стремясь к 1, а вероятности остальных ошибочных действий — уменьшаются, приближаясь к 0. Матрица вероятностей приобретает вид:

$$p_{ij} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

В конце обучения автомат практически без ошибок выполняет требуемую последовательность $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1 \dots$

Компьютерное моделирование обучения. Рассмотрим АМУ с двумя состояниями, соответствующими операциям 1 и 2 (рис. 2.1). Будем считать, что автомат обучен, когда из состояния 1 он переходит в состояние 2, а из состояния 2 — в состояние 1 и т.д.: $1 \Rightarrow 2 \Rightarrow 1 \Rightarrow 2 \Rightarrow 1 \Rightarrow 2 \Rightarrow \dots$ Если из состояния 1 автомат переходит в состояние 1, или из состояния 2 переходит в состояние 2, то он совершает ошибку. Вероятность правильного действия обозначим через p , тогда вероятность неправильного действия равна $q=1-p$. Можно изучать работу вероятностного автомата с большим числом состояний, но при этом всегда один из переходов будет правильным, а остальные — неправильными.

Чтобы промоделировать процесс обучения используется программа ПР-3. Изначально автомат необучен, вероятность правильного действия мала ($p=0,01$). Программа содержит цикл, в котором выбор каждой операции осуществляется с помощью генератора случайных чисел. Если случайное число a , находящееся в интервале $[0; 1]$, меньше p , то АМУ совершает правильное действие, если нет, — делает ошибку. Процесс обучения приводит к изменению матрицы вероятностей: вероятность правильного выбора p увеличивается на αq , где α — коэффициент научения, а вероятность ошибки уменьшается на ту же величину. Уровень сформированности навыка Z будем считать равным вероятности p выбора правильной операции.

```
===== ПР-3 =====
uses dos, crt, graph;                               {Borland Pascal 7.0}
var t,Gd,Gm : integer; x,p,q,a,g : real;
BEGIN
Randomize;
Gd:=Detect; InitGraph(Gd, Gm, 'c:\bp\bgi');
line(0,450,640,450); line(10,0,10,480);
p:=0.01; q:=1-p; a:=0.003; g:=0.0004;
Repeat inc(t); x:=random(1000)/1000;
  If (x>p)and(t<4000) then
    begin
      p:=p+a*q; q:=q-a*q;
    end;
  p:=p-g*p; q:=q+g*p;                               {забывание}
```

```

circle(10+round(t/15), 450-round(400*p), 2);
until (t>10000) or (KeyPressed);
Repeat until KeyPressed; CloseGraph;
END.

```

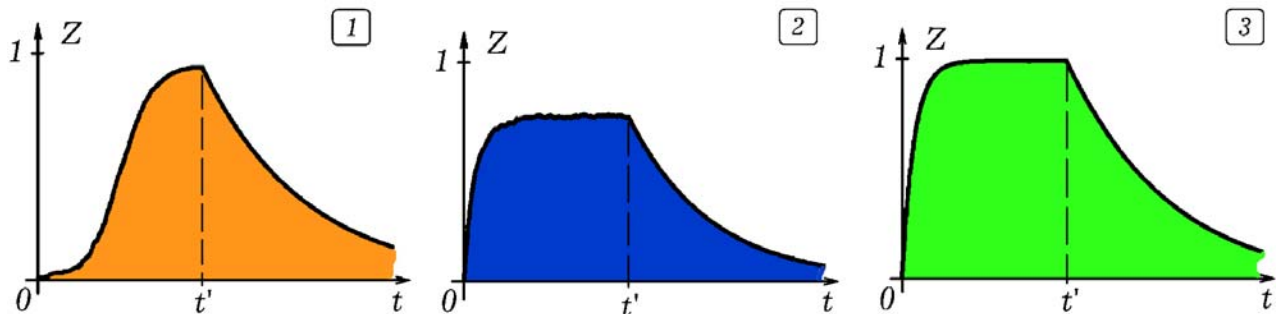


Рис. 3.

Учтем, что часть информации человек забывает. Для этого каждом временном шаге будем уменьшать вероятность правильного выбора операции p на γp , где γ — коэффициент забывания, и на такую же величину увеличивать вероятность ошибки q : $p := p - \gamma p$; $q := q + \gamma p$. При обучении автомата забывания не происходит: $\gamma = 0$.

Результаты работы программы представлены на рис. 3. Про моделированы следующие ситуации:

1. *Обучение с поощрением*: при выполнении правильного действия ученика "поощряют", пересчитывая вероятности p и q . Так как сначала ученик ошибается гораздо чаще (q превосходит p), то сначала обучение происходит медленно (рис. 3.1). Зато по мере увеличения "знаний" вероятность совершения правильного действия растет. Акты обучения происходят все чаще, вероятность p увеличивается до 1. Программа должна содержать оператор:

```

If (x<p) and (t<4000) then
begin p:=p+a*q; q:=q-a*q; end;

```

2. *Обучение с наказанием*: в случае ошибки ученика "наказывают", подсказывая ему правильный ответ, что приводит к росту p и уменьшению q . Сначала ученик ошибается часто, поэтому уровень его "знаний" быстро растет, вероятность ошибки q падает (рис. 3.2). Акты обучения происходят реже, уровень знаний за счет забывания не достигает 1. Программа должна содержать условный оператор:

```

If (x>p) and (t<4000) then
begin p:=p+a*q; q:=q-a*q; end;

```

3. *Обучение с поощрением и наказанием*: при правильном ответе ученика поощряют, а при неправильном наказывают, подсказывая правильный ответ. В обоих случаях вероятность правильного действия p растет, а вероятность ошибки q снижается. Так как при любом действии учащегося его учат, то уровень знаний быстро растет и достигает 1 (рис. 3.3). Программа должна содержать оператор:

```

If t<4000 then begin p:=p+a*q; q:=q-a*q; end;

```

В настоящей статье мы затронули лишь некоторые вопросы кибернетики и теоретической информатики. На первый взгляд кажется, что аналогия между автоматами и живыми организмами не достаточно обоснована. Однако это не так, — дальнейшее развитие компьютерной техники (электронных автоматов) неизбежно приведет к созданию *искусственного интеллекта*, сравнимого по мощности с человеческим мозгом. У некоторых людей даже может возникнуть комплекс неполноценности. Чтобы этого с Вами не произошло, надо решить несколько задач творческого характера:

Задание 1. На основе программы ПР – 1 создайте компьютерную модель автомата без памяти, реализующего следующую логическую функцию:

$$y = (x_1 \text{ and } (\text{not } x_2)) \text{ or } (\text{not } x_1).$$

Задание 2. Промоделируйте на ПЭВМ работу автомата без памяти, осуществляющего побуквенное кодирование сообщения из 20 символов, Алфавит содержит 16 букв, которым соответствуют коды: А – 0000, Б — 0001, В — 0010, ..., Н — 1111.

Задание 3. Используя программу ПР – 2, создайте компьютерную модель вероятностного автомата с тремя внутренними состояниями.

Задание 4. Используя программу ПР – 3, промоделируйте процесс обучения АМУ при различных коэффициентах научения и забывания.

Литература

1. Майер Р.В. Задачи, алгоритмы, программы // [Персональная страница Р.В.Майера] / ГГПИ [Глазов, 2010]. URL: <http://maier-rv.glazov.net> (<http://komp-model.narod.ru>) (дата обращения 25.06.10)
2. Отряшенков Ю.М. Юный кибернетик. — М. Дет. лит., 1978. — 447 с.
3. Попов В.Б. Turbo Pascal для школьников: Учеб. пособие. — М.: Финансы и статистика, 2001. — 528 с.
4. Черненко Г.Т. Простая автоматика: Рассказы об автоматике и автоматах–самоделках. — Л.: Дет. лит., 1989. — 127 с.