

**Майер Р.В.**  
**ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ИЗУЧЕНИЯ**  
**ГЕНЕТИЧЕСКИ СВЯЗАННЫХ ЭЛЕМЕНТОВ УЧЕБНОГО**  
**МАТЕРИАЛА НА ПВМ**

Проблема итеративного обучения, заключающегося в формировании знаний и навыков в результате многократного повторения определенной последовательности действий, достаточно подробно исследована, например, в [3]. Особый интерес представляет собой процесс обучения, при котором изучаемые элементы учебного материала (ЭУМ) связаны генетической связью. Например, запоминание однокоренных или похожих слов иностранного языка, овладение теорией, являющейся развитием ранее рассмотренных теорий, вывод следствий из теорем и т.д. Будем считать, что два объекта ЭУМ 1 и ЭУМ 2 связаны генетической связью, если один объект получается из второго путем добавления или удаления элементов. Допустим, ученик сначала изучил теорию А, затем, расширив свои знания, освоил теорию В, которая включает в себя теорию А как частный случай, после этого изучил теорию С, включающую в себя В и т.д. Например, ученик научился считать (ЭУМ 1), складывать целые числа (ЭУМ 2), складывать дроби (ЭУМ 3) и т.д. Очевидно, что этот случай нельзя свести к простому итеративному (механическому) обучению за счет многократного повторения. В настоящей работе рассмотрены результаты имитационного моделирования [4], в котором ученик заменяется дискретной и непрерывной моделями.

### **1. Дискретная модель ученика (ДМУ)**

При обучении в мозгу человека возникают новые психические образования: нервные центры, которые соединены связями. Некоторые связи углубляются, другие наоборот исчезают. Дискретной моделью ученика (ДМУ) может служить созданный на базе ЭВМ вероятностный автомат переменной структуры, у которого в процессе обучения увеличивается число внутренних состояний, соответствующих изучаемым понятиям, и изменяются вероятности переходов (то есть устанавливаются нужные и исчезают ненужные связи). Пусть ДМУ изучает ЭУМ А, что будем обозначать словом “a\_”, затем ЭУМ АВ (слово “ab\_”), являющийся развитием ЭУМ А, после этого — ЭУМ ABC (слово “abc\_”), который получается их предыдущего ЭУМ АВ и т.д. Сначала ДМУ не обучена и имеет большое число ( $N = 1000$ ) внутренних состояний, связанных между собой связями с вероятностями перехода  $p_{ij} = 1/N$ . Последнее состояние будем считать конечным  $N_{stop} = N$ , в него автомат попадает из любого

другого состояния, считав символ "\_", означающий конец слова. Необученный автомат при этом должен выдать ответ: "не понял", а обученный — значение входного слова.

Обучение должно происходить так. Сначала автомат находится в состоянии  $q = 0$ . На его вход поступает слово "ab\_". Автомат, считав букву "a" совершает переход  $0 \rightarrow 1$  в какое-то состояние, которому он автоматически присваивает номер  $q = 1$ . При этом вероятность  $p_{01}$  перехода  $0 \rightarrow 1$  увеличивается на  $\Delta p = \alpha(1 - p_{01})$ . Затем он считывает букву "b" и автоматически совершает переход в состояние, которому он присваивает номер  $q = 2$ , вероятность  $p_{12}$  перехода  $1 \rightarrow 2$  увеличивается на  $\Delta p = \alpha(1 - p_{12})$ . Когда автомат прочитывает последнюю букву "\_", он переходит в последнее состояние  $N_{stop} = N$ , вероятность этого перехода  $2 \rightarrow N$  увеличивается на  $\Delta p = \alpha(1 - p_{2N})$ . Переменной  $zn$  присваивается значение слова "ab\_", сообщаемое учителем (например, "реакція2").

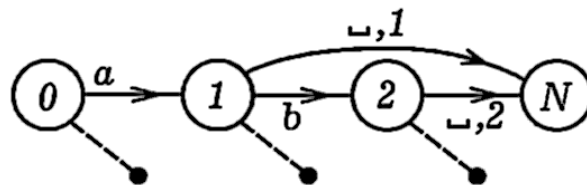


Рис. 1. Обучение дискретной модели ученика.

Допустим, ДМУ считывает слово "a\_". При этом автомат совершает переходы  $0 \rightarrow 1$  и  $1 \rightarrow N$ , их вероятности также увеличиваются. Если на вход автомата поступает слово "abc\_", а он уже усвоил слово "ab\_", то из состояния  $q = 2$  автомат, получив на вход "с", не переходит в состояние  $N_{stop}$ . Он создает другой переход в новое состояние  $q = 3$  (на рис. 1 не изображено). При поступлении символа "\_" автомат из состояния 3 переходит в состояние  $N_{stop}$ , запоминая значение слова "abc\_" ( $zn :=$  "реакція3"). Аналогично происходит обучение словам  $acb_$  и  $adea_$ , имеющим другие значения. На всех этапах обучения "учитель" контролирует работу "ученика", подкрепляя его действия: если ДМУ совершил правильный переход, то соответствующая вероятность увеличивается, а если нет — уменьшается. После многократного обучения ДМУ приобретет способность распознавать слова, поступающие на вход.

Теперь несколько упростим задачу. Пусть автомат ДМУ не самостоятельно создает новые состояния и связи между ними, а прислушивается к "учителю", сообщаящему в какое состояние и по какому переходу следует перейти после поступления на вход нового символа.

Вместо многократного повторения одних и тех же операций, поощрений и наказаний с последующим пересчетом вероятностей переходов, добавим в текст программы условный оператор, обеспечивающий переход автомата из одного состояния в другое. Это соответствует ситуации, когда учитель просто сообщает учащемуся последовательность действий или алгоритм, ведущий к решению задач данного типа. Ниже представлена программа ПР-1, моделирующая работу детерминированного автомата с заданным числом внутренних состояний, в которую можно будет добавлять условные операторы и таким образом создавать новые связи между различными состояниями.

```

uses crt, graph;                                     { ПР-1 }
const alfa=0.2;
var s1,s2,m,q,k,i,j: integer;
x,y:string; f: boolean; slovo: array[1..5] of string;
zn:array[1..5] of string; znn: string; Label m1,m2;
BEGIN clrscr;
slovo[1]:='a_';   zn[1]:='reakciya1';
slovo[2]:='ab_';  zn[2]:='reakciya2';
slovo[3]:='abc_'; zn[3]:='reakciya3';
slovo[4]:='abcd_';zn[4]:='reakciya4';
slovo[5]:='cdb_';zn[5]:='reakciya5'; m:=5; q:=0;
For j:=1 to length(slovo[m]) do begin y:=copy(slovo[m],j,1);
If (q=0)and(y='a') then begin q:=1; znn:='OK'; goto m1; end;
If (q=1)and(y='_') then begin q:=9; znn:=zn[1]; goto m1; end;
{ obuchenie slovu 2 }
If (q=1)and(y='b') then begin q:=2; znn:='OK'; goto m1; end;
if (q=2)and(y='_') then begin q:=9; znn:=zn[2]; goto m1; end;
{ obuchenie slovu 3 }
if (q=2)and(y='c') then begin q:=3; znn:='OK'; goto m1; end;
if (q=3)and(y='_') then begin q:=9; znn:=zn[3]; goto m1; end;
{ obuchenie slovu 4 }
if (q=3)and(y='d') then begin q:=4; znn:='OK'; goto m1; end;
if (q=4)and(y='_') then begin q:=9; znn:=zn[4]; goto m1; end;
{ obuchenie slovu 5 }
if (q=0)and(y='c') then begin q:=5; znn:='OK'; goto m1; end;
if (q=5)and(y='d') then begin q:=6; znn:='OK'; goto m1; end;
if (q=6)and(y='b') then begin q:=7; znn:='OK'; goto m1; end;
if (q=7)and(y='c') then begin q:=8; znn:='OK'; goto m1; end;
if (q=8)and(y='_') then begin q:=9; znn:=zn[5]; goto m1; end;
znn:='neponimau'; writeln(' ',q,' zz: ',znn); goto m2;
m1: writeln(' ',q,' zz: ',znn);
end; m2: readkey;
END.

```

**Ситуация 1.** ДМУ изучает объекты (формирует навыки) в следующей последовательности: 1) объект “abcd\_”; 2) объект “a\_”; 3) объект “ab\_”; 4) объект “abc\_”. После обучения на каждый объект ДМУ реагирует соответствующим образом.

В данном случае учащийся сначала изучает всю теорию ABCD (или осваивает полную последовательность действий), а затем разбирает частные случаи: теорию А, сложную теорию АВ и еще более сложную теорию АВС. Используется программа ПР-1. Для изучения объекта “abcd” необходимо создать пять новых связей с помощью условных операторов:

```
If (q=0)and(y='a')then begin q:=1; znn:='OK'; goto m1; end;
If (q=1)and(y='b')then begin q:=2; znn:='OK'; goto m1; end;
If (q=2)and(y='c')then begin q:=3; znn:='OK'; goto m1; end;
If (q=3)and(y='d')then begin q:=4; znn:='OK'; goto m1; end;
If (q=4)and(y='_')then begin q:=9;znn:=zn[4]; goto m1; end;
```

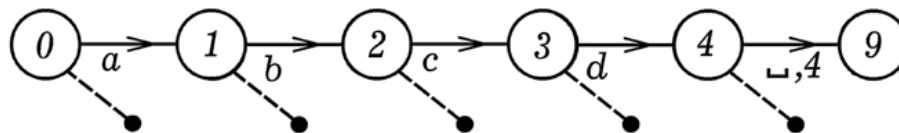


Рис. 2. Граф обучения, соответствующий ситуации 2.

Для того, чтобы теперь обучить автомат словам “a\_”, “ab\_”, “abc\_” достаточно к существующим добавить три новые связи с помощью операторов:

```
If (q=1)and(y='_') then begin q:=9; znn:=zn[1]; goto m1; end;
If (q=2)and(y='_') then begin q:=9; znn:=zn[2]; goto m1; end;
If (q=3)and(y='_') then begin q:=9; znn:=zn[3]; goto m1; end;
```

Если теперь после запуска программы вводить слова “abcd\_”, “a\_”, “ab\_”, “abc\_”, то созданная ДМУ будет реагировать правильно (рис.3). Все это моделирует изучение запоминание генетически связанных объектов.

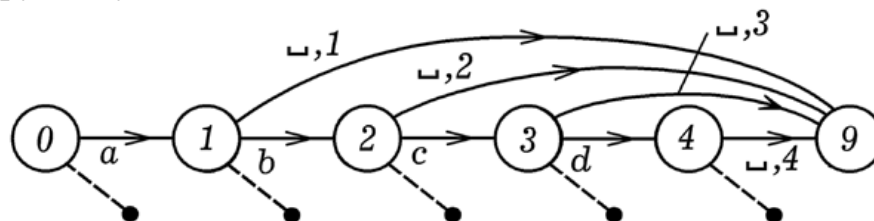


Рис. 3. Граф обучения, соответствующий ситуациям 1 и 2.

**Ситуация 2.** Пусть ДМУ изучает объекты (формирует навыки) в следующей последовательности: 1) объект “a\_”; 2) объект “ab\_”; 3) объект “abc\_”; 4) объект “abcd\_”; 5) объект “cdbc\_”. Про моделируем на компьютере

функционирование детерминированного автомата с фиксированным числом состояний, который можно будет научить различать эти объекты путем добавления условных операторов.

Диаграмма Мура такого автомата изображена на рис. 3. Сначала автомат находится в состоянии  $q = 0$ . Чтобы “научить” его правильно реагировать на слово “a\_”, необходимо организовать переходы  $(0, “a”) \rightarrow (1, “OK”)$  и  $(1, “_”) \rightarrow (9, “реаксіуа1”)$ , в результате чего на экране монитора должно появиться надпись “реаксіуа1”, которое можно считать значением слова “a\_”. Для этого используются следующие два оператора условного перехода:

```
If (q=0)and(y='a') then begin q:=1; znn:='OK'; goto m1; end;
If (q=1)and(y='_') then begin q:=9; znn:=zn[1]; goto m1; end;
```

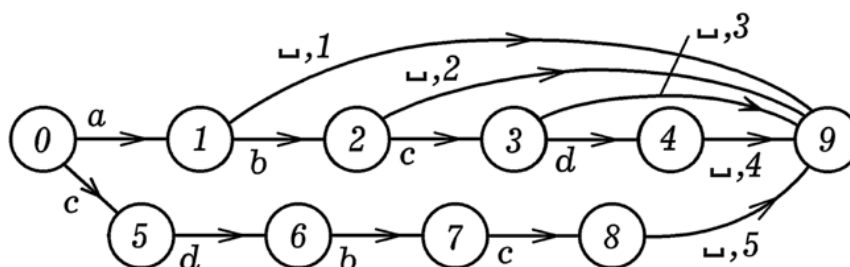


Рис. 4. Граф обучения новому слову “cdbcs\_”.

Чтобы автомат, “изучивший” объект “a\_”, “научить” правильно реагировать на объект “ab\_” также достаточно создать два перехода  $(1, “b”) \rightarrow (2, “OK”)$  и  $(2, “_”) \rightarrow (9, “реаксіуа2”)$ . В результате последнего перехода на экране должно появиться слово “реаксіуа2”. Для этого потребуется еще два оператора:

```
If (q=1)and(y='b') then begin q:=2; znn:='OK'; goto m1; end;
If (q=2)and(y='_') then begin q:=9; znn:=zn[2]; goto m1; end;
```

Аналогичным образом осуществляется “обучение” автомата словам “abc\_” и “abcd\_”: каждый раз достаточно добавлять по два оператора условного перехода, соответствующих созданию двух новых связей. Чтобы проверить распознавание компьютером того или иного объекта, необходимо присвоить переменной  $m$  номер предъявляемого слова. На экран будет выведена последовательность состояний, которую проходит автомат, и его реакция.

С объектом “cdbcs\_” ситуация принципиально иная: он не может быть получен из ранее изученных объектов путем добавления одного или двух элементов. Слово “cdbcs\_” не ассоциируется ни с одним из ранее изученных слов; чтобы “научить” автомат правильно на него реагировать, необходимо создать пять новых связей. В компьютерную программу, моделирующую такой автомат, следует добавить пять операторов:

```

If (q=0)and(y='c') then begin q:=5; znn:='OK'; goto m1; end;
If (q=5)and(y='d') then begin q:=6; znn:='OK'; goto m1; end;
If (q=6)and(y='b') then begin q:=7; znn:='OK'; goto m1; end;
If (q=7)and(y='c') then begin q:=8; znn:='OK'; goto m1; end;
If (q=8)and(y='_') then begin q:=9; znn:=zn[5]; goto m1; end;

```

Из принципа экономии следует, что в сложных системах с большей вероятностью реализуются состояния и процессы, требующие минимальных затрат энергии (ресурсов, сил, времени). Запоминание новых знаний (формирование умений и навыков), основанное на создании ассоциаций с уже имеющимися знаниями, требует создания меньшего числа связей, а значит меньших затрат энергии. Поэтому оно осуществляется легче и быстрее, чем при механическом заучивании.

## 2. Непрерывная модель ученика (НМУ)

Проанализируем проблему изучения генетически связанных ЭУМ с помощью непрерывной модели ученика. Пусть учитель в течение первого урока решает  $n_1$  учебных задач типа 1 на знание теории А, затем после перемены —  $n_2$  задач типа 2 на знание теории В. После второй перемены —  $n_3$  задач типа 3 на знание теории С. После третьей перемены —  $n_4$  задач типа 1–2, требующих знаний теории А и В, затем после перемены —  $n_5$  задач типа 1–2–3, требующих знаний теорий А, В и С, и т.д. в соответствии с программой обучения ПО – 1. А в другой раз учитель работает по программе ПО–2 и на всех 5 уроках сразу решает задачи типа 1–2–3, одновременно формируя знания тем А, В и С.

Допустим, при решении каждой задачи типа  $k$  вероятность правильного решения  $Z_k = p_k$  увеличивается на величину  $\alpha(1 - p_k)$ , где  $\alpha$  — коэффициент научения. Чем выше уровень знаний  $Z_k = p_k$ , тем меньше времени  $t_k$  затрачивается на решение следующей задачи типа  $k$ . Будем считать, что:  $t_k = t_0 - \ln(p_k) / \beta$ , где  $t_0$  — минимальное время, затрачиваемое учеником, имеющим максимальный уровень знаний  $Z_k = p_k = 1$ , на решение задачи типа  $k$ .

Урок заканчивается, ученик несколько дней не занимается изучением данного предмета, теряя при этом приобретенные знания вследствие забывания:  $p_i^{t+1} = p_i^t (1 - \gamma \Delta t)$ ,  $i = 1, 2, 3$ . На втором уроке учитель

расширяет теоретическую модель от А до АВ и решает  $n_2$  задач типа 1–2, в которых используются знания теорий А и В. При каждом решении задачи повышаются уровни знаний  $Z_1 = p_1$  и  $Z_2 = p_2$ , затрачивается время  $t_1 = t_0 - \ln(p_1)/\beta$  и  $t_2 = t_0 - \ln(p_2)/\beta$ . Если у учащегося сформированы знания теории С на уровне  $Z_3 = p_3$ , которая не используется при решении задач типа 1–2, то эти знания уменьшаются вследствие забывания по закону:  $p_3^{t+1} = p_3^t(1 - \gamma\Delta t)$ .

Эта ситуация моделируется с помощью программы ПР–2. В ней перечислены все типы решаемых задач `zadacha[1]='a'; zadacha[2]='b'; zadacha[3]='ab'; zadacha[4]='c'; zadacha[5]='bc'; zadacha[6]='abc'; zadacha[7]='_'`. Последняя `zadacha[7]` соответствует перерыву между уроками. Программа обучения ПО1={а, 6; р, 18; ab, 8; р, 18; abc, 6; р, 18; abc, 6; р, 18} задана в виде: `z:=1; chislo:=6; Raschet; z:=7; chislo:=18; Raschet; z:=3; chislo:=8; Raschet; z:=7; chislo:=18; Raschet;` и т.д. Это означает, что задачи типа 1 ('a') будут решены 6 раз, после чего наступит перерыв длительностью 18 условных единиц времени. Затем решаются задачи типа 1–2 ('ab') 8 раз, снова перерыв, после этого задачи типа 1–2–3 ("abc") 6 раз и т.д. Процедура `Raschet` содержит цикл по времени, в котором пересчитываются уровни знаний  $Z_i = p_i$ ,  $i = 1, 2, 3, \dots$  в процессе решения каждой задачи. При этом учитывается уменьшение времени решения задачи с ростом знаний учащегося, а также уменьшение неиспользуемых знаний вследствие забывания.

```

uses crt, graph;                                     { ПР–2 }
const alfa=0.4; beta=0.1; dt=0.03;
var i,j,j1,k,z,chislo,DV,MV : integer;
zadacha : array[0..10] of string; y: string;
g,t,t0,t1: real; p: array[1..5]of real;
Procedure Raschet;
begin k:=0;
Repeat
  For i:=1 to length(zadacha[z]) do begin
    y:=copy(zadacha[z],i,1);
    If y='a' then j:=1;   If y='b' then j:=2;
    If y='c' then j:=3;   If y='_' then j:=4; { peremena }
    p[j]:=p[j]+alfa*(1-p[j]);
    If p[j]<=1 then t:=t+t0-ln(p[j])/beta; t1:=0; p[4]:=1;
    If j=4 then g:=0.003 else g:=0.001;
    Repeat For j1:=1 to 3 do begin
      If (j1<>j)and(p[j1]>0) then p[j1]:=p[j1]-g*p[j1]*dt;
    end; t1:=t1+dt;
  until t1>t0-ln(p[j])/beta;

```

```

circle(10+round(1.2*t),450-round(p[1]*100),2);
circle(10+round(1.2*t),300-round(p[2]*100),2);
circle(10+round(1.2*t),150-round(p[3]*100),2);
end; inc(k);
until k>=chislo;
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(10,0,10,480); line(0,450,640,450); t0:=3;
zadacha[1]='a'; zadacha[2]='b';
zadacha[3]='ab'; zadacha[4]='c';
zadacha[5]='bc'; zadacha[6]='abc'; zadacha[7]='_';
{--- Programma obucheniya ---}
z:=1; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=3; chislo:=8; Raschet; z:=7; chislo:=18; Raschet;
z:=6; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=6; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
readkey; CloseGraph;
END.

```

Результаты имитационного моделирования представлены на рис. 5. Видно, что на первом уроке растет  $Z_1$  (знания А), затем после перерыва на втором уроке растет  $Z_1$  и  $Z_2$  (знания А и В), на третьем и четвертом уроках происходит увеличение  $Z_1$ ,  $Z_2$  и  $Z_3$  (знания А, В и С). Во время перерывов уровни знаний уменьшаются.

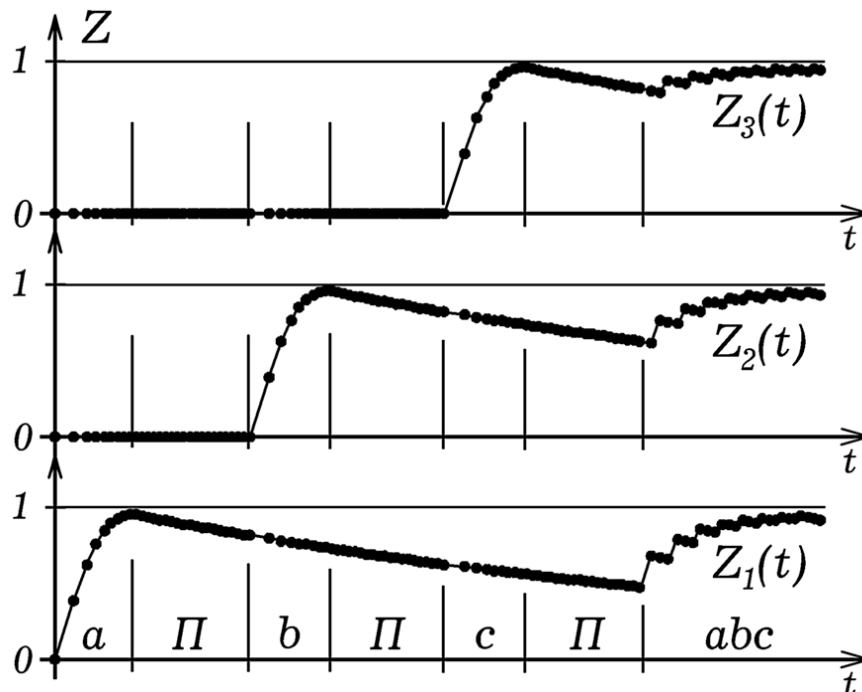


Рис. 5. Моделирование изучения генетически связанных теорий.

**Ситуация 1.** Учитель работает по программе:  $ПО2 = \{a, 6; p, 18; b, 8; p, 18; c, 6; p, 18; abc, 6; p, 18\}$ , то есть на первом уроке решает задачи типа А, на



втором — типа В, на третьем — типа С, а на четвертом — типа АВС. Для моделирования этой ситуации в программу ПР-2 необходимо вставить операторы:

```
z:=1; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=2; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=4; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
z:=6; chislo:=6; Raschet; z:=7; chislo:=18; Raschet;
```

Результаты вычислений представлены на рис. 6. Видно, что на первом уроке растет  $Z_1$ , а затем снижается; на втором уроке происходит увеличение  $Z_2$ , а затем снижается. На третьем уроке растет  $Z_3$ . Во время четвертого урока, на котором решаются задачи типа АВС, имеет место увеличение  $Z_1$ ,  $Z_2$  и  $Z_3$ .

**Ситуация 2.** Учитель работает по программе:  $ПО2 = \{ab, 6; p, 18; b, 8; p, 18; c, 6; p, 18; abc, 6; p, 18\}$ , то есть на первом уроке решает задачи типа АВ, на втором — типа В, на третьем — типа С, а на четвертом — типа АВС. Результаты моделирования приведены на рис. 6.

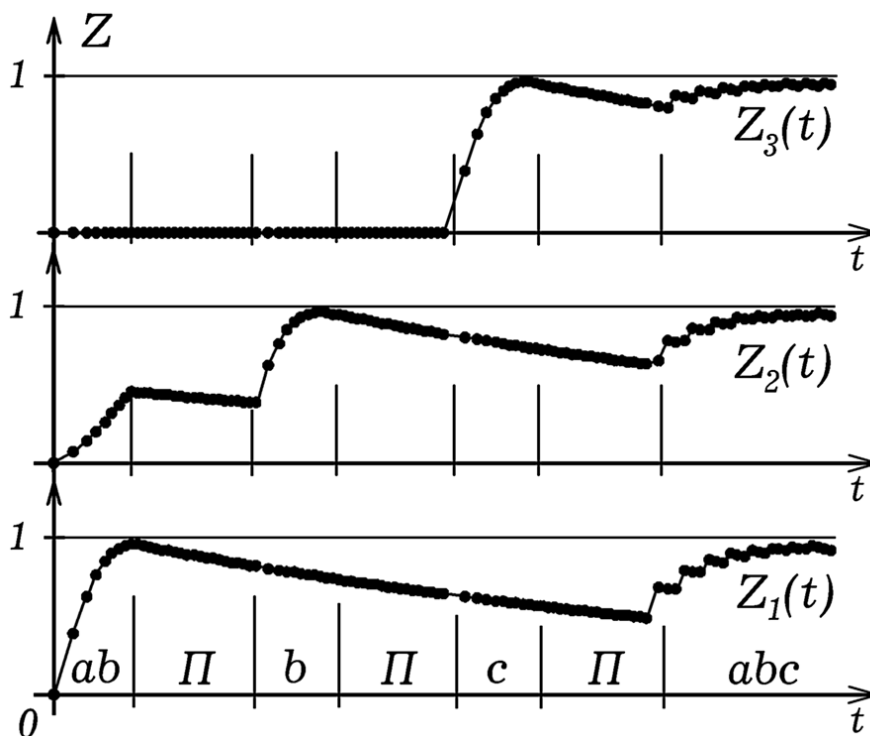


Рис. 6. Результаты моделирования изучения

### 3. Непрерывная многокомпонентная модель обучения

Рассмотренная выше модель слишком примитивна и не учитывает ряд важных факторов. Если бы процесс обучения был подобен наполнению резервуара жидкостью из различных сосудов и описывался бы линейным уравнением, то было бы все равно в какой последовательности осуществляется изучение теорий А, В и С. На самом же деле это не так.

Психологи установили следующее: 1. При обучении человек не может удержать в оперативной памяти слишком большое количество новой информации (обычно, 5–9 блоков). 2. При многократном повторении одних и тех же действий (в том числе интеллектуальных), решении подобных задач прочность знаний растет, и после обучения учащийся их практически не забывает. Поэтому не надо обучать всем буквам алфавита за один урок; необходимо алфавит разбить на порции по 3–4 буквы и на каждом уроке давать новую порцию информации. В то же время, если человек научился читать или считать, а затем выполнил достаточно большое количество упражнений, то знания становятся настолько прочными, что и после большого перерыва он сможет вспомнить изученный материал. 3. Обучение приводит к повышению интеллектуального уровня учащихся, то есть увеличению коэффициента научения. Чем больше учащийся знает, тем легче он усваивает новую информацию, устанавливая ассоциативные связи с уже имеющимися знаниями. 4. Если изучаемый материал слишком сложен и/или длительность занятия достаточно велика, то ученик начинает испытывать усталость, что приводит к снижению коэффициента научения.

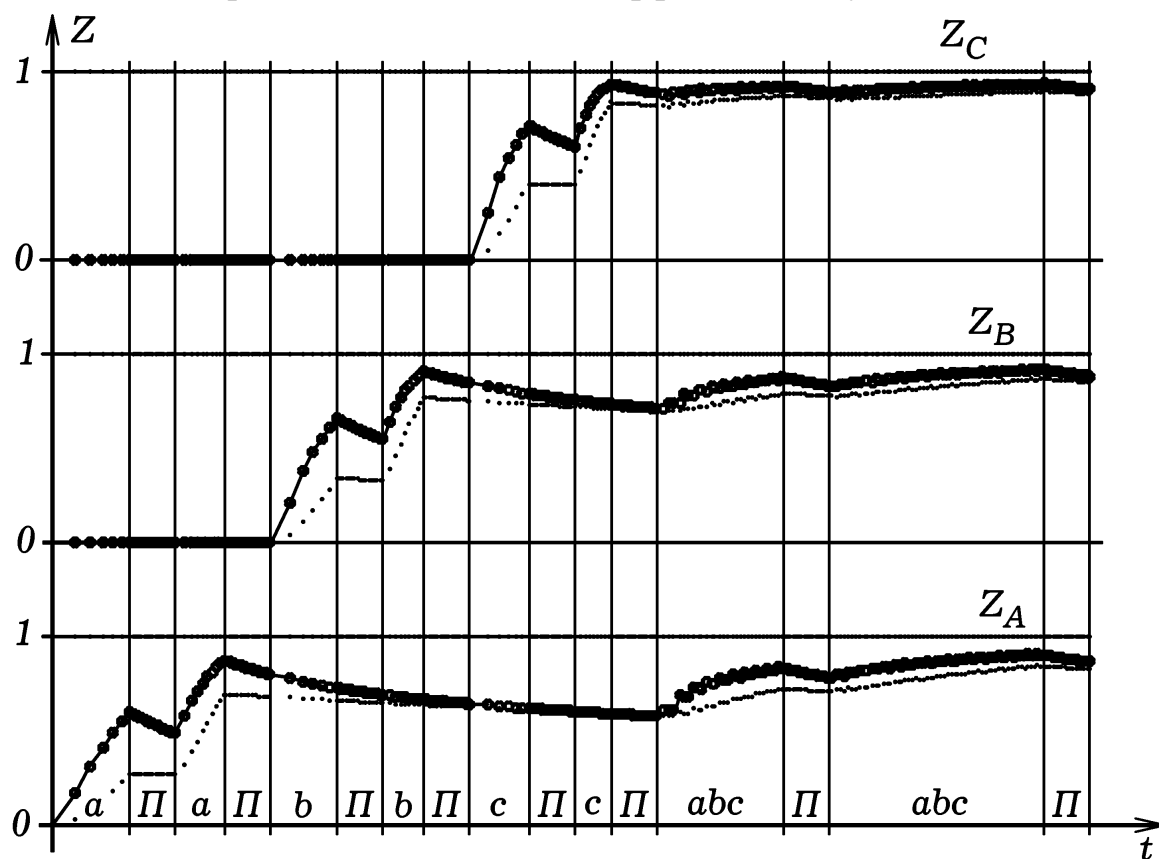


Рис. 7. Результаты использования многокомпонентной модели обучения.

Будем использовать многокомпонентную модель обучения [2], считая, что при многократном повторении часть информации хорошо усваивается и

на ее основе формируются навыки (или более прочные знания) с низким коэффициентом забывания. Математически это можно выразить так:

$$z_i^{t+1} = z_i^t + a_1(1 - p_i^t) - a_2 z_i^t - \gamma_1 z_i^t, \quad n_i^{t+1} = n_i^t + a_2 z_i^t - \gamma_2 n_i^t,$$

$$p_i^{t+1} = z_i^{t+1} + n_i^{t+1}.$$

Здесь  $p_i^t$  — вероятность правильного решения задачи по  $i$ -ой теме, которая складывается из знаний первой категории  $z_i^t$ , забываемых быстро, и знаний второй категории (или навыков)  $n_i^t$ , которые забываются существенно медленнее. За один акт обучения знания первой категории в количестве  $a_2 z_i^t$  становятся знаниями второй категории. Коэффициенты забывания  $\gamma_1$  и  $\gamma_2 < \gamma_1$  определяют быстроту уменьшения знаний первой и второй категории. Коэффициент научения  $a$  тем больше, чем больше средний уровень обучения  $p_{cp}$  (количество знаний), чем меньше длина задачи  $L_3$  и усталость  $K_{уст}$  учащегося. Будем считать, что:

$$a = \frac{A}{(2 - p_{cp})L_3(1 + K_{уст})}.$$

Используется компьютерная программа ПР-3. Результаты моделирования процесса обучения по учебной программе ПОЗ = {a, 6; p, 20; a, 9; p, 20; b, 6; p, 20; b, 9; p, 20; c, 6; p, 20; c, 9; p, 20; abc, 10; p, 20; abc, 10; p, 20} представлены на рис. 7.

```

uses crt, graph;                                     { ПР-3 }
const dt=0.001; g1=0.01; g2=0.0005; beta=0.1; Mt=0.6;
var i,j,r,s,k,chislo,DV,MV : integer;
zadacha : array[0..10] of string; y: string;
a1,a2,t,t0,t1,Ust,SP: real; pp,p,z,n: array[1..5]of real;
Procedure Raschet;
begin k:=0;
Repeat For i:=1 to length(zadacha[s]) do begin
  y:=copy(zadacha[s],i,1);
  If y='a' then j:=1; If y='b' then j:=2;
  If y='c' then j:=3; If y='_' then j:=4; { peremena }
  SP:=(p[1]+p[2]+p[3])/3;
  a1:=0.3/(2-SP)/length(zadacha[s])/(Ust+1);
  a2:=0.3/(2-SP)/length(zadacha[s])/(Ust+1);
  If j<4 then begin Ust:=Ust+(p[j]-pp[j]);
  z[j]:=z[j]+a1*(1-p[j])-a2*z[j];
  n[j]:=n[j]+a2*z[j]; end else Ust:=0;
  pp[j]:=p[j]; p[j]:=z[j]+n[j]; p[4]:=1;

```

```

If (p[j]<=1)and(p[j]>0)then t:=t+t0-ln(p[j])/beta; t1:=0;
Repeat For r:=1 to 3 do begin
  If (r<>j)and(z[r]>0) then z[r]:=z[r]-g1*z[r]*dt;
  If (r<>j)and(n[r]>0) then n[r]:=n[r]-g2*n[r]*dt;
  p[r]:=z[r]+n[r]; end; t1:=t1+dt;
until (t1>t0-ln(p[j])/beta)or(KeyPressed);
circle(10+round(Mt*t),450-round(p[1]*100),2);
circle(10+round(Mt*t),450-round(n[1]*100),1);
circle(10+round(Mt*t),300-round(p[2]*100),2);
circle(10+round(Mt*t),300-round(n[2]*100),1);
circle(10+round(Mt*t),150-round(p[3]*100),2);
circle(10+round(Mt*t),150-round(n[3]*100),1);
end; inc(k); until (k>=chislo)or(KeyPressed);
end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(10,0,10,480); line(0,450,640,450); t0:=2;
For j:=1 to 3 do p[j]:=0.1;
zadacha[1]='a'; zadacha[2]='b';
zadacha[3]='c'; zadacha[4]='ab';
zadacha[5]='bc'; zadacha[6]='abc'; zadacha[7]='_';
{--- Programma obucheniya ---}
s:=1; chislo:=6; Raschet; s:=7; chislo:=20; Raschet;
s:=1; chislo:=9; Raschet; s:=7; chislo:=20; Raschet;
s:=2; chislo:=6; Raschet; s:=7; chislo:=20; Raschet;
s:=2; chislo:=9; Raschet; s:=7; chislo:=20; Raschet;
s:=3; chislo:=6; Raschet; s:=7; chislo:=20; Raschet;
s:=3; chislo:=9; Raschet; s:=7; chislo:=20; Raschet;
s:=6; chislo:=10; Raschet; s:=7; chislo:=20; Raschet;
s:=6; chislo:=20; Raschet; s:=7; chislo:=20; Raschet;
readkey; CloseGraph;
END.

```

## Литература

1. Леонтьев Л.П., Гохман О.Г. Проблемы управления учебным процессом: Математические модели. — Рига, 1984. — 239 с.
2. Майер Р.В. Исследование многокомпонентной модели обучения на ЭВМ // Наука и современность — 2012: Сборник материалов XVI Международной научно–практической конференции: в 2-х частях. Часть 2. — Новосибирск: Издательство НГТУ, 2012. — С. 33–38.
3. Новиков Д.А. Закономерности итеративного научения. — М.: ИПУ, 1998. — 77 с.
4. Шеннон Р. Имитационное моделирование систем: искусство и наука. — М.: Мир, 1978. — 302 с.