Олимпиада для студентов и выпускников вузов -2013 г.

Задание по направлению «Прикладная математика» с решением
Профиль
«Системы управления и обработки информации в инженерии»

Время выполнения задания – 240 мин.
Решите задачи.

Задача 1.(20 баллов)

Найти решение системы дифференциальных уравнений:

$$\begin{cases} \dfrac{dx(t)}{dt} = -1.5x(t) + 0.5y(t) \\ \dfrac{dy(t)}{dt} = -1.4x(t) + 0.2y(t) \end{cases}$$

Определить устойчиво ли решение и, если устойчиво, определить значения переменных $x(t)$ и $y(t)$ при $t \to \infty$. Найти частное решение системы при $x(t_0) = 3,\ \ y(t_0) = -1$.

Решение.

Для установления устойчивости решения системы определим корни характеристического уравнения:

$$\begin{vmatrix} -1.5-\lambda & 0.5 \\ -1.4 & 0.2-\lambda \end{vmatrix} = 0.$$

Вычисляя определитель, получим:

$\lambda^2 + 1.3\lambda + 0.4 = 0$.

Корни этого уравнения $\lambda_1 = -0.8; \lambda_2 = -0.5.$

Оба корня отрицательные и действительные, следовательно система имеет установившееся решение при $t \to \infty.$ Эти установившиеся решения определяются системой алгебрагических уравнений:

$$\begin{cases} -1.5x + 0.5y = 0 \\ -1.4x + 0.2y = 0 \end{cases}$$

и равны $x(\infty) = 0; y(\infty) = 0.$

Найдем общее решение однородной системы

$$\begin{cases} \dfrac{dx(t)}{dt} = -1.5x(t) + 0.5y(t) \\ \dfrac{dy(t)}{dt} = -1.4x(t) + 0.2y(t) \end{cases}.$$

В соответствии с полученными корнями характеристического уравнения будем иметь два возможных решения:

$x_1 = \alpha_1^{(1)} \cdot e^{-0.8\cdot t}; y_1 = \alpha_2^{(1)} \cdot e^{-0.8\cdot t};$

$x_2 = \alpha_1^{(2)} \cdot e^{-0.5\cdot t}; y_2 = \alpha_2^{(2)} \cdot e^{-0.5\cdot t},$

где в $\alpha_i^{(j)}$, *i* – номер коэффициента, *j* – номер решения.

Подставим первое из решений в первое уравнение однородной системы

$$\frac{dx_1}{dt} = -0.8\alpha_1^{(1)} \cdot e^{-0.8\cdot t} = -1.5\alpha_1^{(1)} \cdot e^{-0.8\cdot t} + 0.5\alpha_2^{(1)} \cdot e^{-0.8\cdot t},$$

тогда

$$0.7\alpha_1^{(1)}e^{-0.8t} = 0.5\alpha_2^{(1)}e^{-0.8t};\Rightarrow \alpha_2^{(1)} = \frac{0.7}{0.5}\alpha_1^{(1)}.$$

Положив $\alpha_1^{(1)} = C_1$, получим $\alpha_2^{(1)} = 1.4\cdot C_1$. И, далее

$$x_1 = C_1\cdot e^{-0.8t}; y_1 = 1.4\cdot C_1\cdot e^{-0.8t}.$$

Подставим теперь второе решение во второе уравнение однородной системы:

$$\frac{dy_2}{dt} = -0.5\cdot\alpha_2^{(2)}\cdot e^{-0.5t} = -1.4\cdot\alpha_1^{(2)}\cdot e^{-0.5t} + 0.2\cdot\alpha_2^{(2)}\cdot e^{-0.5t}.$$

$$-0.7\cdot\alpha_2^{(2)}\cdot e^{-0.5t} = -1.4\cdot\alpha_1^{(2)}\cdot e^{-0.5t};\Rightarrow \alpha_2^{(2)} = \frac{1.4}{0.7}\cdot\alpha_1^{(2)}.$$

Положив $\alpha_1^{(2)} = C_2$, получим $\alpha_2^{(2)} = 2\cdot C_2$ и далее

$$x_2 = C_2\cdot e^{-0.5t}; y_2 = 2\cdot C_2\cdot e^{-0.5t}.$$

Обобщая найденные решения можно записать:

$$\begin{cases} x(t) = C_1\cdot e^{-0.8t} + C_2\cdot e^{-0.5t} \\ y(t) = 1.4\cdot C_1\cdot e^{-0.8t} + 2\cdot C_2\cdot e^{-0.5t} \end{cases}.$$

Определим частное решение при начальных условиях $x(0) = 3; y(0) = -1$.

$$\begin{cases} C_1 + C_2 = 3 \\ 1.4\cdot C_1 + 2\cdot C_2 = -1 \end{cases}$$

$$C_2 = -5.3333; C_1 = 8.33333;$$

Таким образом, имеем решение как функцию независимой переменной *t*:

$$\begin{cases} x(t) = 8.3333\cdot e^{-0.8t} - 5.3333\cdot e^{-0.5t} \\ y(t) = 1.4\cdot 8.3333\cdot e^{-0.8t} - 2\cdot 5.3333\cdot e^{-0.5t} \end{cases}$$

Задача 2.(10 баллов)

При параллельной передаче информации через два обрабатывающих узла компьютерной сети известно, что первый узел искажает ее обработку с вероятностью $1/3$, второй узел – с вероятностью $2/5$. Какова вероятность того, что первый узел безошибочно обработал информацию, если предположить, что второй узел обработал ее без искажений?

Решение.

Пусть событие А – первый узел безошибочно передал информацию; событие В – второй узел безошибочно передал информацию. Тогда по формуле условной вероятности

$$P(A/B)=\frac{P(AB)}{P(B)}=\frac{\frac{2}{3}\cdot\frac{3}{5}}{\frac{2}{3}\cdot\frac{3}{5}+\frac{1}{3}\cdot\frac{2}{5}}=\frac{6}{8}=\frac{3}{4}.$$

Задача 3.(20 балов)

Дана контекстно-свободная грамматика $G=(T, N, P, S)$, где T – множество терминальных символов, N – множество нетерминальных символов, P – множество продукций, S – аксиома. Построить для неё таблицу простого предшествования символов.

T: a, b

N: S

P: $S \to aSb$

$S \to ab$

Решение.

Используем для построения таблицы простого предшествования алгоритм Р. Флойда. Построим множества левовыводимых и правовыводимых символов для всех нетерминальных символов грамматики (в данном случае он всего один – S):

$L(S)=\{a\}$,

$R(S)=\{b\}$.

Просматривая правые части продукций грамматики, выпишем все пары соседних символов и запишем для них отношения предшествования.

aS	$a=S$	$a<L(S)$
Sb	$S=b$	$R(S)>b$
ab	$a=b$	

Объединим выписанные отношения в таблицу.

	S	a	b
S	.	.	=
a	=	<	=
b	.	.	>

Задача 4.(20 балов)

Найти все примитивные элементы поля Галуа $GF(16)=F_2[x]/x^4+x+1$.

Решение.

Так как в поле $GF(q)$ число примитивных элементов равно $\varphi(q-1)$, где $\varphi()$ – функция Эйлера, то в поле $GF(16)$ имеется $\varphi(15)=\varphi(3)\cdot\varphi(5)=8$ примитивных элементов. При этом достаточно найти один из них - α. Все остальные будут вычисляться в виде его степеней $\alpha^t \pmod{x^4+x+1}$ при всех натуральных t, которые взаимно просты с $q-1=15$.

Учитывая, что мультипликативный порядок примитивного элемента поля $GF(q)$ равен $q-1$, элемент θ поля $GF(16)$ является примитивным в том и только том случае, если $\theta^3 \neq 1$ и $\theta^5 \neq 1$.

Проверим элемент $\theta = x$. Имеем:

$$x^3 \neq 1(\operatorname{mod} x^4 + x + 1),$$
$$x^5 = x^4 \cdot x = (x+1)\cdot x = x^2 + x \neq 1(\operatorname{mod} x^4 + x + 1).$$

Следовательно, $\theta = x$ - один из примитивных элементов поля *GF(16)*. Остальные можем вычислить через его степени:

$$\theta_2 = x^2,$$
$$\theta_3 = x^4 = x+1,$$
$$\theta_4 = x^7 = x^4 \cdot x^3 = (x+1)\cdot x^3 = x^3 + x + 1,$$
$$\theta_5 = x^8 = x^4 \cdot x^4 = (x+1)\cdot(x+1) = x^2 + 1,$$
$$\theta_6 = x^{11} = x^8 \cdot x^3 = (x^2+1)\cdot x^3 = x^5 + x^3 = x \cdot x^4 + x^3 = x\cdot(x+1) + x^3 = x^3 + x^2 + x,$$
$$\theta_7 = x^{13} = x^{11}\cdot x^2 = (x^3 + x^2 + x)\cdot x^2 = x^5 + x^4 + x^3 = x\cdot(x+1) + x + 1 + x^3 = x^3 + x^2 + 1,$$
$$\theta_8 = x^{14} = x^{13}\cdot x = (x^3 + x^2 + 1)\cdot x = x^4 + x^3 + x = x + 1 + x^3 + x = x^3 + 1.$$

Задача 5.(30 баллов)

Чему будут равны значения операндов *a, b, c, d, n, k, m* после выполнения нижеприведенного фрагмента программы на языке программирования С в UNIX – подобной операционной системе и почему ?

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
int main()
{ int a=0, b=1, c=2, d=3, n=4, k=5, m=6, p[2];
char buf[80];
close(1);
pipe(p);
if( fork()==0)
{creat(“a.txt”, 0644);
m=open(“a.txt”,0);
a=write(p[1], “Привет участникам олимпиады\n”, 27);
close( p[0]);
b=dup(4);
exit(4);
}
else
{wait(&c);
d=read(1, buf,15);
n=read(p[0], buf,1 9);
k=dup2(m, b);
}
return 0;
}
```

Решение.

При создании процесса в UNIX-подобной ОС в таблице пользовательских дескрипторов файлов (ТПДФ) автоматически открываются три пользовательских дескриптора с индексами: 0 – для файла стандартного ввода, 1 – для файла стандартного вывода, 2 – для файла стандартного протокола ошибок. С выполнением системного вызова *close(1)* закрывается файл стандартного вывода, т.е. освобождается строка с индексом 1 ТПДФ. Далее выполняется системный вызов (СВ) *pipe(p).* По его завершению открываются два файла, и эта информация прописывается в свободные строки таблицы пользовательских дескрипторов файлов процесса, т.е. файл с пользовательским дескриптором *p[0]* прописывается в строку с индексом 1 и файл с пользовательским дескриптором *p[1]* – в строку с индексом 3. После этой операции выполняется системный вызов *fork().* Он создает копию текущего процесса – дочерний процесс, который начинает параллельно функционировать с родительским процессом. Таблица пользовательских дескрипторов открытых файлов процесса у дочернего процесса идентична ТПДФ родительского процесса.

Дочерний процесс создает файл *a.txt* в текущем каталоге с правами доступа «для владельца – читать и писать; для группы и прочих – только читать» и одновременно открывает его на запись (СВ *creat()*). В таблице пользовательских дескрипторов открытых файлов (ТПДФ) процесса занимается первая свободная строка и заполняется соответствующими атрибутами файла *a.txt.* Номер этой строки – 4.
Далее, с помощью СВ *open()* файл *a.txt* из текущего каталога открывается для чтения. Эта информация прописывается в первой свободной строке ТПДФ, т.е. в 5, поэтому значение операнда *m* дочернего процесса равно 5.
После этого осуществляется с помощью СВ *write()* запись в межпроцессный канал, т.е. в файл с пользовательским дескриптором *p[1],* 27 байт. СВ *write()* возвращает количество записанных байтов, в данном случае – 27, - значение операнда *a* равно 27.
Следующим действием *close()* закрывается файл с пользовательским дескриптором *p[0]*. В ТПДФ освобождается строка с индексом 1.
После выполнения СВ *dup()* строка ТПДФ с индексом 4 копируется в первую свободную строку, т.е. 1, поэтому значение операнда *b* равно 1.
С помощью функции *exit()* дочерний процесс прекращает свое существование. Значение запрашиваемых операндов в этот момент равны:

a=27
b=1
c=2
d=3
n=4
k=5
m=5

Пока выполнялся дочерний процесс, процесс-родитель ждет его завершения – СВ *wait().* По- завершению дочернего процесса процесс-родитель выходит из состояния блокировки и начинает выполняться. Значение операнда *c* соответствует аргументу функции *exit()* завершившегося дочернего процесса, т.е. *c=4.* Процесс-родитель считывает из файла с пользовательским дескриптором 1, равному *p[0],* т.е. файла межпроцессного канала, в переменную *buf* 15 байтов. Значение операнда *d* равно 15 - количеству считанных байтов. Далее считывается из межпроцессного канала (файла с пользовательским дескриптором *p[0]*) в переменную *buf* 19 байтов. Но так как в межпроцессном канале осталось всего 12 байтов, возвращаемое значение СВ *read()* будет равно 12, т.е. *n=12*. После этого выполняется СВ *dup2().* В строке с индексом *b=1* ТПДФ процесса должна записаться информация из строки *m=6* ТПДФ, но т.к. строка с индексом

6 – пустая, то СВ *dup2()* завершится аварийно, т.е. вернет значение -1, поэтому значение операнда *k* равно -1. Процесс-родитель завершает свое выполнение. Значения запрашиваемых операндов в этот момент равны:

a=0
b=1
c=4
d=15
n=12
k=-1
m=6,

поэтому ответ на эту задачу следующий:

Операнды	Значения операндов по завершении процесса-отца	Значения операндов по завершении процесса-сына
a	0	27
b	1	1
c	4	2
d	15	3
n	12	4
k	-1	5
m	6	5