

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Качество и метрология программного обеспечения»
ТЕМА: «Построение операционной графовой модели программы
(ОГМП) и расчет характеристик эффективности ее выполнения
методом эквивалентных преобразований»

Студент гр. 7304

Комаров А.О.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

1. Цель работы

Построение операционной графовой модели программы (ОГМП) и расчет характеристик эффективности ее выполнения методом эквивалентных преобразований.

2. Задание

2.1. Построение ОГМП.

Для рассматривавшегося в лабораторных работах 1-3 индивидуального задания разработать операционную модель управляющего графа программы на основе схемы алгоритма. При выполнении работы рекомендуется для упрощения обработки графа исключить диалог при выполнении операций ввода-вывода данных, а также привести программу к структурированному виду.

Выбрать вариант графа с нагруженными дугами, каждая из которых должна представлять фрагмент программы, соответствующий линейному участку или ветвлению. При расчете вероятностей ветвлений, зависящих от распределения данных, принять равномерное распределение обрабатываемых данных в ограниченном диапазоне (например, $[0,100]$ - для положительных чисел или $[-100,100]$ - для произвольных чисел). В случае ветвлений, вызванных проверкой выхода из цикла, вероятности рассчитываются исходя априорных сведений о числе повторений цикла. Сложные случаи оценки вероятностей ветвлений согласовать с преподавателем.

В качестве параметров, характеризующих потребление ресурсов, использовать времена выполнения команд соответствующих участков программы. С помощью монитора Sampler выполнить оценку времен выполнения каждого линейного участка в графе программы.

2.2. Расчет характеристик эффективности выполнения программы методом эквивалентных преобразований.

Полученную в части 2.1 данной работы ОГМП, представить в виде графа с нагруженными дугами, у которого в качестве параметров, характеризующих потребление ресурсов на дуге ij , использовать тройку $\{P_{ij}, M_{ij}, D_{ij}\}$, где:

P_{ij} - вероятность выполнения процесса для дуги ij ,

M_{ij} - мат.ожидание потребления ресурса процессом для дуги ij ,

D_{ij} - дисперсия потребления ресурса процессом для дуги ij .

В качестве потребляемого ресурса в данной работе рассматривается время процессора, а оценками мат. ожиданий времен для дуг исходного графа следует принять времена выполнения операторов (команд), соответствующих этим дугам участков программы. Дисперсиям исходных дуг следует присвоить нулевые значения.

Получить описание полученной ОГМП на входном языке пакета CSA III в виде поглощающей марковской цепи (ПМЦ) – (англ.) AMC (absorbing Markov chain) и/или эргодической марковской цепи (ЭМЦ) - EMC (ergodic Markov chain).

С помощью предоставляемого пакетом CSA III меню действий выполнить расчет среднего времени и дисперсии времени выполнения как для всей программы, так и для ее фрагментов, согласованных с преподавателем. Сравнить полученные результаты с результатами измерений, полученными в работе 3.

Построение операционной графовой модели программы

3. Текст программы (исходный)

```
1 #include <math.h>
2 #include <stdlib.h>
3 #include <time.h>
4 #include <stdio.h>
5
6 void sort1(float* x, int n){
7     float hold;
8     for (int i = 0; i < n - 1; i++) {
9         for (int j = i + 1; j < n; j++) {
10             if (x[i] > x[j]) {
11                 hold = x[i];
12                 x[i] = x[j];
13                 x[j] = hold;
14             }
15         }
16     }
17 }
18 void swap(float *a, float *b) {
```

```

19     float hold = (*a);
20     *a = (*b);
21     *b = hold;
22 }
23 void sort2(float *x, int n){
24     int no_change = 0;
25     while(!no_change) {
26         no_change = 1;
27         for (int j=0; j < n-1; j++) {
28             if (x[j] > x[j+1]) {
29                 swap(&x[j], &x[j+1]);
30                 no_change = 0;
31             }
32         }
33     }
34 }
35 int main(){
36     float x[1000];
37     float y[1000];
38
39     srand(time(NULL));
40     for (int i=0; i <1000; i++) {
41         x[i] = 1 + rand() % 999;
42         y[i] = x[i];
43     }
44     sort1(x,1000);
45     sort2(y,1000);
46     return 0;
47 }

```

Граф управления программы

Граф управления строился для каждого из функциональных участков отдельно, т.к. подсчёт затрат времени в предыдущей л/р выполнялся на них отдельно. Граф представлен на рис.1.

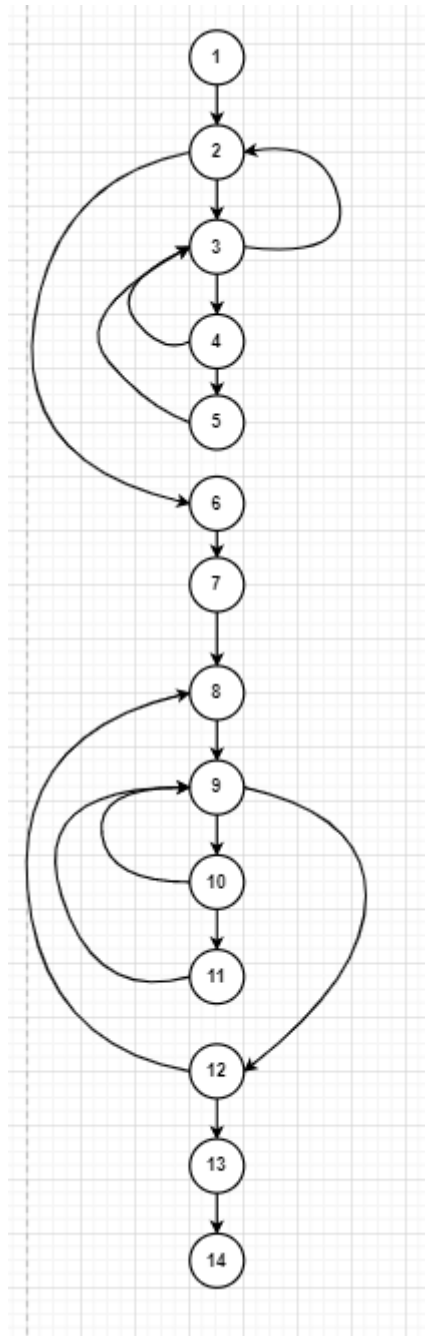


Рисунок 1 - Граф управления

4. Профилирование

Текст программы (подготовленный для профилирования)

```
1 #include <math.h>
2 #include <stdlib.h>
3 #include <time.h>
4 #include <stdio.h>
5 #include "Sampler.h"
6
7 void sort1(float* x, int n){
8     float hold;
9     SAMPLE;
```

```

10  for (int i = 0; i < n - 1; i++) {
11      SAMPLE;
12      for (int j = i + 1; j < n; j++) {
13          SAMPLE;
14          if (x[i] > x[j]) {
15              SAMPLE;
16              hold = x[i];
17              x[i] = x[j];
18              x[j] = hold;
19              SAMPLE;
20          }
21          SAMPLE;
22      }
23      SAMPLE;
24  }
25  SAMPLE;
26 }
27 void swap(float *a, float *b) {
28     float hold = (*a);
29     *a = (*b);
30     *b = hold;
31 }
32 void sort2(float *x, int n){
33     int no_change = 0;
34     SAMPLE;
35     while(!no_change) {
36         SAMPLE;
37         no_change = 1;
38         SAMPLE;
39         for (int j=0; j < n-1; j++) {
40             SAMPLE;
41             if (x[j] > x[j+1]) {
42                 SAMPLE;
43                 swap(&x[j], &x[j+1]);
44                 no_change = 0;
45                 SAMPLE;
46             }
47             SAMPLE;
48         }
49         SAMPLE;
50     }
51     SAMPLE;
52 }
53 int main(){
54     float x[1000];
55     float y[1000];
56     srand(time(NULL));
57     for (int i=0; i <1000; i++) {
58         x[i] = 1 + rand() % 999;
59         y[i] = x[i];
60     }
61     SAMPLE;
62     sort1(x,1000);
63     SAMPLE;
64     sort2(y,1000);
65     SAMPLE;
66     return 0;
67 }

```

Результаты профилирования

NN	Имя обработанного файла
1.	LR4.CPP

Таблица с результатами измерений (используется 17 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 9	1 : 11	2.51	1	2.51
1 : 11	1 : 13	2580.50	999	2.58
1 : 13	1 : 15	6378208.65	56572	112.74
1 : 13	1 : 21	12868364.37	49712	258.86
1 : 15	1 : 19	7203976.77	56572	127.34
1 : 19	1 : 21	5671376.49	56572	100.25
1 : 21	1 : 13	20678949.53	39749	520.24
1 : 21	1 : 23	1244.57	999	1.25
1 : 23	1 : 11	1912.54	998	1.92
1 : 23	1 : 25	1.68	1	1.68
1 : 25	1 : 63	1.68	1	1.68
1 : 34	1 : 36	0.84	1	0.84
1 : 36	1 : 38	243.89	937	0.26
1 : 38	1 : 40	1816.16	937	1.94
1 : 40	1 : 42	9745300.79	62044	157.07
1 : 40	1 : 47	31233734.22	22051	1416.43
1 : 42	1 : 45	11846909.10	62044	190.94
1 : 45	1 : 47	8220649.02	62044	132.50
1 : 47	1 : 40	40600346.97	17622	2303.96
1 : 47	1 : 49	1487.62	937	1.59
1 : 49	1 : 36	882.52	936	0.94
1 : 49	1 : 51	0.84	1	0.84
1 : 51	1 : 65	2.51	1	2.51
1 : 61	1 : 9	3.35	1	3.35
1 : 63	1 : 34	4.19	1	4.19

5. Расчет вероятностей и затрат ресурсов для дуг управляющего графа

	Номера строк	Количество проходов
L1 = 3.35 мкс	61:9	1
L2 = 4.43 мкс	9:11, 23:11	1:998
L3 = 261.44 мкс	11:13, 21:13	999:39749
L4 = 371.6 мкс	13:15, 13:21	56572:49712
L5 = 127.34 мкс	15:19	1
L6 = 104.86 мкс	19:21, 21:23, 23:25, 25:63	56572:39749:1:1
L7 = 4.19 мкс	63:34	1
L8 = 0.84 мкс	34:36	1
L9 = 2305.9 мкс	38:40, 47:40	937:17622
L10 = 1573.5 мкс	40:42, 40:47	62044:22051
L11 = 190.94 мкс	42:45	1
L12 = 1.78	49:36, 49:51	936:1
L13 = 2.51	51:65	1

6. Операционная графовая модель программы

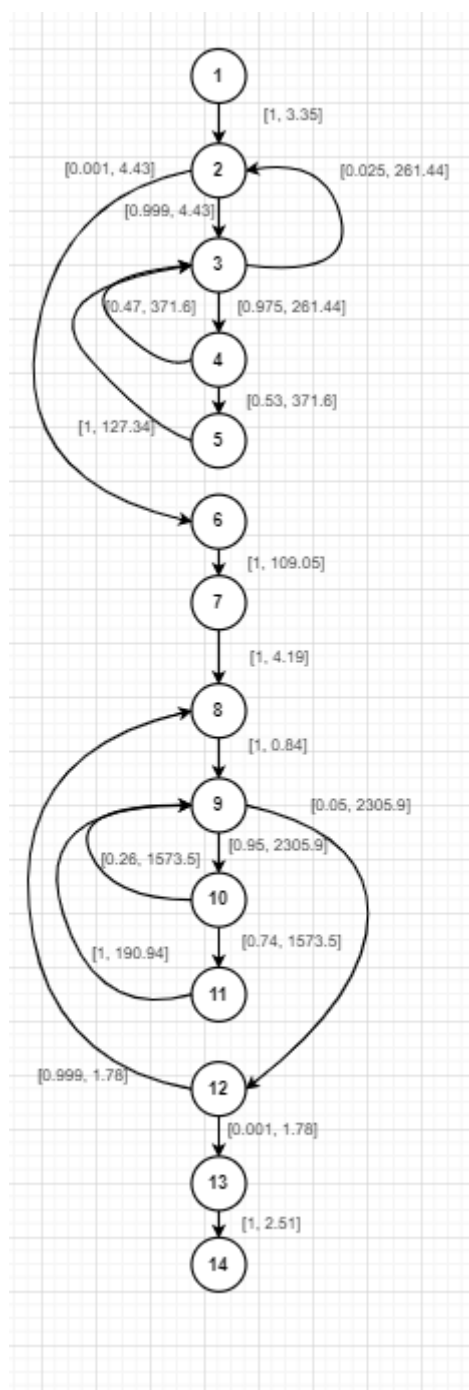
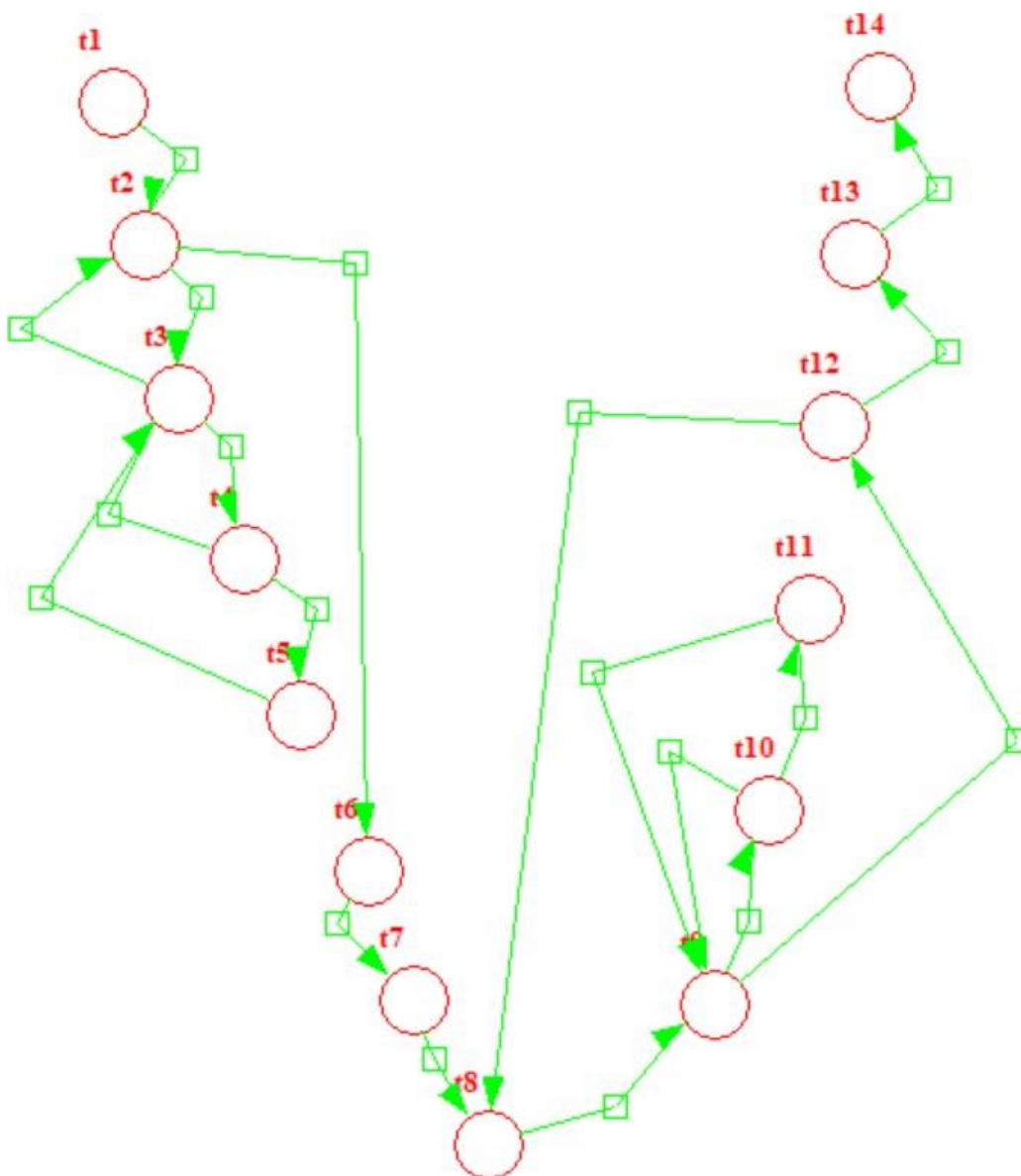


Рисунок 2 – Графовая модель

Расчет характеристик эффективности выполнения программы с помощью пакета CSA III методом эквивалентных преобразований

ГНД



7. Описание модели

lr4.xml

```
<model type = "Objects::AMC::Model" name = "lr4">
  <node type = "Objects::AMC::Top" name = "t1"></node>
  <node type = "Objects::AMC::Top" name = "t2"></node>
  <node type = "Objects::AMC::Top" name = "t3"></node>
  <node type = "Objects::AMC::Top" name = "t4"></node>
```

```

<node type = "Objects::AMC::Top" name = "t5"></node>
<node type = "Objects::AMC::Top" name = "t6"></node>
<node type = "Objects::AMC::Top" name = "t7"></node>
<node type = "Objects::AMC::Top" name = "t8"></node>
<node type = "Objects::AMC::Top" name = "t9"></node>
<node type = "Objects::AMC::Top" name = "t10"></node>
<node type = "Objects::AMC::Top" name = "t11"></node>
<node type = "Objects::AMC::Top" name = "t12"></node>
<node type = "Objects::AMC::Top" name = "t13"></node>
<node type = "Objects::AMC::Top" name = "t14"></node>
<link type = "Objects::AMC::Link" name = "t1-2" probability = "1" intensity
= "3.35" deviation = "0.0" source = "t1" dest = "t2"></link>
<link type = "Objects::AMC::Link" name = "t2-3" probability = "0.999"
intensity = "4.43" deviation = "0.0" source = "t2" dest = "t3"></link>
<link type = "Objects::AMC::Link" name = "t2-6" probability = "0.001"
intensity = "4.43" deviation = "0.0" source = "t2" dest = "t6"></link>
<link type = "Objects::AMC::Link" name = "t3-4" probability = "0.975"
intensity = "261.44" deviation = "0.0" source = "t3" dest = "t4"></link>
<link type = "Objects::AMC::Link" name = "t3-2" probability = "0.025"
intensity = "261.44" deviation = "0.0" source = "t3" dest = "t2"></link>
<link type = "Objects::AMC::Link" name = "t4-5" probability = "0.53"
intensity = "371.6" deviation = "0.0" source = "t4" dest = "t5"></link>
<link type = "Objects::AMC::Link" name = "t4-3" probability = "0.47"
intensity = "371.6" deviation = "0.0" source = "t4" dest = "t3"></link>
<link type = "Objects::AMC::Link" name = "t5-3" probability = "1" intensity
= "127.34" deviation = "0.0" source = "t5" dest = "t3"></link>
<link type = "Objects::AMC::Link" name = "t6-7" probability = "1" intensity
= "109.05" deviation = "0.0" source = "t6" dest = "t7"></link>
<link type = "Objects::AMC::Link" name = "t7-8" probability = "1" intensity
= "4.19" deviation = "0.0" source = "t7" dest = "t8"></link>
<link type = "Objects::AMC::Link" name = "t8-9" probability = "1" intensity
= "0.84" deviation = "0.0" source = "t8" dest = "t9"></link>
<link type = "Objects::AMC::Link" name = "t9-10" probability = "0.95"
intensity = "2305.9" deviation = "0.0" source = "t9" dest = "t10"></link>
<link type = "Objects::AMC::Link" name = "t9-12" probability = "0.05"
intensity = "2305.9" deviation = "0.0" source = "t9" dest = "t12"></link>
<link type = "Objects::AMC::Link" name = "t10-11" probability = "0.74"
intensity = "1573.5" deviation = "0.0" source = "t10" dest = "t11"></link>
<link type = "Objects::AMC::Link" name = "t10-9" probability = "0.26"
intensity = "1573.5" deviation = "0.0" source = "t10" dest = "t9"></link>
<link type = "Objects::AMC::Link" name = "t11-9" probability = "1" intensity
= "190.94" deviation = "0.0" source = "t11" dest = "t9"></link>
<link type = "Objects::AMC::Link" name = "t12-13" probability = "0.001"
intensity = "1.78" deviation = "0.0" source = "t12" dest = "t13"></link>

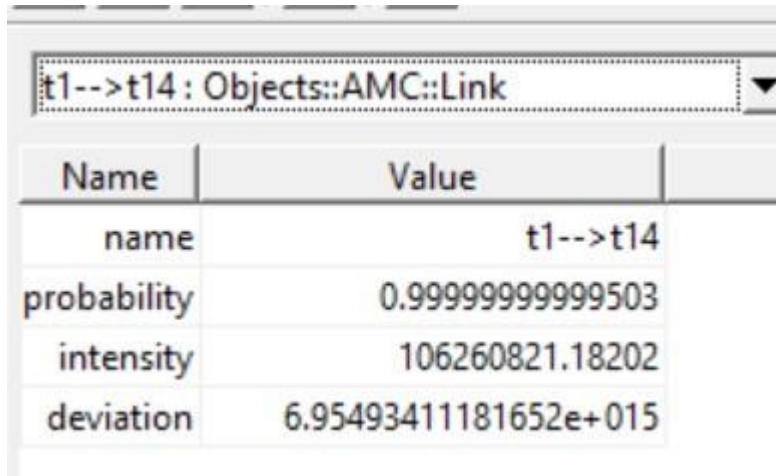
```

<link type = "Objects::AMC::Link" name = "t12-8" probability = "0.999"
intensity = "1.78" deviation = "0.0" source = "t12" dest = "t8"></link>

<link type = "Objects::AMC::Link" name = "t13-14" probability = "1"
intensity = "2.51" deviation = "0.0" source = "t13" dest = "t14"></link>

</model>

Результаты



The screenshot shows a software window with a dropdown menu at the top displaying "t1-->t14 : Objects::AMC::Link". Below the menu is a table with two columns: "Name" and "Value". The table contains four rows of data.

Name	Value
name	t1-->t14
probability	0.999999999999503
intensity	106260821.18202
deviation	6.95493411181652e+015

В прошлой лабораторной работе был получен результат 116423338. В результате чего разница между результатами составляет менее 10%.

Вывод

При выполнении лабораторной работы была построена операционная графовая модель заданной программы, нагрузочные параметры которой были оценены с помощью профилировщика Sampler и методом эквивалентных преобразований с помощью пакета CSA III были вычислены математическое ожидание и дисперсия времени выполнения как для всей программы, так и для заданного фрагмента. Результаты сравнения этих характеристик с полученными в работе 3 согласуются.