

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. 8304

Рыжиков А.В.

Преподаватель

Кирияничиков В. А.

Санкт-Петербург

2022

Цель работы.

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

Задание.

Для задания из лабораторных работ 1-3 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.

Полученную 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.

Ход работы.

1. Текст программы (исходный).

Исходный код программы представлен в приложении А.

2. Профилирование.

Код программы для профилирования, разделенной на функциональные участки, представлен в приложении В.

3. Граф управления программы.

Граф управления был построен на основе разбиения программы на функциональные участки, т.к. подсчёт затрат времени в лабораторной работе №3 осуществлялся на каждом из таких участков отдельно. На графе также отмечены метки (номера строк) и количество проходов для удобного вычисления вероятностей. Граф представлен на рисунке 1.

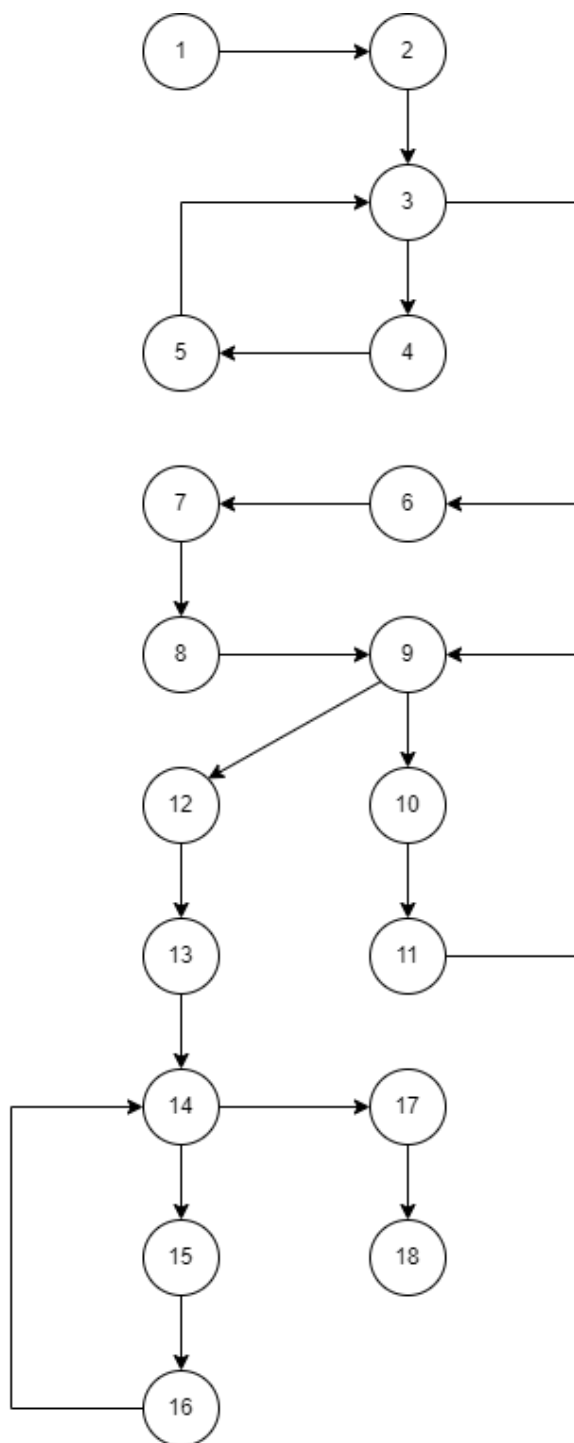


Рисунок 1 - Граф управления

4. Результаты профилирования.

Результаты профилирования из лабораторной работы №3 представлены в таблице 1.

Таблица 1 – Результаты профилирования

| исх | прием | общее время | кол-во проходов | среднее время |
|------------|--------------|--------------------|------------------------|----------------------|
| 41 | 45 | 2818.315 | 1 | 2818.315 |
| 45 | 46 | 34.191 | 1 | 34.191 |
| 46 | 47 | 1553.607 | 1000 | 1.554 |
| 46 | 52 | 28.383 | 1 | 28.383 |
| 47 | 50 | 20759.611 | 1000 | 20.760 |
| 50 | 46 | 1614.491 | 1000 | 1.614 |
| 52 | 10 | 17.649 | 1 | 17.649 |
| 10 | 13 | 11.586 | 1 | 11.586 |
| 13 | 14 | 3.339 | 1 | 3.339 |
| 14 | 15 | 1826.766 | 1000 | 1.827 |
| 14 | 25 | 20.424 | 1 | 20.424 |
| 15 | 23 | 7983.902 | 1000 | 7.984 |
| 23 | 14 | 1645.977 | 1000 | 1.646 |
| 25 | 31 | 14.578 | 1 | 14.578 |
| 31 | 32 | 11.841 | 1 | 11.841 |
| 32 | 33 | 1749.492 | 1000 | 1.749 |
| 32 | 54 | 64.172 | 1 | 64.172 |
| 33 | 35 | 4795.168 | 1000 | 4.795 |
| 35 | 32 | 1618.420 | 1000 | 1.618 |
| 54 | 58 | 631.871 | 1 | 631.871 |

5. Расчет вероятностей и затрат ресурсов для дуг управляющего графа.

Результаты расчета представлены в таблице 1.

Таблица 2 – Расчет вероятностей и затрат ресурсов для дуг.

| | Номера строк | Количество проходов | Вероятность |
|------------------------|--------------|---------------------|-------------|
| $L_1 = 2818.315$ мкс | 41:45 | 1 | 1 |
| $L_2 = 34.191$ мкс | 45:46 | 1 | 1 |
| $L_3 = 1.554$ мкс | 46:47 | 1000 | 0.999 |
| $L_4 = 20.76$ мкс | 47:50 | 1000 | 1 |
| $L_5 = 1.614$ мкс | 50:46 | 1000 | 1 |
| $L_6 = 28.383$ мкс | 46:52 | 1 | 0.001 |
| $L_7 = 17.649$ мкс | 52:10 | 1 | 1 |
| $L_8 = 11.586$ мкс | 10:13 | 1 | 1 |
| $L_9 = 3.339$ мкс | 13:14 | 1 | 1 |
| $L_{10} = 1.827$ мкс | 14:15 | 1000 | 0.999 |
| $L_{11} = 7.984$ мкс | 15:23 | 1000 | 1 |
| $L_{12} = 1.646$ мкс | 23:14 | 1000 | 1 |
| $L_{13} = 20.424$ мкс | 14:25 | 1 | 0.001 |
| $L_{14} = 14.576$ мкс | 25:31 | 1 | 1 |
| $L_{15} = 11.841$ мкс | 31:32 | 1 | 1 |
| $L_{16} = 1.749$ мкс | 32:33 | 1000 | 0.999 |
| $L_{17} = 4.795$ мкс | 33:35 | 1000 | 1 |
| $L_{18} = 1.618$ мкс | 35:32 | 1000 | 1 |
| $L_{19} = 64.172$ мкс | 32:54 | 1 | 0.001 |
| $L_{20} = 631.871$ мкс | 54:58 | 1 | 1 |

Операционная графовая модель программы.

Операционная графовая модель программы представлена на рисунке 3.

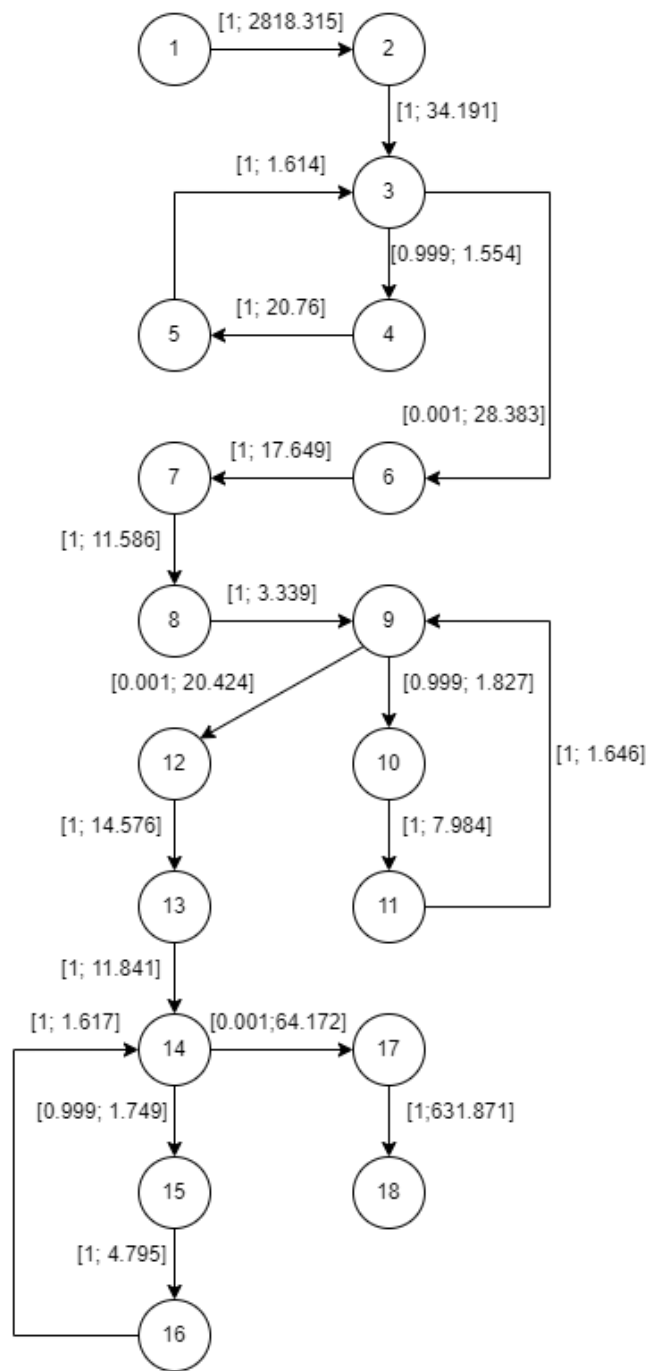


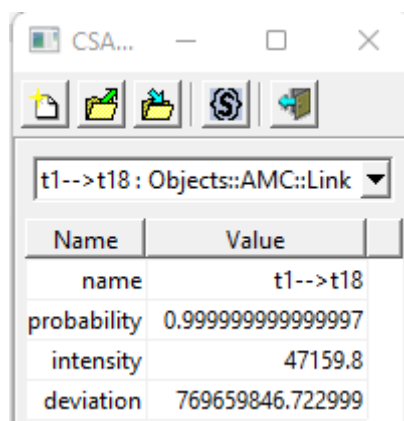
Рисунок 2 - Операционная модель

6. Описание модели model.xml.

Описание модели представлено в приложении С.

7. Результаты.

Результаты работы программы представлены на рисунке 4.



The screenshot shows a window titled 'CSA...' with a toolbar and a table of results. The table has two columns: 'Name' and 'Value'. The data is as follows:

| Name | Value |
|-------------|-------------------|
| name | t1-->t18 |
| probability | 0.999999999999997 |
| intensity | 47159.8 |
| deviation | 769659846.722999 |

Рисунок 4 - Результат работы программы

Вероятность получилась равной 1. Согласно расчётам программы, среднее время выполнения составляет 47159.8 мкс. В пункте 4 данного отчёта приведен результат профилирования программы с использованием SAMPLER_v2, где суммарное время выполнения составило 47203,78 мкс. В итоге, разница между результатами составляет менее 1 %.

Выводы.

В ходе выполнения лабораторной работы была построена операционная графовая модель заданной программы, нагрузочные параметры которой были оценены с помощью профилировщика SAMPLER_v2.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <stdio.h>
#include <stdlib.h>
#include "sampler.h"

#ifdef n
#define n 1000
#endif

void linfit2(float *x, float *y, float *y_calc) {
    float sum_x = 0, sum_y = 0, sum_xy = 0, sum_x2 = 0, sum_y2 = 0;
    float xi, yi, sxy, sxx, syy;
    for (int i = 0; i < n; ++i) {
        xi = x[i];
        yi = y[i];
        sum_x += xi;
        sum_y += yi;
        sum_xy += xi * yi;
        sum_x2 += xi * xi;
        sum_y2 += yi * yi;
    }
    sxx = sum_x2 - sum_x * sum_x / n;
    sxy = sum_xy - sum_x * sum_y / n;
    syy = sum_y2 - sum_y * sum_y / n;
    float b = sxy / sxx;
    float a = ((sum_x2 * sum_y - sum_x * sum_xy) / n) / sxx;
    for (int i = 0; i < n; ++i) {
        y_calc[i] = a + b * x[i];
    }
}

int main(int argc, char **argv) {
    sampler_init(&argc, argv);
    float *x = (float*) malloc(n * sizeof(float));
    float *y = (float*) malloc(n * sizeof(float));
    float *y_calc = (float*) malloc(n * sizeof(float));
    for (int i = 0; i < n; i++) {
        x[i] = (float) (rand() % 100) / 100;
        y[i] = (float) (rand() % 100) / 100;
    }
    linfit2(x, y, y_calc);
    free(x);
    free(y);
    free(y_calc);
    return 0;
}
```

ПРИЛОЖЕНИЕ В.

КОД ПРОГРАММЫ С РАЗДЕЛЕНИЕМ НА ФУ

```
#include <stdio.h>
#include <stdlib.h>
#include "sampler.h"

#ifdef n
#define n 1000
#endif

void linfit2(float *x, float *y, float *y_calc) {
    SAMPLE;
    float sum_x = 0, sum_y = 0, sum_xy = 0, sum_x2 = 0, sum_y2 = 0;
    float xi, yi, sxy, sxx, syy;
    SAMPLE;
    for (int i = 0; SAMPLE, i < n; ++i) {
        SAMPLE;
        xi = x[i];
        yi = y[i];
        sum_x += xi;
        sum_y += yi;
        sum_xy += xi * yi;
        sum_x2 += xi * xi;
        sum_y2 += yi * yi;
        SAMPLE;
    }
    SAMPLE;
    sxx = sum_x2 - sum_x * sum_x / n;
    sxy = sum_xy - sum_x * sum_y / n;
    syy = sum_y2 - sum_y * sum_y / n;
    float b = sxy / sxx;
    float a = ((sum_x2 * sum_y - sum_x * sum_xy) / n) / sxx;
    SAMPLE;
    for (int i = 0; SAMPLE, i < n; ++i) {
        SAMPLE;
        y_calc[i] = a + b * x[i];
        SAMPLE;
    }
}

int main(int argc, char **argv) {
    sampler_init(&argc, argv);
    SAMPLE;
    float *x = (float*) malloc(n * sizeof(float));
    float *y = (float*) malloc(n * sizeof(float));
    float *y_calc = (float*) malloc(n * sizeof(float));
    SAMPLE;
    for (int i = 0; SAMPLE, i < n; i++) {
        SAMPLE;
        x[i] = (float) (rand() % 100) / 100;
        y[i] = (float) (rand() % 100) / 100;
        SAMPLE;
    }
    SAMPLE;
    linfit2(x, y, y_calc);
    SAMPLE;
```

```
    free(x);  
    free(y);  
    free(y_calc);  
    SAMPLE;  
    return 0;  
}
```

ПРИЛОЖЕНИЕ С.

ОПИСАНИЕ МОДЕЛИ .XML

```
<model type = "Objects::AMC::Model" name = "model">
  <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>
  <node type = "Objects::AMC::Top" name = "t15"></node>
  <node type = "Objects::AMC::Top" name = "t16"></node>
  <node type = "Objects::AMC::Top" name = "t18"></node>
  <node type = "Objects::AMC::Top" name = "t17"></node>
  <link type = "Objects::AMC::Link" name = "t1-->t2" probability = "1.0"
intensity = "2818.315" deviation = "0.0" source = "t1" dest = "t2"></link>
  <link type = "Objects::AMC::Link" name = "t2-->t3" probability = "1.0"
intensity = "34.191" deviation = "0.0" source = "t2" dest = "t3"></link>
  <link type = "Objects::AMC::Link" name = "t3-->t4" probability = "0.999"
intensity = "1.554" deviation = "0.0" source = "t3" dest = "t4"></link>
  <link type = "Objects::AMC::Link" name = "t4-->t5" probability = "1.0"
intensity = "20.76" deviation = "0.0" source = "t4" dest = "t5"></link>
  <link type = "Objects::AMC::Link" name = "t3-->t6" probability = "0.001"
intensity = "28.383" deviation = "0.0" source = "t3" dest = "t6"></link>
  <link type = "Objects::AMC::Link" name = "t5-->t3" probability = "1.0"
intensity = "1.614" deviation = "0.0" source = "t5" dest = "t3"></link>
  <link type = "Objects::AMC::Link" name = "t6-->t7" probability = "1.0"
intensity = "17.649" deviation = "0.0" source = "t6" dest = "t7"></link>
  <link type = "Objects::AMC::Link" name = "t7-->t8" probability = "1.0"
intensity = "11.586" deviation = "0.0" source = "t7" dest = "t8"></link>
  <link type = "Objects::AMC::Link" name = "t8-->t9" probability = "1.0"
intensity = "3.339" deviation = "0.0" source = "t8" dest = "t9"></link>
  <link type = "Objects::AMC::Link" name = "t9-->t10" probability = "0.999"
intensity = "1.827" deviation = "0.0" source = "t9" dest = "t10"></link>
  <link type = "Objects::AMC::Link" name = "t10-->t11" probability = "1.0"
intensity = "7.984" deviation = "0.0" source = "t10" dest = "t11"></link>
  <link type = "Objects::AMC::Link" name = "t11-->t9" probability = "1.0"
intensity = "1.646" deviation = "0.0" source = "t11" dest = "t9"></link>
  <link type = "Objects::AMC::Link" name = "t9-->t12" probability = "0.001"
intensity = "20.424" deviation = "0.0" source = "t9" dest = "t12"></link>
  <link type = "Objects::AMC::Link" name = "t12-->t13" probability = "1.0"
intensity = "14.576" deviation = "0.0" source = "t12" dest = "t13"></link>
  <link type = "Objects::AMC::Link" name = "t13-->t14" probability = "1.0"
intensity = "11.841" deviation = "0.0" source = "t13" dest = "t14"></link>
  <link type = "Objects::AMC::Link" name = "t14-->t15" probability =
"0.999" intensity = "1.749" deviation = "0.0" source = "t14" dest =
"t15"></link>
  <link type = "Objects::AMC::Link" name = "t15-->t16" probability = "1.0"
intensity = "4.795" deviation = "0.0" source = "t15" dest = "t16"></link>
```

```
<link type = "Objects::AMC::Link" name = "t16-->t14" probability = "1.0"
intensity = "1.618" deviation = "0.0" source = "t16" dest = "t14"></link>
<link type = "Objects::AMC::Link" name = "t14-->t17" probability =
"0.001" intensity = "64.172" deviation = "0.0" source = "t14" dest =
"t17"></link>
<link type = "Objects::AMC::Link" name = "t17-->t18" probability = "1.0"
intensity = "631.871" deviation = "0.0" source = "t17" dest = "t18"></link>
</model>
```