

**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»**

Студентка гр. 8304

Мельникова О.А.

Преподаватель

Кириянчиков 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 Б.

#### **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.

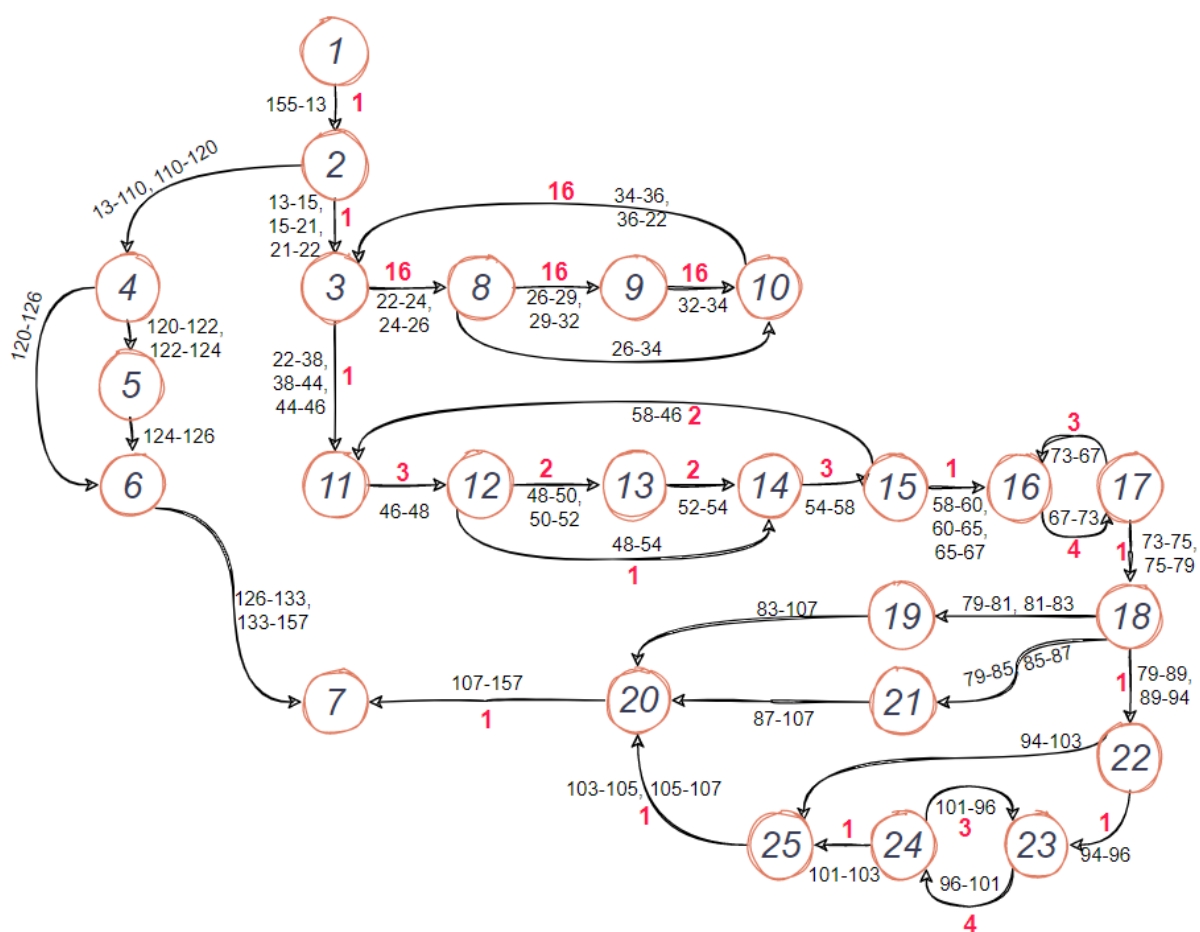


Рисунок 1 - Граф управления

#### 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
155	13	47.000	1	47.000
13	15	53.500	1	53.500
15	21	17.500	1	17.500
21	22	22.500	1	22.500
22	24	157.000	16	9.812
22	38	65.500	1	65.500
24	26	90.000	16	5.625
26	29	147.000	16	9.188
29	32	213.000	16	13.312
32	34	85.000	16	5.312
34	36	209.500	16	13.094
36	22	199.500	16	12.469
38	44	34.500	1	34.500
44	46	26.500	1	26.500
46	48	42.000	3	14.000
48	54	57.000	1	57.000
48	50	58.000	2	29.000
54	58	95.500	3	31.833
58	46	81.000	2	40.500
58	60	87.500	1	87.500
50	52	62.500	2	31.250
52	54	52.500	2	26.250
60	65	37.500	1	37.500
65	67	28.000	1	28.000
67	73	266.000	4	66.500
73	67	89.000	3	29.667
73	75	57.000	1	57.000
75	79	35.500	1	35.500
79	89	77.500	1	77.500
89	94	30.500	1	30.500
94	96	66.000	1	66.000
96	101	75.000	4	18.750
101	96	63.000	3	21.000
101	103	84.500	1	84.500
103	105	29.000	1	29.000
105	107	27.000	1	27.000
107	157	105.500	1	105.500

## 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
$L_{1-2} = 47$	155-13	1	1
$L_{2-3} = 93,5$	13-15, 15-21, 21-22	1	1
$L_{2-4} = 0$	13-110, 110-120	0	0
$L_{3-8} = 15,437$	22-24, 24-26	16	0,9412
$L_{3-11} = 126,5$	22-38, 38-44, 44-46	1	0.0588
$L_{4-5} = 0$	120-122, 122-124	0	0
$L_{4-6} = 0$	120-126	0	0
$L_{5-6} = 0$	124-126	0	0
$L_{6-7} = 0$	126-133, 133-157	0	0
$L_{8-9} = 22,5$	26-29, 29-32	16	1
$L_{8-10} = 0$	26-34	0	0
$L_{9-10} = 5,312$	32-34	16	1
$L_{10-3} = 25,563$	34-36, 36-22	16	1
$L_{11-12} = 14$	46-48	3	1
$L_{12-13} = 60,25$	48-50, 50-52	2	0.6667
$L_{12-14} = 57$	48-54	1	0.3333
$L_{13-14} = 26,25$	52-54	1	26.25
$L_{14-15} = 31,833$	54-58	3	1
$L_{15-11} = 40,5$	58-46	2	0.6667
$L_{15-16} = 153$	58-60, 60-65, 65-67	1	0.3333
$L_{16-17} = 66,5$	67-73	4	1
$L_{17-16} = 29,667$	73-67	3	0.75
$L_{17-18} = 92,5$	73-75, 75-79	1	0,25
$L_{18-19} = 0$	79-81, 81-83	0	0
$L_{18-21} = 0$	79-85, 85-87	0	0
$L_{18-22} = 108$	79-89, 89-94	1	1
$L_{19-20} = 0$	83-107	0	0
$L_{20-7} = 105,5$	107-157	1	1
$L_{21-20} = 0$	87-107	0	0
$L_{22-23} = 66$	94-96	1	1
$L_{22-25} = 0$	94-103	0	0

Таблица 2 (продолжение) – Расчет вероятностей и затрат ресурсов для дуг.

	Номера строк	Количество проходов	Вероятность
$L_{23-24} = 18,75$	96-101	4	1
$L_{24-23} = 21$	101-96	3	0.75
$L_{24-25} = 84,5$	101-103	1	1
$L_{25-20} = 56$	103-105, 105-107	1	1

## 6. Операционная графовая модель программы.

Операционная графовая модель программы представлена на рисунке 3.

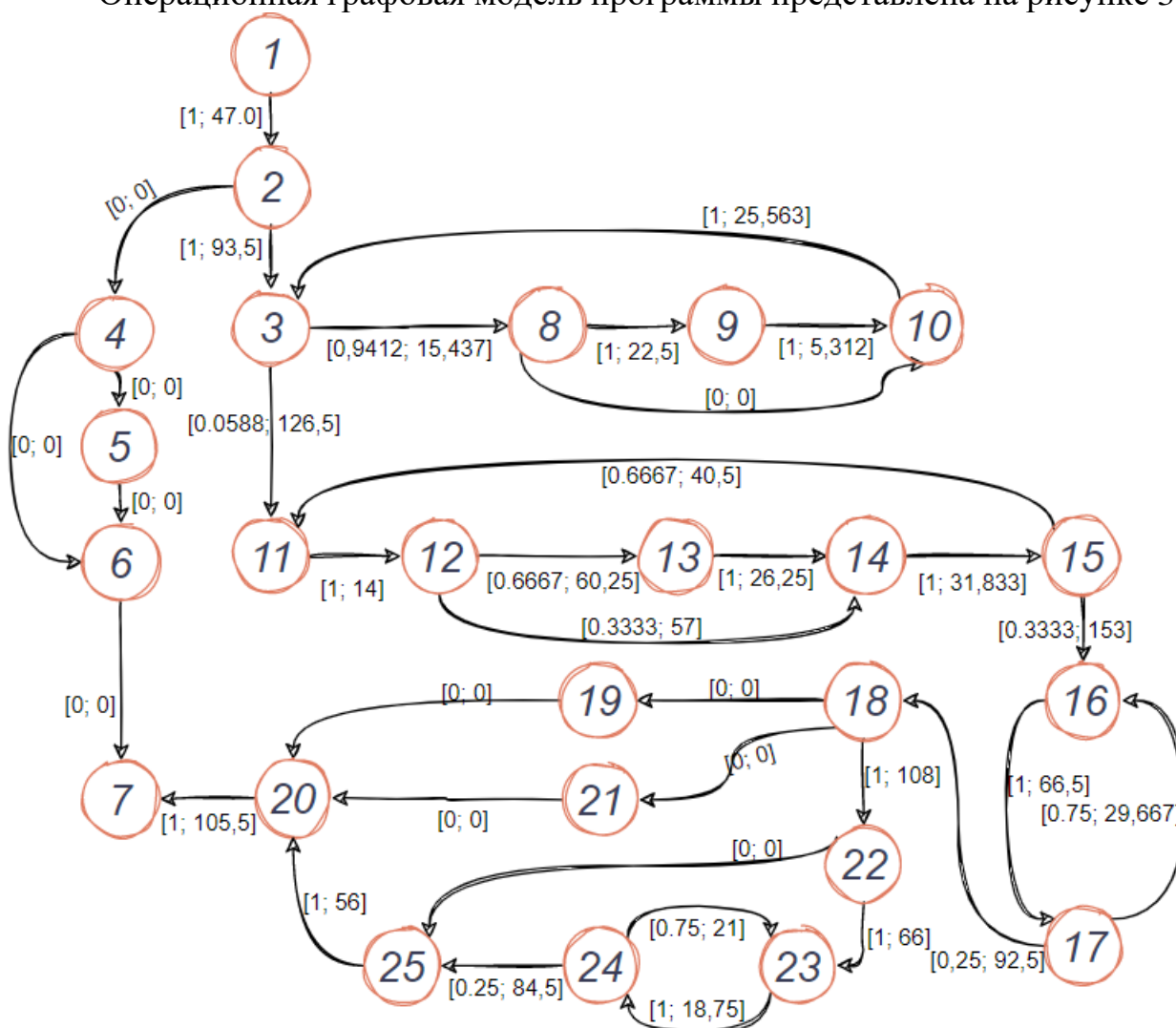


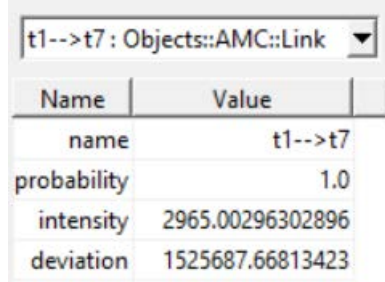
Рисунок 2 - Операционная модель

## 7. Описание модели model.xml.

Описание модели представлено в приложении В.

## 8. Результаты.

Результаты работы программы представлены на рисунке 4.



The screenshot shows a window titled 't1-->t7 : Objects::AMC::Link'. Inside the window is a table with two columns: 'Name' and 'Value'. The table contains four rows of data.

Name	Value
name	t1-->t7
probability	1.0
intensity	2965.00296302896
deviation	1525687.66813423

Рисунок 4 - Результат работы программы

Вероятность получилась равной 1. Согласно расчётам программы, среднее время выполнения составляет 2965 мкс. В пункте 4 данного отчёта приведен результат профилирования программы с использованием SAMPLER\_v2, где суммарное время выполнения составило 2975 мкс. В итоге, разница между результатами составляет менее 3 %.

## Выводы.

В ходе выполнения лабораторной работы была построена операционная графовая модель заданной программы, нагрузочные параметры которой были оценены с помощью профилировщика SAMPLER\_v2 и методом эквивалентных преобразований с помощью пакета CSA III были вычислены математическое ожидание и дисперсия времени выполнения как для всей программы, так и для фрагментов программы. Результаты сравнения этих характеристик с полученными в ходе выполнения лабораторной работы №3 согласуются.



## ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ

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

float bessy(float x, float n){
    const float small = 1.0E-8;
    const float euler = 0.57721566;
    const float pi = 3.1415926;
    const float pi2 = 0.63661977;

    float x2, sum, t, ts, term, xx, y0, y1, ya, yb, yc, ans;

    if(x<12){
        xx = 0.5 * x;
        x2 = xx * xx;
        t = log(xx) + euler;
        sum = 0.0;
        term = t;
        y0 = t;
        int j = 0;
        do{
            j = j+1;
            if(j != 1) sum = sum + 1/(j-1);
            ts = t-sum;
            term = -x2 * term / (j*j) * (1-1 / (j*ts));
            y0 = y0+term;
        }while(!(abs(term) < small));
        term = xx * (t-0.5);
        sum = 0.0;
        y1 = term;
        j = 1;
        do{
            j = j+1;
            sum = sum+1/(j-1);
            ts = t-sum;
            term = (-x2 * term) / (j * (j-1)) * ((ts-0.5 / j) / (ts + 0.5 / (j-
1))));
            y1 = y1+term;
        }while(!(abs(term) < small));
        y0 = pi2 * y0;
        y1 = pi2 * (y1 - 1/x);
        if(n == 0.0){
            ans = y0;
        }else if(n == 1.0){
            ans = y1;
        }else{
            ts = 2.0/x;
            ya = y0;
            yb = y1;
            int j=2;
            if(j<=trunc(n+0.01)){
                do{
                    yc = ts*(j-1)*yb-ya;
                    ya = yb;
                }while(j<=trunc(n+0.01));
            }
        }
    }
}
```

```

        yb = yc;
        j+=1;
    }while(j<=trunc(n+0.01));
}
ans = yc;
}
return ans;
}else{
    return sqrt(2 / (pi*x)) * sin(x - pi/4 - n * pi/2);
}
}

int main(int argc, char** argv){
    sampler_init(&argc, argv);
    float x, ordr;
    int done = 0;
    printf("\n");
    do{
        printf("Order? \n");
        scanf("%f", &ordr);
        if(ordr < 0.0){
            done = 1;
        }else{
            do{
                printf("Arg? \n");
                scanf("%f", &x);
            }while(!(x >= 0.0));
            SAMPLE;
            printf("Y Bessel is %f \n", bessy(x,ordr));
            SAMPLE;
        }
    }while(!(done));
    return 0;
}

```

## ПРИЛОЖЕНИЕ Б.

### КОД ПРОГРАММЫ С РАЗДЕЛЕНИЕМ НА ФУ

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

float bessy(float x, float n){
    const float small = 1.0E-8;
    const float euler = 0.57721566;
    const float pi = 3.1415926;
    const float pi2 = 0.63661977;

    float x2, sum, t, ts, term, xx, y0, y1, ya, yb, yc, ans;
SAMPLE;
    if(x<12){
        SAMPLE;
        xx = x/2;
        x2 = xx * xx;
        double myLog = 0.0;
        double a = 0.5;
        double u=(double)xx;
        SAMPLE;
        for(int i=0; SAMPLE, i<16; i++)
        {
            SAMPLE;
            u=u*u;
            SAMPLE;
            if(u>2.7182818)
            {
                SAMPLE;
                u=u/2.7182818;
                myLog+=a;
                SAMPLE;
            }
            SAMPLE;
            a/=2;
            SAMPLE;
        }
        SAMPLE;
        t = myLog + euler;
        sum = 0.0;
        term = t;
        y0 = t;
        int j = 0;
        SAMPLE;
        do{
            SAMPLE;
            j = j+1;
            SAMPLE;
            if(j != 1) {
                SAMPLE;
                sum = sum + 1/(j-1);
                SAMPLE;
            }
        }
```

```

        SAMPLE;
        ts = t-sum;
        term = -x2 * term / (j*j) * (1-1 / (j*ts));
        y0 = y0+term;
        SAMPLE;
    }while(!(abs(term) < small));
    SAMPLE;
    term = xx * (t-0.5);
    sum = 0.0;
    y1 = term;
    j = 1;
    SAMPLE;
    do{
        SAMPLE;
        j = j+1;
        sum = sum+1/(j-1);
        ts = t-sum;
        term = (-x2 * term) / (j * (j-1)) * ((ts-0.5 / j) / (ts + 0.5 / (j-
1)))));

        y1 = y1+term;
        SAMPLE;
    }while(!(abs(term) < small));
    SAMPLE;
    y0 = pi2 * y0;
    y1 = pi2 * (y1 - 1/x);
    float res = 0;
    SAMPLE;
    if(n == 0.0){
        SAMPLE;
        res = y0;
        SAMPLE;
    }else if(n == 1.0){
        SAMPLE;
        res = y1;
        SAMPLE;
    }else{
        SAMPLE;
        ts = 2.0/x;
        ya = y0;
        yb = y1;
        int j=2;
        SAMPLE;
        while(j<=(int)(n+0.01)){
            SAMPLE;
            yc = ts*(j-1)*yb-ya;
            ya = yb;
            yb = yc;
            j+=1;
            SAMPLE;
        }
        SAMPLE;
        res = yc;
        SAMPLE;
    }
    SAMPLE;
    return res;
}else{
    SAMPLE;

```

```

float mySqrt = 2 / (pi*x);
__asm__ ( "fsqrt" : "+t" (mySqrt) );

float toSin = x - pi/4 - n * pi/2;
toSin *= 0.63661977236758134308; // 2/Pi
int sign = toSin < 0.0;
toSin = sign ? -toSin : toSin;
int xf = (int)toSin;
toSin -= xf;
SAMPLE;
if ((xf & 1) == 1){
    SAMPLE;
    toSin = 1 - toSin;
    SAMPLE;
}
SAMPLE;
int per = ((xf >> 1) & 1) == 1;
float xxForSin = toSin * toSin;
float y = toSin * (1.5707903005870776 + xxForSin * (-0.6458858977085938
+
xxForSin*(0.07941798513358536 - 0.0043223880120647346 * xxForSin)));
float mySin = sign ^ per ? -y : y;
float res = mySqrt * mySin;
SAMPLE;
return res;
}
}

int main(int argc, char** argv){
    sampler_init(&argc, argv);
    float x, ordr;
    int done = 0;
    printf("\n");
    do{
        printf("Order? \n");
        scanf("%f", &ordr);
        if(ordr < 0.0){
            done = 1;
        }else{
            do{
                printf("Arg? \n");
                scanf("%f", &x);
            }while(!(x >= 0.0));
            SAMPLE;
            float res = bessy(x,ordr);
            SAMPLE;
            printf("Y Bessel is %f \n", res);
        }
    }while(!(done));
    return 0;
}

```

## ПРИЛОЖЕНИЕ В.

### 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 = "t12"></node>
<node type = "Objects::AMC::Top" name = "t11"></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 = "t17"></node>
<node type = "Objects::AMC::Top" name = "t18"></node>
<node type = "Objects::AMC::Top" name = "t19"></node>
<node type = "Objects::AMC::Top" name = "t20"></node>
<node type = "Objects::AMC::Top" name = "t21"></node>
<node type = "Objects::AMC::Top" name = "t22"></node>
<node type = "Objects::AMC::Top" name = "t23"></node>
<node type = "Objects::AMC::Top" name = "t24"></node>
<node type = "Objects::AMC::Top" name = "t25"></node>
<link type = "Objects::AMC::Link" name = "t1-->t2" probability = "1.0"
intensity = "47.0" deviation = "0.0" source = "t1" dest =
"t2"></link>
<link type = "Objects::AMC::Link" name = "t2-->t3" probability = "1.0"
intensity = "93.5" deviation = "0.0" source = "t2" dest =
"t3"></link>
<link type = "Objects::AMC::Link" name = "t2-->t4" probability = "0.0"
intensity = "0.0" deviation = "0.0" source = "t2" dest =
"t4"></link>
<link type = "Objects::AMC::Link" name = "t3-->t8" probability =
"0.9412" intensity = "15.437" deviation = "0.0" source = "t3"
dest = "t8"></link>
<link type = "Objects::AMC::Link" name = "t3-->t11" probability =
"0.0588" intensity = "126.5" deviation = "0.0" source = "t3" dest
= "t11"></link>
<link type = "Objects::AMC::Link" name = "t4-->t5" probability = "0.0"
intensity = "0.0" deviation = "0.0" source = "t4" dest =
"t5"></link>
<link type = "Objects::AMC::Link" name = "t4-->t6" probability = "0.0"
intensity = "0.0" deviation = "0.0" source = "t4" dest =
"t6"></link>
<link type = "Objects::AMC::Link" name = "t5-->t6" probability = "0.0"
intensity = "0.0" deviation = "0.0" source = "t5" dest =
```

```

    "t6"></link>
<link type = "Objects::AMC::Link" name = "t6-->t7" probability = "0.0"
intensity = "0.0" deviation = "0.0" source = "t6" dest =
    "t7"></link>
<link type = "Objects::AMC::Link" name = "t8-->t9" probability = "1.0"
intensity = "22.5" deviation = "0.0" source = "t8" dest =
    "t9"></link>
<link type = "Objects::AMC::Link" name = "t8-->t10" probability = "0.0"
intensity = "0.0" deviation = "0.0" source = "t8" dest =
    "t10"></link>
<link type = "Objects::AMC::Link" name = "t9-->t10" probability = "1.0"
intensity = "5.312" deviation = "0.0" source = "t9" dest =
    "t10"></link>
<link type = "Objects::AMC::Link" name = "t10-->t3" probability = "1.0"
intensity = "25.563" deviation = "0.0" source = "t10" dest =
    "t3"></link>
<link type = "Objects::AMC::Link" name = "t11-->t12" probability =
    "1.0" intensity = "14.0" deviation = "0.0" source = "t11" dest =
    "t12"></link>
<link type = "Objects::AMC::Link" name = "t12-->t13" probability =
    "0.6667" intensity = "60.25" deviation = "0.0" source = "t12"
dest = "t13"></link>
<link type = "Objects::AMC::Link" name = "t12-->t14" probability =
    "0.3333" intensity = "57.0" deviation = "0.0" source = "t12" dest
= "t14"></link>
<link type = "Objects::AMC::Link" name = "t13-->t14" probability =
    "1.0" intensity = "26.25" deviation = "0.0" source = "t13" dest =
    "t14"></link>
<link type = "Objects::AMC::Link" name = "t14-->t15" probability =
    "1.0" intensity = "31.833" deviation = "0.0" source = "t14" dest
= "t15"></link>
<link type = "Objects::AMC::Link" name = "t15-->t11" probability =
    "0.6667" intensity = "10.5" deviation = "0.0" source = "t15" dest
= "t11"></link>
<link type = "Objects::AMC::Link" name = "t15-->t16" probability =
    "0.3333" intensity = "153.0" deviation = "0.0" source = "t15"
dest = "t16"></link>
<link type = "Objects::AMC::Link" name = "t16-->t17" probability =
    "1.0" intensity = "66.5" deviation = "0.0" source = "t16" dest =
    "t17"></link>
<link type = "Objects::AMC::Link" name = "t17-->t16" probability =
    "0.75" intensity = "29.667" deviation = "0.0" source = "t17" dest
= "t16"></link>
<link type = "Objects::AMC::Link" name = "t17-->t18" probability =
    "0.25" intensity = "92.5" deviation = "0.0" source = "t17" dest =
    "t18"></link>
<link type = "Objects::AMC::Link" name = "t18-->t22" probability =
    "1.0" intensity = "108.0" deviation = "0.0" source = "t18" dest =
    "t22"></link>
<link type = "Objects::AMC::Link" name = "t18-->t19" probability =
    "0.0" intensity = "0.0" deviation = "0.0" source = "t18" dest =
    "t19"></link>
<link type = "Objects::AMC::Link" name = "t18-->t21" probability =
    "0.0" intensity = "0.0" deviation = "0.0" source = "t18" dest =

```

```

    "t21"></link>
<link type = "Objects::AMC::Link" name = "t19-->t20" probability =
    "0.0" intensity = "0.0" deviation = "0.0" source = "t19" dest =
    "t20"></link>
<link type = "Objects::AMC::Link" name = "t20-->t7" probability = "1.0"
    intensity = "105.5" deviation = "0.0" source = "t20" dest =
    "t7"></link>
<link type = "Objects::AMC::Link" name = "t21-->t20" probability =
    "0.0" intensity = "0.0" deviation = "0.0" source = "t21" dest =
    "t20"></link>
<link type = "Objects::AMC::Link" name = "t22-->t25" probability =
    "0.0" intensity = "0.0" deviation = "0.0" source = "t22" dest =
    "t25"></link>
<link type = "Objects::AMC::Link" name = "t22-->t23" probability =
    "1.0" intensity = "115.5" deviation = "0.0" source = "t22" dest =
    "t23"></link>
<link type = "Objects::AMC::Link" name = "t23-->t24" probability =
    "1.0" intensity = "18.75" deviation = "0.0" source = "t23" dest =
    "t24"></link>
<link type = "Objects::AMC::Link" name = "t24-->t23" probability =
    "0.75" intensity = "21.0" deviation = "0.0" source = "t24" dest =
    "t23"></link>
<link type = "Objects::AMC::Link" name = "t24-->t25" probability =
    "0.25" intensity = "84.5" deviation = "0.0" source = "t24" dest =
    "t25"></link>
<link type = "Objects::AMC::Link" name = "t25-->t20" probability =
    "1.0" intensity = "56.0" deviation = "0.0" source = "t25" dest =
    "t20"></link>
</model>

```