

Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Сибирский государственный университет
телекоммуникаций и информатики»
(СибГУТИ)

Кафедра прикладной математики и кибернетики
Современные технологии программирования

Практическая работа №11
«Вероятностное моделирование метрических характеристик программ»

Выполнил: студент 4 курса
группы ИП-111
Кузьменок Денис Витальевич

Проверил преподаватель:
Зайцев Михаил Георгиевич

Новосибирск, 2024 г.

Цель:

Целью работы является разработка программы, имитирующей процесс написания программы программистом, и расчет статистических оценок метрических характеристик полученных программ. Для имитации используется вероятностная модель выборки с возвратом из генеральной совокупности, состоящей из η символов, пока последняя не будет исчерпана, а также проведение серий испытаний при разных значениях η .

Задание

1. Разработать программу для вероятностного моделирования процесса написания программы программистом с длиной словаря программы $\eta = 16, 32, 64, 128$.
2. С помощью разработанной программы получить статистические оценки:
 - длины программы L ,
 - дисперсии длины $D(L\eta)$,
 - среднеквадратического отклонения ($\sqrt{D(L\eta)}$),
 - относительной ожидаемой погрешности δ .
3. С помощью приведенных формул получить теоретические значения и сравнить их с результатами моделирования.
4. По тексту разработанной программы посчитать длину ее словаря и длину программы. Рассчитать длину программы по размеру ее словаря с помощью приведенных формул. Сравнить посчитанное по тексту значение длины текста программы, с длиной текста программы, полученной по формуле.
5. По первому и второму пунктам задания определить $\eta^* 2$ – число единых по смыслу входных и выходных параметров представленных в сжатой без избыточной формы. Сравнить прогнозируемую длину программы с длиной программы, рассчитанной по тексту программы.

Ход работы

Программа на языке С# показала следующие результаты:

Для словаря длины 16:

Теоретическая длина программы = 57,6

Эмулированная длина программы = 55

Теоретическая дисперсия = 421,103121113146

Стандартное отклонение = 20,5207972825898

Относительная погрешность = 0,125 (в процентах = 12,5%)

Для словаря длины 32:

Теоретическая длина программы = 144

Эмулированная длина программы = 134

Теоретическая дисперсия = 1684,41248445258

Стандартное отклонение = 41,0415945651797

Относительная погрешность = 0,1 (в процентах = 10%)

Для словаря длины 64:

Теоретическая длина программы = 345,6

Эмулированная длина программы = 322

Теоретическая дисперсия = 6737,64993781034

Стандартное отклонение = 82,0831891303593

Относительная погрешность = 0,0833333333333333 (в процентах = 8,33333333333333%)

Для словаря длины 128:

Теоретическая длина программы = 806,4

Эмулированная длина программы = 763

Теоретическая дисперсия = 26950,5997512413

Стандартное отклонение = 164,166378260719

Относительная погрешность = 0,0714285714285714 (в процентах = 7,14285714285714%)

Вывод:

В ходе выполнения лабораторной работы была разработана программа для вероятностного моделирования процесса написания программ программистом. Основной целью работы было исследование метрических характеристик программ, включая длину программы, дисперсию, среднее квадратическое отклонение и относительную ожидаемую погрешность.

Листинг программы

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab11
{
    class Program
    {
        private static void Main(string[] args)
        {
            List<int> dictionarySizes = new List<int>()
            {
                16,
                32,
                64,
                128
            };

            foreach(int element in dictionarySizes)
            {
                double theoreticalLength = CalculateProgramLength(element / 2,
element / 2);
                double theoreticalVariance = CalculateVariance(element);
                double standardDeviation = Math.Sqrt(theoreticalVariance);
                double delta = 1.0 / (2 * Math.Log(element, 2));

                double simulatedLength = SimulateProgramLength(element);

                Console.WriteLine($"Для словаря длины {element}: ");
                Console.WriteLine($"Теоретическая длина программы =
{theoreticalLength}");
                Console.WriteLine($"Эмулированная длина программы =
{simulatedLength}");
                Console.WriteLine($"Теоретическая дисперсия =
{theoreticalVariance}");
                Console.WriteLine($"Стандартное отклонение =
{standardDeviation}");
                Console.WriteLine($"Относительная погрешность = {delta} (в
процентах = {delta * 100}%)");
                Console.WriteLine("-----\n");
            }

            private static double CalculateProgramLength(int operators, int
operands)
            {
                double eta = operators + operands;
                return 0.9 * eta * Math.Log(eta, 2);
            }

            private static double CalculateVariance(int eta)
            {
                return (Math.PI * Math.PI * eta * eta) / 6.0;
            }
        }
    }
}
```

```

    }

    private static double SimulateProgramLength(int eta)
    {
        List<bool> dictionary = new List<bool>(eta);

        for (int i = 0; i < eta; i++)
        {
            dictionary.Add(false);
        }

        int drawn = 0, length = 0;

        Random rnd = new Random();

        while (drawn < eta)
        {
            int randomIndex = rnd.Next(eta);
            length++;
            if (!dictionary[randomIndex])
            {
                dictionary[randomIndex] = true;
                drawn++;
            }
        }

        return length;
    }
}

```