МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Ф. Уткина»

Кафедра САПР ВС

КУРСОВАЯ РАБОТА

по дисциплине

«Информационные технологии»

на тему:

**«**Разработка программной реализации минимизации

слабоопределенных булевых функций, заданных десятичными интервалами.»

Выполнил:

Студент группы 8410

Проверил:

Рязань 2021г.

Оглавление

[Введение. 3](#_Toc91263185)

[Теоретическая часть алгоритма. 4](#_Toc91263186)

[Блок-схема разрабатываемого алгоритма 5](#_Toc91263187)

[Решение контрольного примера 7](#_Toc91263188)

[Заключение 8](#_Toc91263189)

[Список литературы 9](#_Toc91263190)

[Приложение 10](#_Toc91263191)

# Введение.

Дискретная математика – это изучение математики, ограниченное набором целых чисел. В то время как приложения таких областей непрерывной математики, как исчисление и алгебра, очевидны для многих, приложения дискретной математики поначалу могут быть неясными. Тем не менее, дискретная математика лежит в основе многих реальных научных областей, особенно информатики. Основные методы, изученные в рамках курса дискретной математики, могут применяться во многих различных областях.

Алгоритмы - это правила, по которым работает компьютер. Эти правила созданы с помощью законов дискретной математики. Программист использует дискретную математику для разработки эффективных алгоритмов. Этот дизайн включает применение дискретной математики для определения количества шагов, которые алгоритм должен выполнить, что подразумевает скорость алгоритма. Благодаря дискретным математическим приложениям в алгоритмах современные компьютеры работают быстрее, чем когда-либо прежде.

Основываясь на этом можно выделить цель работы – реализовать алгоритм минимизации слабоопределенных булевых функций основываясь на языке C++.

**Язык C++** был разработан Бьёрном Страуструпом в подразделении Bell Labs компании AT&T в качестве дополнения к языку Cи. С++ добавил множество новых возможностей в язык Си. Его популярность была вызвана объектно-ориентированностью языка. Сейчас C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. С его помощью создают операционные системы, разнообразные прикладные программы, драйверы устройств, игры и пр.

Среди самого популярного программного обеспечения, написанного на C++ (или с его использованием), находятся СУБД MySQL, интернет-браузер Mozilla Firefox, большая часть программного обеспечения от Microsoft: операционные системы семейства Windows, IDE Visual Studio, Internet Explorer, Microsoft Office. Adobe Photoshop, Adobe Illustrator и Adobe Premiere Pro целиком написаны на C++. Также данный язык лежит в основе игрового движка Unity.

В ходе выполнения работы необходимо выполнить следующие задачи:

1. Реализовать алгоритм поглощения формул,
2. Реализовать метод заполнения десятичными интервлами.

# Теоретическая часть алгоритма.

В курсе математического анализа изучаются функции, определённые на числовой прямой или на отрезке числовой прямой или на (гипер-) плоскости и т.п. Так или иначе область определения – *непрерывное* множество. В курсе дискретной математики изучаться должны функции, область определения которых – *дискретное* множество. Простейшим (но нетривиальным) таким множеством является множество, состоящее из двух элементов. Так мы и приходим к понятию булевой функции.

*Булевой функцией* от *n* аргументов называется функция *f* из *n*-ой степени множества { 0, 1 } в множество { 0, 1 }.

Минимизация функций будет осуществляться по методу Блейка.

Алгоритм Блейка –Порецкого.

*Начало*. Задана произвольная ДНФ булевой функции.

*Шаг 1*. Построим список троичных векторов, представляющих конъюнкции ДНФ. Удалим из списка все векторы, поглощаемые другими. Если останется лишь один вектор, пойдем на конец. Иначе обозначим второй из оставшихся векторов через α.

*Шаг 2*. Найдем для вектора α очередной смежный вектор β среди векторов, расположенных в списке выше α. Если такого вектора β нет, то идем на шаг 5. Иначе обобщенно склеим α и β и полученный вектор γ припишем к списку последним.

*Шаг 3*. Если вектор γ поглощается хотя бы одним вектором из списка, то удалим γ (в частном случае γ может совпадать с одним из векторов списка, тогда удалим именно приписанный вектор γ, иначе произойдет "зацикливание" алгоритма, так как вектор γ будет вновь появляться в списке) и идем на шаг 2. Если вектор γ не поглощается, то удалим все векторы, поглощаемые им.

*Шаг 4*. Если вектор α не удален, то идем на шаг 2.

*Шаг 5*. Если вектор α был не последним в списке, то выберем в качестве нового α следующий по списку вектор и вернемся на шаг 2.

*Конец*. Неудаленные векторы задают сокращенную ДНФ функции.

# Блок-схема разрабатываемого алгоритма

Рисунок 1-2 – блок-схема программы и алгоритма.



Рисунок 1 – Схема программы



Рисунок 2 - Схема работы функции создания минимизированной функции

# Решение контрольного примера

Минимизировать формулу, находящуюся в СДНФ (Рисунок 3):

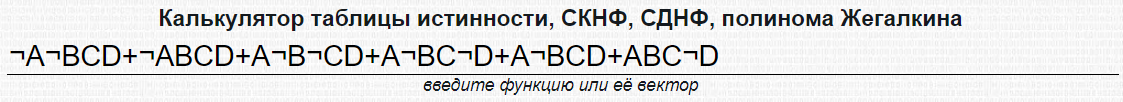


Рисунок 3 - Формула в калькуляторе

Которую также можно представить в виде чисел: 3, 7, 9, 10, 11, 14.

Результат выполнения:

On what interval to define the function?

[0] - exit

[1] - 0-7

[2] - 0-15

2

Enter size:

6

[0] = 3

[1] = 7

[2] = 9

[3] = 10

[4] = 11

[5] = 14

[0] = 0 0 1 1

[1] = 0 1 1 1

[2] = 1 0 0 1

[3] = 1 0 1 0

[4] = 1 0 1 1

[5] = 1 1 1 0

Result:

[0] = 0 1 1

[1] = 1 0 1

[2] = 1 1 0

# Заключение

В результате выполнения курсовой работы была создана программа, реализующая минимизацию слабоопределенных функций, задаваемых десятичными интервалами, на основе алгоритма Блейка Порецкого на языке программирования C++. В ходе выполнения были использованы все преимущества языка, что позволило создать простую, но эффективную программу.

В результате была получено программное обеспечение, автоматически выдающее результаты аналогичные решению вручную, что в век компьютерных технологий очень важно.

# Список литературы

1. Страуструп, Б. Программирование: принципы и практика использования C++, испр. изд. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2011. — 1248 с.
2. Шилдт, Г. C++ для начинающих. Серия «Шаг за шагом»/ Шилдт Г.; пер. с англ. – М.: ЭКОМ Паблишерз, 2013. – 640 с.: ил.
3. Прата, С. Язык программирования С++. Лекции и упражнения, 6-е изд. : Пер. с англ. – М. : ООО «И.Д. Вильямс», 2012. – 1248 с. : ил. – Парал. Тит. Англ.

# Приложение

Текст программы:

#include <iostream>

#include <vector>

#include <string>

using namespace std;

typedef vector< vector<int> > Matrix;

typedef vector<int> Array;

void dec2bin(int num, int length, Array& result)

{ //получение десятичного числа в двоичном виде

int k = 1, b = 0;

Array bin;

while (num)

{

b += (num % 2) \* k;

k \*= 10;

num /= 2;

}

while(b)

{

bin.push\_back(b % 10);

b = b / 10;

}

while (bin.size() < length){

bin.push\_back(0);

}

for (int i = bin.size()-1; i>=0; i--){

result.push\_back(bin[i]);

}

}

void print(const Matrix& mat){

for (int i = 0; i<mat.size(); i++){

cout << "[" << i << "] = ";

for (int j = 0; j<mat[i].size(); j++)

cout << mat[i][j] << " ";

cout << endl;

}

cout << endl;

}

bool areShrinking(const Array& array1, const Array& array2){

bool oneDifference = false; //проверка, сокращаются ли формулы

if (array1.size() == array2.size()){

for (int i = 0; i<array1.size(); i++){

if (oneDifference == false){

if (array1[i] != array2[i]) oneDifference = true;

}

else{

if (array1[i] != array2[i]) return false;

}

}

}

return oneDifference;

}

Array getShrink(const Array& array1, const Array& array2){

Array result; //реализация поглощения

for (int i = 0; i<array1.size(); i++){

if (array1[i] == array2[i])

result.push\_back(array1[i]);

}

return result;

}

bool inArray(const Array& array, int el){

for (int i = 0; i<array.size(); i++)

if (array[i] == el) return true;

}

Matrix getMinFunc(const Matrix& mat){

Matrix newMat;

Array visitingIndex;

int i=0, j=0;

while (i < mat.size())

{

j = i + 1;

while (j < mat.size())

{

if ((!inArray(visitingIndex, i)) && (!inArray(visitingIndex, j)))

if (areShrinking(mat[i], mat[j])){

visitingIndex.push\_back(i);

visitingIndex.push\_back(j);

newMat.push\_back(getShrink(mat[i], mat[j]));

}

j++;

}

i++;

}

for ( i=0; i<mat.size(); i++){

if (!inArray(visitingIndex, i)) newMat.push\_back(mat[i]);

}

if (visitingIndex.size() == 0)

return mat;

else return getMinFunc(newMat);

}

int main(){

while (true){

int action, length;

cout << "On what interval to define the function?" << endl;

cout << "[0] - exit" << endl;

cout << "[1] - 0-7" << endl;

cout << "[2] - 0-15" << endl;

cin >> action;

if (action == 0) break;

length = 3+action-1;

int size;

cout << "Enter size:" << endl;

cin >> size;

Matrix matrix(size);

for (int i=0; i<size; i++){

int num;

cout << "[" << i << "] = ";

cin >> num;

if ((action == 1))

if ((num >= 0) && (num <= 7))

dec2bin(num, length, matrix[i]);

else break;

if ((action == 2))

if ((num >= 0) && (num <= 15))

dec2bin(num, length, matrix[i]);

else break;

}

cout << endl;

print(matrix);

Matrix resMatrix = getMinFunc(matrix);

cout << "Result" << endl;

print(resMatrix);

}

return 0;

}