Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

"Белгородский государственный технологический университет им. В. Г. Шухова"

(БГТУ им. В.Г. Шухова)

Институт энергетики, информационных технологий и управляющих систем

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

**Лабораторная работа № 1.4**

**по дисциплине дискретная математика**

**тема: Теоретико-множественные уравнения**

**Выполнил: студент группы ПВ-223**

**Игнатьев Артур Олегович**

**Проверил: доцент   
Рязанов Юрий Дмитриевич**

**старший преподаватель**

**Бондаренко Татьяна Владимировна**

Белгород 2022

Лабораторная работа № 1.4

**Тема:** Теоретико-множественные уравнения

**Цель работы:** научиться решать теоретико-множественные уравне-

ния с применением ЭВМ.

**Задания**

1. Преобразовать исходное уравнение (см. “Варианты заданий”) в уравнение с пустой правой частью.

2. Преобразовать левую часть уравнения к виду используя разложение Шеннона по неизвестному множеству X.

3. Написать программу, вычисляющую значения множеств **и**  при заданных исходных множествах.

4. Вычислить значения множеств **и**  и сделать вывод о существовании решения уравнения. Если решения уравнения не существует, то выполнить п.п. 1—4 для следующего (предыдущего) варианта.

5. Определить мощность общего решения, найти некоторые (или все) частные решения, в том числе частные решения наименьшей и наибольшей мощности.

6. Написать программу для проверки найденных решений.

Вариант 3

U={1,2,3,4,5,6,7,8,9,10}

A={1,5,7} B={2,4,6,10} C={1,3,5,6,8,10}

X-?

Решение заданий:

**1. Преобразовать исходное уравнение (см. “Варианты заданий”) в уравнение с пустой правой частью.**

**2. Преобразовать левую часть уравнения к виду используя разложение Шеннона по неизвестному множеству X.**

**3. Написать программу, вычисляющую значения множеств и при заданных исходных множествах.**

Программа была выполнена на python для большей простоты реализации:

U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}  
A = {1, 5, 7}  
B = {2, 4, 6, 10}  
C = {1, 3, 5, 6, 8, 10}  
  
# φ^∅=(A - (B - C)) ∆ (A - (B ⋂ C))  
phi\_empty = (A - (B - C)) ^ (A - (B & C))  
  
# ¯(φ^U ) = U - φ^U  
phi\_universal\_complement = U - ((A - B) ^ A)  
  
print("φ^∅ =", phi\_empty)  
print("¯(φ^U ) =", phi\_universal\_complement)

Результат выполнения программы:



**4. Вычислить значения множеств и и сделать вывод о существовании решения уравнения. Если решения уравнения не существует, то выполнить п.п. 1—4 для следующего (предыдущего) варианта.**

Для вычисления значений множеств и , мы должны использовать данные значения множеств A, B, C и U. Начнем с вычисления значения множества

Таким образом, значение множества равно пустому множеству.

Теперь, вычислим значение множества :

Таким образом, дополнение множества совпадает с универсальным множеством U.

Таким образом, значения множеств и равны соответственно ∅ и U.

**5. Определить мощность общего решения, найти некоторые (или все) частные решения, в том числе частные решения наименьшей и наибольшей мощности.**

Для получения общего решения необходимо каждое подмножество множества объединить с φ^∅= ∅

Мощность общего решения будет равняться: 2^10= 1024

Найдём некоторые частные решения:

{{∅},{1},{2},{3},{4},{5},{6},{7},{8},{9},{10},{1,2},{1,3},…,{1,2,3,4,5,6,7,8,9,10}}

Частное решение наименьшей мощности:

Частное решение наибольшей мощности:

**6. Написать программу для проверки найденных решений.**

#include <stdio.h>  
#include <windows.h>  
  
typedef struct {  
 int values[10];  
 int size;  
} Set;  
  
Set setDifference(Set A, Set B) {  
 Set result;  
 result.size = 0;  
  
 for (int i = 0; i < A.size; i++) {  
 int isInB = 0;  
 for (int j = 0; j < B.size; j++) {  
 if (B.values[j] == A.values[i]) {  
 isInB = 1;  
 break;  
 }  
 }  
 if (!isInB) {  
 result.values[result.size] = A.values[i];  
 result.size++;  
 }  
 }  
  
 return result;  
}  
  
Set setIntersection(Set A, Set B) {  
 Set result;  
 result.size = 0;  
  
 for (int i = 0; i < A.size; i++) {  
 int isInB = 0;  
 for (int j = 0; j < B.size; j++) {  
 if (B.values[j] == A.values[i]) {  
 isInB = 1;  
 break;  
 }  
 }  
 if (isInB) {  
 result.values[result.size] = A.values[i];  
 result.size++;  
 }  
 }  
  
 return result;  
}  
  
int main() {  
 SetConsoleOutputCP(CP\_UTF8);  
  
 Set U = {{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, 10};  
 Set A = {{1, 5, 7}, 3};  
 Set B = {{2, 4, 6, 10}, 4};  
 Set C = {{1, 3, 5, 6, 8, 10}, 6};  
  
 int correct\_solution = 1;  
  
 for (int i = 0; i < U.size; i++) {  
 int X = U.values[i];  
  
 Set CMinusX;  
 CMinusX.size = C.size;  
 for (int j = 0; j < C.size; j++) {  
 if (C.values[j] != X) {  
 CMinusX.values[CMinusX.size] = C.values[j];  
 CMinusX.size++;  
 }  
 }  
  
 Set BMinusCMinusX = setDifference(B, CMinusX);  
  
 Set equationLeft = setDifference(A, BMinusCMinusX);  
 Set equationRight = setDifference(A, setIntersection(B, CMinusX));  
  
 if (equationLeft.size != equationRight.size) {  
 correct\_solution = 0;  
 break;  
 }  
  
 for (int j = 0; j < equationLeft.size; j++) {  
 int found = 0;  
 for (int k = 0; k < equationRight.size; k++) {  
 if (equationRight.values[k] == equationLeft.values[j]) {  
 found = 1;  
 break;  
 }  
 }  
 if (!found) {  
 correct\_solution = 0;  
 break;  
 }  
 }  
  
 if (!correct\_solution) {  
 break;  
 }  
 }  
  
 if (correct\_solution) {  
 printf("Найденное решение верно.\n");  
 } else {  
 printf("Найдено неправильное решение.\n");  
 }  
  
 return 0;  
}

Результат выполнения программы:



Вывод: на этой лабораторной работе я научился решать теоретико-множественные уравнения с применением ЭВМ.