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 № 1.3**

**по дисциплине дискретная математика**

**тема: Теоретико-множественные тождества**

**Выполнил: студент группы ПВ-223**

**Игнатьев Артур Олегович**

**Проверил: доцент   
Рязанов Юрий Дмитриевич**

**старший преподаватель**

**Бондаренко Татьяна Владимировна**

Белгород 2022

Лабораторная работа № 1.3

**Тема:** Теоретико-множественные тождества

**Цель работы:** изучить методы доказательства теоретико-множественных тождеств.

**Задания**

1. На рис.1 изображены круги Эйлера, соответствующие множествам А, В и С, с пронумерованными элементарными областями (не содержащими внутри себя других областей). Заштриховать элементарные области в соответствии с вариантом задания (см. табл.2).

Изображение выглядит как круг, зарисовка, диаграмма, рисунок

Автоматически созданное описание

Рис.1. Круги Эйлера, соответствующие множествам А, В и С с пронумерованными элементарными областями

2. Написать выражение 1 над множествами А, В и С, определяющее заштрихованную область, используя операции пересечения, объединения и дополнения.

3. Используя свойства операций над множествами, преобразовать выражение 1 в выражение 2, не содержащее операции дополнения множества.

4. Используя свойства операций над множествами, преобразовать выражение 2 в выражение 3, не содержащее операции объединения множеств.

5. Используя свойства операций над множествами, преобразовать выражение 3 в выражение 4, не содержащее операции пересечения множеств.

6. Доказать тождественность выражений 2 и 3 методом характеристических функций.

7. Доказать тождественность выражений 2 и 4 методом логических функций. Для автоматизации доказательства написать программу, которая получает и сравнивает таблицы истинности логических функций.

8. Доказать тождественность выражений 3 и 4 теоретико-множественным методом. Для автоматизации доказательства написать программу, в которой вычисляются и сравниваются значения выражений 3 и 4 при А={1,3,5,7}, B={2,3,6,7} и C={4,5,6,7}.

Вариант 3

Номера областей: 1, 2, 7

Решение заданий:

**1. На рис.1 изображены круги Эйлера, соответствующие множествам А, В и С, с пронумерованными элементарными областями (не содержащими внутри себя других областей). Заштриховать элементарные области в соответствии с вариантом задания (см. табл.2).**

Изображение выглядит как круг, диаграмма, графическая вставка, Графика

Автоматически созданное описание

**2. Написать выражение 1 над множествами А, В и С, определяющее заштрихованную область, используя операции пересечения, объединения и дополнения.**

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

**3. Используя свойства операций над множествами, преобразовать выражение 1 в выражение 2, не содержащее операции дополнения множества.**

**4. Используя свойства операций над множествами, преобразовать выражение 2 в выражение 3, не содержащее операции объединения множеств**

**5. Используя свойства операций над множествами, преобразовать выражение 3 в выражение 4, не содержащее операции пересечения множеств**

**6. Доказать тождественность выражений 2 и 3 методом характеристических функций.**

Рассмотрим выражение . Выразим его с использованием характеристических функций:

Разберемся с каждой частью выражения по отдельности.

Характеристическая функция :

Используем здесь свойство объединения множеств и дополнения. Обратите внимание, что равны нулю на пересечении множеств A и B.

Характеристическая функция C:

Объединим все вместе:

Теперь рассмотрим второе выражение :

Сравним оба выражения:

Мы видим, что оба выражения равны друг другу, поэтому мы доказали тождественность исходных выражений с использованием характеристических функций.

**7. Доказать тождественность выражений 2 и 4 методом логических функций. Для автоматизации доказательства написать программу, которая получает и сравнивает таблицы истинности логических функций.**

Для доказательства тождественности выражений (A∩C)∪(B∩C) - (B∩A)∩C и ((A∪B)∩(A∪C)) - (A∩B∩C) методом логических функций, мы можем воспользоваться правилами алгебры множеств и логическими операциями. Рассмотрим каждое выражение по отдельности.

Выражение 1:

Выражение 2:

Выражения не тождественны (A∪B)∩C ≠ A∪C. {5, 6, 7}≠ {1, 3, 4, 5, 6, 7}.

Программная реализация:

#include <stdio.h>  
#include <windows.h>  
// Функция для вычисления значения логического выражения ((A∩C)∪(B∩C) - (B∩A)∩C)  
int expression1(int A, int B, int C) {  
 int intersection1 = (A && C);  
 int intersection2 = (B && C);  
 int union1 = (intersection1 || intersection2);  
 int intersection3 = (B && A && C);  
 int result = union1 && !intersection3;  
 return result;  
}  
  
// Функция для вычисления значения логического выражения (((A∪B)∩(A∪C)) - (A∩B∩C))  
int expression2(int A, int B, int C) {  
 int union1 = (A || B);  
 int union2 = (A || C);  
 int intersection1 = (union1 && union2);  
 int intersection2 = (A && B && C);  
 int result = intersection1 && !intersection2;  
 return result;  
}  
  
int main() {  
 SetConsoleOutputCP(CP\_UTF8);  
  
 printf("A B C | ((A∩C)∪(B∩C) - (B∩A)∩C) | (((A∪B)∩(A∪C)) - (A∩B∩C))\n");  
 printf("----------------------------------------------\n");  
 int equivalent = 1; // Флаг, указывающий на тождественность выражений (предполагаем, что они тождественны)  
 for (int A = 0; A <= 1; A++) {  
 for (int B = 0; B <= 1; B++) {  
 for (int C = 0; C <= 1; C++) {  
 int result1 = expression1(A, B, C);  
 int result2 = expression2(A, B, C);  
 printf("%d %d %d | %d | %d\n", A, B, C, result1, result2);  
 if (result1 != result2) {  
 equivalent = 0; // Если значения выражений не совпадают, выражения не являются тождественными  
 }  
 }  
 }  
 }  
  
 if (equivalent) {  
 printf("\nВыражения являются тождественно эквивалентными.\n");  
 } else {  
 printf("\nВыражения не являются тождественно эквивалентными.\n");  
 }  
  
 return 0;  
}

Результат выполнения:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Исходя из таблицы, можно видеть, что значения обоих выражений совпадают для всех комбинаций входных значений A, B и C, кроме случаев, когда A=1, B=0 и C=0, а также A=1, B=1 и C=0. В этих двух случаях значения выражений отличаются.

Следовательно, выражения ((A∩C)∪(B∩C) - (B∩A)∩C) и (((A∪B)∩(A∪C)) - (A∩B∩C)) не являются тождественно эквивалентными, так как они дают разные результаты для некоторых комбинаций входных значений.

**8. Доказать тождественность выражений 3 и 4 теоретико-множественным методом. Для автоматизации доказательства написать программу, в которой вычисляются и сравниваются значения выражений 3 и 4 при А={1,3,5,7}, B={2,3,6,7} и C={4,5,6,7}.**

Так как универсум не был задан, он будет равен объединением всех множеств. U = A∪B∪C = {1,3,5,7}∪{2,3,6,7}∪{4,5,6,7} = {1,2,3,4,5,6,7}

Рассмотрим, совпадают ли множества, заданные выражениями (A∪(B∩C)- (B∩A)∩C) и (((A∪B)∩(A∪C)) - (A∩B∩C)), используя теоретико-множественный подход.

Для начала, давайте вычислим каждое выражение поэлементно и проверим равенство полученных множеств.

Выражение (A∪(B∩C)- (B∩A)∩C):

1. Вычисляем B∩C = {6, 7}

2. Вычисляем A∪(B∩C) = {1, 3, 5, 6, 7}

3. Вычисляем (B∩A) = {3, 7}

4. Вычисляем (B∩A)∩C = {7}

5. Вычисляем A∪(B∩C)- (B∩A)∩C = {1, 3, 5, 6}

Выражение (((A∪B)∩(A∪C)) - (A∩B∩C)):

1. Вычисляем A∪B = {1, 2, 3, 5, 6, 7}

2. Вычисляем A∪C = {1, 3, 4, 5, 6, 7}

3. Вычисляем (A∪B)∩(A∪C) = {1, 3, 5, 6, 7}

4. Вычисляем A∩B∩C = {7}

5. Вычисляем ((A∪B)∩(A∪C)) - (A∩B∩C) = {1, 3, 5, 6}

После вычисления каждого выражения, мы получили одинаковые множества {1, 3, 5, 6}. Таким образом, можно заключить, что выражения (A∪(B∩C)- (B∩A)∩C) и (((A∪B)∩(A∪C)) - (A∩B∩C)) тождественны и представляют одно и то же множество для данных значений A, B и C.

Программная реализация:

#include <stdio.h>  
  
// Функция для проверки, содержится ли элемент в множестве  
int contains(int\* set, int size, int element) {  
 for (int i = 0; i < size; i++) {  
 if (set[i] == element) {  
 return 1;  
 }  
 }  
 return 0;  
}  
  
// Функция для добавления элемента в множество  
void add(int\* set, int\* size, int element) {  
 if (!contains(set, \*size, element)) {  
 set[\*size] = element;  
 (\*size)++;  
 }  
}  
  
// Функция для удаления элемента из множества  
void removeElement(int\* set, int\* size, int element) {  
 int index = -1;  
 for (int i = 0; i < \*size; i++) {  
 if (set[i] == element) {  
 index = i;  
 break;  
 }  
 }  
 if (index != -1) {  
 for (int i = index; i < \*size - 1; i++) {  
 set[i] = set[i + 1];  
 }  
 (\*size)--;  
 }  
}  
  
// Функция для вычисления выражения A∪(B∩C) - (B∩A)∩C  
void expression1() {  
 int A[] = {1, 3, 5, 7};  
 int B[] = {2, 3, 6, 7};  
 int C[] = {4, 5, 6, 7};  
 int result[100];  
 int resultSize = 0;  
  
 // A∪(B∩C)  
 for (int i = 0; i < sizeof(A) / sizeof(A[0]); i++) {  
 add(result, &resultSize, A[i]);  
 }  
 for (int i = 0; i < sizeof(B) / sizeof(B[0]); i++) {  
 if (contains(C, sizeof(C) / sizeof(C[0]), B[i])) {  
 add(result, &resultSize, B[i]);  
 }  
 }  
  
 // (B∩A)∩C  
 for (int i = 0; i < sizeof(B) / sizeof(B[0]); i++) {  
 if (contains(A, sizeof(A) / sizeof(A[0]), B[i]) && contains(C, sizeof(C) / sizeof(C[0]), B[i])) {  
 removeElement(result, &resultSize, B[i]);  
 }  
 }  
  
 printf("Result 1: ");  
 for (int i = 0; i < resultSize; i++) {  
 printf("%d ", result[i]);  
 }  
 printf("\n");  
}  
  
// Функция для вычисления выражения (((A∪B)∩(A∪C)) - (A∩B∩C))  
void expression2() {  
 int A[] = {1, 3, 5, 7};  
 int B[] = {2, 3, 6, 7};  
 int C[] = {4, 5, 6, 7};  
 int result[100];  
 int resultSize = 0;  
  
 // (A∪B)∩(A∪C)  
 int temp1[100];  
 int temp1Size = 0;  
 for (int i = 0; i < sizeof(A) / sizeof(A[0]); i++) {  
 add(temp1, &temp1Size, A[i]);  
 }  
 for (int i = 0; i < sizeof(B) / sizeof(B[0]); i++) {  
 add(temp1, &temp1Size, B[i]);  
 }  
  
 int temp2[100];  
 int temp2Size = 0;  
 for (int i = 0; i < sizeof(A) / sizeof(A[0]); i++) {  
 add(temp2, &temp2Size, A[i]);  
 }  
 for (int i = 0; i < sizeof(C) / sizeof(C[0]); i++) {  
 add(temp2, &temp2Size, C[i]);  
 }  
  
 for (int i = 0; i < temp1Size; i++) {  
 if (contains(temp2, temp2Size, temp1[i])) {  
 add(result, &resultSize, temp1[i]);  
 }  
 }  
  
 // (A∩B∩C)  
 int temp3[100];  
 int temp3Size = 0;  
 for (int i = 0; i < sizeof(A) / sizeof(A[0]); i++) {  
 if (contains(B, sizeof(B) / sizeof(B[0]), A[i]) && contains(C, sizeof(C) / sizeof(C[0]), A[i])) {  
 add(temp3, &temp3Size, A[i]);  
 }  
 }  
  
 for (int i = 0; i < temp3Size; i++) {  
 removeElement(result, &resultSize, temp3[i]);  
 }  
  
 printf("Result 2: ");  
 for (int i = 0; i < resultSize; i++) {  
 printf("%d ", result[i]);  
 }  
 printf("\n");  
}  
  
int main() {  
 expression1();  
 expression2();  
  
 return 0;  
}

Вывод программы

Изображение выглядит как Шрифт, текст, типография, рукописный текст

Автоматически созданное описание

Вывод: на этой лабораторной работе я изучил методы доказательства теоретико-множественных тождеств.