

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)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №1.3

по дисциплине: Дискретная математика
тема: «Теоретико-множественные тождества»

Выполнил: ст. группы ПВ-223
Пахомов Владислав Андреевич

Проверили:
ст. пр. Рязанов Юрий Дмитриевич
ст. пр. Бондаренко Татьяна Владими-
ровна

Белгород 2023 г.

Лабораторная работа №1.3

Теоретико-множественные тождества

Вариант 10

Цель работы: изучить методы доказательства теоретико-множественных тождеств.

№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 (в соответствии с вариантом №10, необходимо заштриховать области 2, 3, 4).

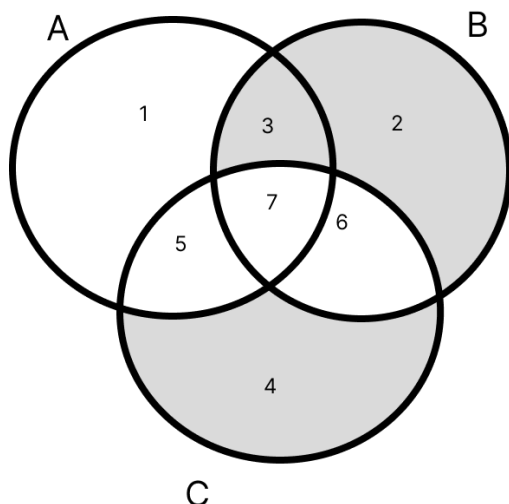


Рис. 1: Круги Эйлера, соответствующие множествам А, В и С с пронумерованными элементарными областями

№2. Написать выражение 1 над множествами А, В и С, определяющее заштрихованную область, используя операции пересечения, объединения и дополнения.

$$F = \bar{A} \cap B \cap \bar{C} \cup A \cap B \cap \bar{C} \cup \bar{A} \cap \bar{B} \cap C \quad (1)$$

№3. Используя свойства операций над множествами, преобразовать выражение 1 в выражение 2, не содержащее операции дополнения множества.

$$F = \bar{A} \cap B \cap \bar{C} \cup A \cap B \cap \bar{C} \cup \bar{A} \cap \bar{B} \cap C = B \cap \bar{C} \cup \bar{A} \cap \bar{B} \cap C = (B - C) \cup (C - \overline{\bar{A} \cap \bar{B}}) = (B - C) \cup (C - (A \cup B))$$

$$F = (B - C) \cup (C - (A \cup B)) \quad (2)$$

№4. Используя свойства операций над множествами, преобразовать выражение 2 в выражение 3, не содержащее операции объединения множеств.

$$F = (B - C) \cup (C - (A \cup B)) = (B - C) \cup (C - A) \cap (C - B) = ((B - C) \cup (C - B)) \cap ((B - C) \cup (C - A)) = (B \Delta C) \cap (((B - C) \cup (C - A)) \Delta (C - A))$$

$$F = (B \Delta C) \cap (((B - C) - (C - A)) \Delta (C - A)) \quad (3)$$

№5. Используя свойства операций над множествами, преобразовать выражение 3 в выражение 4, не содержащее операции пересечения множеств.

$$F = (B \Delta C) \cap (((B - C) - (C - A)) \Delta (C - A)) = (B \Delta C) - ((B \Delta C) - (((B - C) - (C - A)) \Delta (C - A)))$$

$$F = (B \Delta C) - ((B \Delta C) - (((B - C) - (C - A)) \Delta (C - A))) \quad (4)$$

№6. Доказать тождественность выражений 2 и 3 методом характеристических функций.

Используем метод арифметических характеристических функций

$$(B - C) \cup (C - (A \cup B)) = (B \Delta C) - ((B \Delta C) - (((B - C) - (C - A)) \Delta (C - A)))$$

В дальнейших вычислениях для упрощения введём обозначения:

$$a = X_A(x)$$

$$b = X_B(x)$$

$$c = X_C(x)$$

Преобразуем левую часть выражения

$$\begin{aligned} X_{(B-C) \cup (C-(A \cup B))}(x) &= X_{B-C}(x) + X_{C-(A \cup B)}(x) - X_{B-C}(x) \cdot X_{C-(A \cup B)}(x) = (X_B(x) - \\ &X_B(x) \cdot X_C(x)) + (X_C(x) - X_C(x) \cdot X_{A \cup B}(x)) - (X_B(x) - X_B(x) \cdot X_C(x)) \cdot (X_C(x) - \\ &X_C(x) \cdot X_{A \cup B}(x)) = (X_B(x) - X_B(x) \cdot X_C(x)) + (X_C(x) - X_C(x) \cdot (X_A(x) + X_B(x) - \\ &X_A(x) \cdot X_B(x))) - (X_B(x) - X_B(x) \cdot X_C(x)) \cdot (X_C(x) - X_C(x) \cdot (X_A(x) + X_B(x) - \\ &X_A(x) \cdot X_B(x))) = (b - bc) + (c - c(a + b - ab)) - (b - bc)(c - c(a + b - ab)) = \\ &b + c + bc^2 + cb^2 - ac - b^2c^2 - 3bc + ab^2c^2 - abc^2 - acb^2 + 2abc = b + c + bc + cb - \\ &ac - bc - 3bc + abc - abc - acb + 2abc = b + c - ac - 2bc + abc = X_B(x) + X_C(x) - \\ &X_A(x) \cdot X_C(x) - 2 \cdot X_B(x) \cdot X_C(x) + X_A(x) \cdot X_B(x) \cdot X_C(x) \end{aligned}$$

Преобразуем правую часть выражения

$$\begin{aligned} X_{(B \Delta C) \cap (((B-C)-(C-A)) \Delta (C-A))}(x) &= X_{B \Delta C}(x) \cdot X_{((B-C)-(C-A)) \Delta (C-A)}(x) = (X_B(x) + \\ &X_C(x) - 2 \cdot X_B(x) \cdot X_C(x)) \cdot (X_{(B-C)-(C-A)}(x) + X_{C-A}(x) - 2 \cdot X_{(B-C)-(C-A)}(x) \cdot \\ &X_{C-A}(x)) = (X_B(x) + X_C(x) - 2 \cdot X_B(x) \cdot X_C(x)) \cdot (X_{B-C}(x) - X_{B-C}(x) \cdot X_{C-A} + \\ &X_C(x) - X_C(x) \cdot X_A(x) - 2 \cdot (X_{B-C}(x) - X_{B-C}(x) \cdot X_{C-A}) \cdot (X_C(x) - X_C(x) \cdot X_A(x))) = \\ &(X_B(x) + X_C(x) - 2 \cdot X_B(x) \cdot X_C(x)) \cdot (X_B(x) - X_B(x) \cdot X_C(x) - (X_B(x) - X_B(x) \cdot \\ &X_C(x)) \cdot (X_C(x) - X_C(x) \cdot X_A(x)) + X_C(x) - X_C(x) \cdot X_A(x) - 2 \cdot (X_B(x) - X_B(x) \cdot \\ &X_C(x) - (X_B(x) - X_B(x) \cdot X_C(x)) \cdot (X_C(x) - X_C(x) \cdot X_A(x))) \cdot (X_C(x) - X_C(x) \cdot \\ &X_A(x))) = (b + c - 2bc)(b - bc - (b - bc)(c - ca) + c - ca - 2(b - bc - (b - bc)(c - \\ &ca))(c - ca)) = (b + c - 2bc)(b - bc + c - ca) = b - bc + bc - abc + bc - bc + c - \\ &ac - 2bc + 2bc - 2bc + 2abc = b + c - ac + abc - 2bc = X_B(x) + X_C(x) - X_A(x) \cdot \\ &X_C(x) - 2 \cdot X_B(x) \cdot X_C(x) + X_A(x) \cdot X_B(x) \cdot X_C(x) \end{aligned}$$

АХФ совпали, значит F_2 и F_3 тождественны

№7. Доказать тождественность выражений 2 и 4 методом логических функций. Для автоматизации доказательства написать программу, которая получает и сравнивает таблицы истинности логических функций.

$$(B - C) \cup (C - (A \cup B)) = (B \Delta C) \cap (((B - C) - (C - A)) \Delta (C - A))$$

Преобразуем левую часть выражения

$$\lambda_{(B-C) \cup (C-(A \cup B))}(x) = (\lambda_B(x) \wedge \overline{\lambda_C(x)}) \vee (\lambda_C(x) \wedge \overline{\lambda_A(x) \vee \lambda_B(x)})$$

Преобразуем правую часть выражения

$$\lambda_{(B \Delta C) \cap (((B-C)-(C-A)) \Delta (C-A))}(x) = (\lambda_B(x) \oplus \lambda_C(x)) \wedge (((\lambda_B(x) \wedge \overline{\lambda_C(x)}) \wedge \lambda_C(x) \wedge \overline{\lambda_A(x)}) \oplus (\lambda_C(x) \wedge \overline{\lambda_A(x)}))$$

Получили ЛХФ, на их основе составили две функции на языке программирования C++, функция F2 соответствует функции F_2 , функция F4 соответствует функции F_4 . Параметры a, b, c в обеих функциях соответственно равны $\lambda_A(x), \lambda_B(x), \lambda_C(x)$. Операторам $\wedge, \vee, \oplus, \overline{}$ в формуле в программе соответствуют логическое "И"(&&), логическое "ИЛИ"(|), исключающее "ИЛИ"(^) и "НЕ"(!)

Полный код программы:

```
#include <iostream>

bool F2(bool a, bool b, bool c) {
    return (b && !c) || (c && !(a || b));
}

bool F4(bool a, bool b, bool c) {
    return (b ^ c) && (((b && !c) && !(c && !a)) ^ (c && !a));
}

int main() {
    std::cout << "A B C F2 F4" << std::endl;
    for (int bA = 0; bA <= 1; bA++)
        for (int bB = 0; bB <= 1; bB++)
            for (int bC = 0; bC <= 1; bC++) {
                std::cout << bA << " " << bB << " " << bC << " " << F2(bA, bB, bC) << " "
                << F4(bA, bB, bC) << std::endl;
            }
    return 0;
}
```

Результат выполнения программы:

```
/Users/vlad/Desktop/C/discrete_math/cmake-build-debug/bin/lab3_task7
```

```
A B C F2 F4
```

```
0 0 0 0 0
```

```
0 0 1 1 1
```

```
0 1 0 1 1
```

```
0 1 1 0 0
```

```
1 0 0 0 0
```

```
1 0 1 0 0
```

```
1 1 0 1 1
```

```
1 1 1 0 0
```

```
Process finished with exit code 0
```

Таблицы истинности совпали, следовательно F_2 и F_4 тождественны.

- №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 $A=\{1,3,5,7\}$, $B=\{2,3,6,7\}$ и $C=\{4,5,6,7\}$. Используя функции, полученные в л.р. № 1.1 получим класс Sett:

```
class Sett {
public:
    std::vector<int> elements;

    Sett(std::vector<int> elms) {
        elements = elms;
        std::sort(elements.begin(), elements.end());
    }

    ~Sett() {}

    Sett operator*(Sett anotherSet) {
        std::vector<int> arrayC(0, 0);
        int arrayASize = elements.size();
        int arrayBSize = anotherSet.elements.size();
        size_t i = 0, j = 0;

        while (i < arrayASize && j < arrayBSize)
            if (elements[i] < anotherSet.elements[j])
                i++;
            else if (elements[i] > anotherSet.elements[j])
                j++;
            else {
                arrayC.push_back(elements[i]);
                i++;
                j++;
            }

        return Sett(arrayC);
    }

    Sett operator-(Sett anotherSet) {
```

```

        std::vector<int> arrayC(0, 0);
        int arrayASize = elements.size();
        int arrayBSize = anotherSet.elements.size();
        size_t i = 0, j = 0;

        while (i < arrayASize && j < arrayBSize)
            if (elements[i] < anotherSet.elements[j])
                arrayC.push_back(elements[i++]);
            else if (elements[i] > anotherSet.elements[j])
                j++;
            else {
                i++;
                j++;
            }

        while (i < arrayASize)
            arrayC.push_back(elements[i++]);

        return Sett(arrayC);
    }

    Sett operator+(Sett anotherSet) {
        std::vector<int> arrayC(0, 0);
        int arrayASize = elements.size();
        int arrayBSize = anotherSet.elements.size();
        size_t i = 0, j = 0;

        while (i < arrayASize && j < arrayBSize)
            if (elements[i] < anotherSet.elements[j])
                arrayC.push_back(elements[i++]);
            else if (elements[i] > anotherSet.elements[j])
                arrayC.push_back(anotherSet.elements[j++]);
            else {
                arrayC.push_back(elements[i]);
                i++;
                j++;
            }

        while (i < arrayASize)
            arrayC.push_back(elements[i++]);

        while (j < arrayBSize)
            arrayC.push_back(anotherSet.elements[j++]);

        return Sett(arrayC);
    }

    Sett non(Sett universum) {
        std::vector<int> arrayC(0, 0);
        int arraySize = elements.size();
        int universumSize = universum.elements.size();
        size_t i = 0, j = 0;
        // Проверяем, что универсум действительно универсум
        assert(elements[arraySize - 1] <= universum.elements[universumSize - 1]);
    }

```

```

        while (i < universumSize && j < arraySize) {
            if (universum.elements[i] < elements[j])
                arrayC.push_back(universum.elements[i++]);
            else if (universum.elements[i] == elements[j]) {
                i++;
                j++;
                // вторым его отличием будет то, что если элемент есть в A
→ и его нет в universum, программа будет падать
            } else
                assert(elements[j] >= universum.elements[i]);
        }

        while (i < universumSize)
            arrayC.push_back(universum.elements[i++]);

        return Sett(arrayC);
    }

Sett operator^(Sett anotherSet) {
    std::vector<int> arrayC(0, 0);
    int arrayASize = elements.size();
    int arrayBSize = anotherSet.elements.size();
    size_t i = 0, j = 0;

    while (i < arrayASize && j < arrayBSize)
        if (elements[i] < anotherSet.elements[j])
            arrayC.push_back(elements[i++]);
        else if (elements[i] > anotherSet.elements[j])
            arrayC.push_back(anotherSet.elements[j++]);
        else {
            j++;
            i++;
        }

    while (i < arrayASize)
        arrayC.push_back(elements[i++]);

    while (j < arrayBSize)
        arrayC.push_back(anotherSet.elements[j++]);

    return Sett(arrayC);
}

void print() {
    for (int i = 0; i < elements.size(); i++) {
        std::cout << elements[i] << " ";
    }

    std::cout << std::endl;
}

};

```

В котором переопределены следующие операторы и находятся следующие методы:

\wedge – симметрическая разность

$*$ - пересечение

$+$ - объединение

\cup – дополнение

$-$ - разность

print – метод, выводящий множество

Итоговая программа выглядит так:

```
int main() {
    Sett A(std::vector<int>({1, 3, 5, 7}));
    Sett B(std::vector<int>({2, 3, 6, 7}));
    Sett C(std::vector<int>({4, 5, 6, 7}));
    Sett U(std::vector<int>({1, 2, 3, 4, 5, 6, 7}));

    Sett F3 = ((B - C) ^ (C - ((A ^ B) ^ (A * B)))) ^ ((B - C) * (C - ((A ^ B) ^ (A *
→ B))));
    Sett F4 = ((B - C) ^ (C - ((A ^ B) ^ (A - (A - B))))) ^ ((B - C) - ((B - C) - (C -
→ ((A ^ B) ^ (A - (A - B))))));

    F3.print();
    F4.print();

    return 0;
}
```

Результат выполнения программы:

```
/Users/vlad/Desktop/C/discrete_math/cmake-build-debug/bin/lab3_task8
```

```
2 3 4
```

```
2 3 4
```

```
Process finished with exit code 0
```

Результаты вычисления выражений равны, следовательно F_3 и F_4 тождественны.

Вывод: в ходе лабораторной работы изучили методы доказательства теоретико-множественных тождеств.