

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 №1.4**

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 №1.4**  
Теоретико-множественные тождества  
Вариант 10

**Цель работы:** научиться решать теоретико-множественные уравнения с применением ЭВМ.

№1. Преобразовать исходное уравнение  $A \cap (B \cup X) \Delta C = A - (B \cap \overline{X}) \Delta C \cap X$  в уравнение с пустой правой частью.

$$(A \cap (B \cup X) \Delta C) \Delta (A - (B \cap \overline{X}) \Delta C \cap X) = \emptyset$$

№2. Преобразовать левую часть уравнения к виду  $\overline{X} \cap \varphi^\emptyset \cup X \cap \varphi^U$ , используя разложение Шеннона по неизвестному множеству X.

$$\begin{aligned}\varphi^\emptyset &= (A \cap (B \cup \emptyset) \Delta C) \Delta (A - (B \cap \overline{\emptyset}) \Delta C \cap \emptyset) \\ \varphi^U &= (A \cap (B \cup U) \Delta C) \Delta (A - (B \cap \overline{U}) \Delta C \cap U) \\ \overline{X} \cap ((A \cap (B \cup \emptyset) \Delta C) \Delta (A - (B \cap \overline{\emptyset}) \Delta C \cap \emptyset)) \cup X \cap ((A \cap (B \cup U) \Delta C) \Delta (A - (B \cap \overline{U}) \Delta C \cap U)) &= \emptyset\end{aligned}$$

№3. Написать программу, вычисляющую значения множеств  $\varphi^\emptyset$  и  $\varphi^U$  при заданных исходных множествах.  $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ ,  $A = \{4, 5, 7, 8, 9, 10\}$ ,  $B = \{2, 3, 4, 5, 6, 10\}$ ,  $C = \{4, 5, 7, 8, 10\}$

Для вычисления значений воспользуемся классом, полученным в л.р. №3, внося в него некоторые правки

```
#include <vector>
#include <iostream>

template<class T>
class Sett {
public:
    std::vector<T> elements;

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

    ~Sett() {}

    Sett<T> operator*(Sett<T> anotherSet) {
        std::vector<T> 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 Set<T>(arrayC);
    }

Set<T> operator-(Set<T> anotherSet) {
    std::vector<T> 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 Set<T>(arrayC);
}

Set<T> operator+(Set<T> 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 Set<T>(arrayC);
    }

    Set<T> non(Set<T> universum) {
        std::vector<T> arrayC(0, 0);
        int arraySize = elements.size();
        int universumSize = universum.elements.size();
        size_t i = 0, j = 0;
        // Проверяем, что универсум действительно универсум
        assert(arraySize == 0 || 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 Set<T>(arrayC);
    }

    Set<T> operator^(Set<T> anotherSet) {
        std::vector<T> 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 Set<T>(arrayC);
    }

    bool operator==(Set<T> other) {
        return elements == other.elements;
    }

    bool operator!=(Set<T> other) {
        return !(*this == other);
    }

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

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

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

        out << std::endl;
    }
};

```

Получили программу:

```

#include "../libs/alg/alg.h"

int main() {
    Set<int> nothing(std::vector<int>({}));
    Set<int> U(std::vector<int>({1, 2, 3, 4, 5, 6, 7, 8, 9, 10}));
    Set<int> A(std::vector<int>({4, 5, 7, 8, 9, 10}));
    Set<int> B(std::vector<int>({2, 3, 4, 5, 6, 10}));
    Set<int> C(std::vector<int>({4, 5, 7, 8, 10}));

    Set<int> phiNothing = ((A * (B + nothing)) ^ C) ^ (A - ((B * nothing.non(U)) ^ (C
→ * nothing)));
    Set<int> nonPhiU = (((A * (B + U)) ^ C) ^ (A - ((B * U.non(U)) ^ (C *
→ U)))) .non(U);

    phiNothing.print();
    nonPhiU.print();
}

```

- №4. Вычислить значения множеств  $\varphi^\emptyset$  и  $\overline{\varphi^U}$  и сделать вывод о существовании решения уравнения. Если решения уравнения не существует, то выполнить п.п. 1—4 для следующего (предыдущего) варианта.

Результат выполнения программы:

```
/Users/vlad/Desktop/C/discrete_math/cmake-build-debug/bin/lab4_task3
```

```
9
```

```
1 2 3 4 5 6 7 8 9 10
```

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

Множество  $\varphi^\emptyset = \{9\}$  является подмножеством множества  $\overline{\varphi^U} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ , следовательно, уравнение имеет решения.

- №5. Определить мощность общего решения, найти некоторые (или все) частные решения, в том числе частные решения наименьшей и наибольшей мощности.

$$\overline{\varphi^U} - \varphi^\emptyset = \{1, 2, 3, 4, 5, 6, 7, 8, 10\}$$

Количество возможных подмножеств слишком велико, мощность общего решения будет равна количеству подмножеств множества  $\overline{\varphi^U} - \varphi^\emptyset$ , вычислим его по формуле  $2^{|\overline{\varphi^U} - \varphi^\emptyset|}$

$$2^{|\overline{\varphi^U} - \varphi^\emptyset|} = 2^9 = 512$$

Приведём 8 возможных решений уравнения:

$X = \{9\}$  - решение наименьшей мощности

$X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$  - решение наибольшей мощности

$X = \{1, 5, 6, 9\}$

$X = \{1, 2, 3, 4, 5, 9, 10\}$

$X = \{7, 8, 9, 10\}$

$X = \{1, 2, 5, 6, 7, 9\}$

$X = \{2, 3, 7, 8, 9, 10\}$

$X = \{3, 4, 5, 6, 8, 9, 10\}$

- №6. Написать программу для проверки найденных решений.

```
#include "../libs/alg/alg.h"

Sett<int> U(std::vector<int>({1, 2, 3, 4, 5, 6, 7, 8, 9, 10}));
Sett<int> A(std::vector<int>({4, 5, 7, 8, 9, 10}));
Sett<int> B(std::vector<int>({2, 3, 4, 5, 6, 10}));
Sett<int> C(std::vector<int>({4, 5, 7, 8, 10}));

// Функция для вычисления левой части уравнения
template<typename T>
Sett<T> left(Sett<T> X) {
    return A * (B + X) ^ C;
}

// Функция для вычисления правой части уравнения
template<typename T>
```

```

Sett<T> right(Sett<T> X) {
    return A - (B * X.non(U)) ^ (C * X);
}

int main() {
    // Решения уравнения
    std::vector<Sett<int>> decisions = {
        Sett<int>({9}),
        Sett<int>({1, 2, 3, 4, 5, 6, 7, 8, 9, 10}),
        Sett<int>({1, 5, 6, 9}),
        Sett<int>({1, 2, 3, 4, 5, 9, 10}),
        Sett<int>({7, 8, 9, 10}),
        Sett<int>({1, 2, 5, 6, 7, 9}),
        Sett<int>({2, 3, 7, 8, 9, 10}),
        Sett<int>({3, 4, 5, 6, 8, 9, 10})
    };

    for (auto &x: decisions)
        // Проверяем, что левая и правая части уравнений равны
        if (left(x) != right(x)) {
            std::cerr << "Множество" << std::endl;
            x.print(std::cerr);
            std::cerr << "не является решением уравнения" << std::endl;
            return 1;
        }

    std::cout << "Решения подходят для уравнения" << std::endl;

    return 0;
}

```

Результат выполнения программы:

`/Users/vlad/Desktop/C/discrete_math/cmake-build-debug/bin/lab4_task6`

Решения подходят для уравнения

`Process finished with exit code 0`

**Вывод:** в ходе лабораторной работы научились решать теоретико-множественные уравнения с применением ЭВМ.