

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ
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
дисциплина: Дискретная математика
тема: «Теоретико-множественные уравнения»

Выполнил: ст. группы ПВ-201
Машуров Дмитрий Русланович
Проверил: Бондаренко Т.В.

Белгород 2021

Лабораторная работа №1.4

Теоретико-множественные уравнения

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

Задания

1. Преобразовать исходное уравнение (см. —Варианты заданий) в уравнение с пустой правой частью.
2. Преобразовать левую часть уравнения к виду $\bar{X} \cap \varnothing \cup X \cap \varnothing^U$ используя разложение Шеннона по неизвестному множеству X .
3. Написать программу, вычисляющую значения множеств \varnothing и $U \varnothing$ при заданных исходных множествах.
4. Вычислить значения множеств \varnothing и $U \varnothing$ 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

Задание варианта №17

$$(C - X) \triangle \overline{A - \bar{X}} = (A - X) \triangle B \cap X$$

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

$$A = \{3, 7, 9, 10\}$$

$$B = \{1, 3, 8, 9, 10\}$$

$$C = \{2, 4, 5, 6, 7\}$$

$$X = ?$$

Выполнение:

1. Преобразую уравнение своего варианта к уравнению с пустой правой частью:

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

2. Выполню разложение Шеннона левой части уравнения по неизвестному множеству X :

$$\bar{X} \cap \varphi^\emptyset \cup X \cap \varphi^U = \emptyset, \text{ где}$$

$$\varphi^\emptyset = (C - \emptyset) \Delta \overline{A - \bar{\emptyset}} \Delta (A - \emptyset) \Delta B \cap \emptyset,$$

$$\varphi^U = (C - U) \Delta \overline{A - \bar{U}} \Delta (A - U) \Delta B \cap U$$

Преобразую их

$$\varphi^\emptyset = (C - \emptyset) \Delta \overline{A - \bar{\emptyset}} \Delta (A - \emptyset) \Delta B \cap \emptyset = C \Delta U \Delta A \Delta \emptyset = \bar{C} \Delta A$$

$$\varphi^U = (C - U) \Delta \overline{A - \bar{U}} \Delta (A - U) \Delta B \cap U = \emptyset \Delta \bar{A} \Delta \emptyset \Delta B = \bar{A} \Delta B$$

3. Программа

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define MIN(a, b) (a < b ? a : b)
#define MAX(a, b) (a > b ? a : b)

struct set {
    int *values;
    int min;
    int max;
};

int is_elem_in(struct set *s, int elem) {
    int *a = s->values;

    if (a[elem] == 1) {
        return 1;
    }

    return 0;
}

struct set create_set(int min, int max) {
    int n = max - min,
        *a = (int *) calloc(sizeof(int), n);
    struct set temp;

    temp.max = max;
    temp.min = min;

    temp.values = a;

    return temp;
}
```

```

void delete_set(struct set *s) {
    free(s->values);
    free(s);
}

struct set complemt_set(struct set *s) {
    int *a = s->values;
    struct set temp = create_set(s->min, s->max);
    int *b = temp.values;

    for (int i = s->min; i <= s->max; i++) {
        b[i] = (a[i] == 0 ? 1 : 0);
    }

    return temp;
}

void output_set(struct set *s) {
    int *values = s->values;

    printf("{ ");

    for (int i = s->min; i <= s->max - 1; i++) {
        if (values[i] == 1)
            printf("%d ", i);
    }
    if (values[s->max] == 1)
        printf("%d}\n", s->max);
    else
        printf("}");
}

void add(struct set *s, int value) {
    int *values = s->values;
    values[value] = 1;
}

void delete(struct set *s, int value) {
    int *values = s->values;

    if (values[value] == 1) {
        values[value] = 0;
    }
}

struct set union_set(struct set *s1, struct set *s2) {
    int new_min = MIN(s1->min, s2->min),
        new_max = MAX(s1->max, s2->max),
        *a = s1->values,
        *b = s2->values;

    struct set temp = create_set(new_min, new_max);
    int *t = temp.values;

    for (int i = s1->min; i < s1->max; i++) {
        if (t[i] == 0) t[i] = a[i];
    }

    for (int i = s2->min; i < s2->max; i++) {
        if (t[i] == 0) t[i] = b[i];
    }

    temp.max = new_max;
}

```

```

    temp.min = new_min;
    return temp;
}

struct set intersection_set(struct set *s1, struct set *s2) {
    int new_min = MAX(s1->min, s2->min),
        new_max = MIN(s1->max, s2->max),
        *a = s1->values, *b = s2->values;
    struct set temp = create_set(new_min, new_max);
    int *t = temp.values;

    for (int i = new_min; i < new_max; i++) {
        if ((a[i] == 1) && (b[i] == 1))
            t[i] = 1;
    }

    temp.max = new_max;
    temp.min = new_min;
    return temp;
}

int is_subset(struct set *s1, struct set *s2) {
    int *a = s2->values;

    for (int i = s2->min; i < s2->max; i++) {
        if (a[i] == 1) {
            if (!is_elem_in(s1, i)) {
                return 0;
            }
        }
    }

    return 1;
}

struct set subtraction_set(struct set *s1, struct set *s2) {
    int *a = s1->values,
        *b = s2->values;
    struct set temp = create_set(s1->min, s1->max);
    int *t = temp.values;

    for (int i = s1->min; i < s2->max; i++) {
        if (s2->min <= i && i <= s2->max) {
            if (a[i] == 1 && b[i] == 1) {
                t[i] = 0;
            } else t[i] = a[i];
        } else t[i] = a[i];
    }

    temp.min = s1->min;
    temp.max = s1->max;
    return temp;
}

struct set sym_subtraction(struct set *s1, struct set *s2) {
    struct set t1, t2, temp;

    t1 = subtraction_set(s1, s2);
    t2 = subtraction_set(s2, s1);
    temp = union_set(&t1, &t2);

    delete_set(&t1);

```

```

        delete_set(&t2);

        return temp;
    }

    struct set get_set_from_array(int min, int max, const int *a, int n) {
        struct set temp = create_set(min, max);
        int *t = temp.values;

        for (int i = 0; i < n; ++i) {
            t[a[i]] = 1;
        }

        temp.max = max;
        temp.min = min;
        return temp;
    }

    void print_set_of_solves(struct set *s, struct set *emp) {
        int *a = s->values, r;

        for (int i = 0; i < 10; ++i) {
            if (a[i] == 0) {
                continue;
            } else {
                r = i;

                if (is_subset(s, emp) && is_subset(emp, s)) {
                    output_set(emp);
                } else {
                    add(emp, r);
                    delete(s, r);
                    print_set_of_solves(s, emp);
                }
            }
        }
    }

    int main() {
        struct set t1, t2, t3, t4, t5, t6, t7, te1, res_emp, res_uni,
            u = create_set(1, 10),
            e = create_set(1, 10),
            a = create_set(1, 10),
            b = create_set(1, 10),
            c = create_set(1, 10);

        //init U - universum
        for (int i = 1; i <= 11; ++i) {
            add(&u, i);
        }

        //init A
        add(&a, 3);
        add(&a, 7);
        add(&a, 9);
        add(&a, 10);

        //init B
        add(&b, 1);
        add(&b, 3);
        add(&b, 8);
    }

```

```

add(&b, 9);
add(&b, 10);

//init C
add(&c, 2);
add(&c, 4);
add(&c, 5);
add(&c, 6);
add(&c, 7);

print_set_of_solves(&a, &b);

//expression 1
te1 = complemt_set(&e);

t1 = subtraction_set(&c, &e);

t2 = subtraction_set(&a, &te1);
t3 = complemt_set(&t2);

t4 = subtraction_set(&a, &e);

t5 = intersection_set(&b, &e);

t6 = sym_subtraction(&t1, &t3);
t7 = sym_subtraction(&t4, &t5);
res_emp = sym_subtraction(&t6, &t7);

printf("fi^empty = ");
output_set(&res_emp);

delete_set(&te1);
delete_set(&t1);
delete_set(&t2);
delete_set(&t3);
delete_set(&t4);
delete_set(&t5);
delete_set(&t6);
delete_set(&t7);

printf("\n");

//expression 2
te1 = complemt_set(&u);

t1 = subtraction_set(&c, &u);

t2 = subtraction_set(&a, &te1);
t3 = complemt_set(&t2);

t4 = subtraction_set(&a, &u);

t5 = intersection_set(&b, &u);

t6 = sym_subtraction(&t1, &t3);
t7 = sym_subtraction(&t4, &t5);
res_uni = sym_subtraction(&t6, &t7);
res_uni = complemt_set(&res_uni);

printf("fi^u = ");
output_set(&res_uni);

```

```

delete_set(&te1);
delete_set(&t1);
delete_set(&t2);
delete_set(&t3);
delete_set(&t4);
delete_set(&t5);
delete_set(&t6);
delete_set(&t7);

printf("\n");

//equals

if (is_subset(&res_uni, &res_emp)) {
    t1 = subtraction_set(&res_uni, &res_emp);
    printf("fi^empty is subset of fi^u\n");

    if (is_subset(&e, &t1) && is_subset(&t1, &e)) {

        printf("General solve is {");
        output_set(&res_emp);
        printf("}\n");
        printf("Its power is 1");
    }

} else printf("No any sets");

}

```

4. Вычислю значения множеств $\overline{\varphi^U}$ и φ^\emptyset

$$\varphi^\emptyset = \bar{C} \triangle A = \overline{\{2, 4, 5, 6, 7\}} \triangle \{3, 7, 9, 10\} = \{1, 3, 8, 9, 10\} \triangle \{3, 7, 9, 10\} \\ = \{1, 7, 8\}$$

$$\overline{\varphi^U} = \overline{\bar{A} \triangle B} = \overline{\{3, 7, 9, 10\} \triangle \{1, 3, 8, 9, 10\}} \\ = \overline{\{1, 2, 4, 5, 6, 8\} \triangle \{1, 3, 8, 9, 10\}} = \overline{\{2, 3, 4, 5, 6, 9, 10\}} \\ = \{1, 7, 8\}$$

Множество φ^\emptyset является подмножеством $\overline{\varphi^U}$, следовательно – уравнение имеет решения

5. Определяю мощность общего решения, найду некоторые частные решения

Частное решение – $\{1, 7, 8\}$

Поскольку $\overline{\varphi^U} - \varphi^\emptyset = \emptyset$, то частное решение $\{1, 7, 8\}$ будет является единственным решением данного теоретико-множественного уравнения