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 №3.3

по дисциплине: «Дискретная математика»
по теме: **Фактормножества**

Выполнил/а: ст. группы ПВ-231

Чупахина София Александровна

Проверил: Рязанов Юрий Дмитриевич

Белгород, 2024

Вариант 6

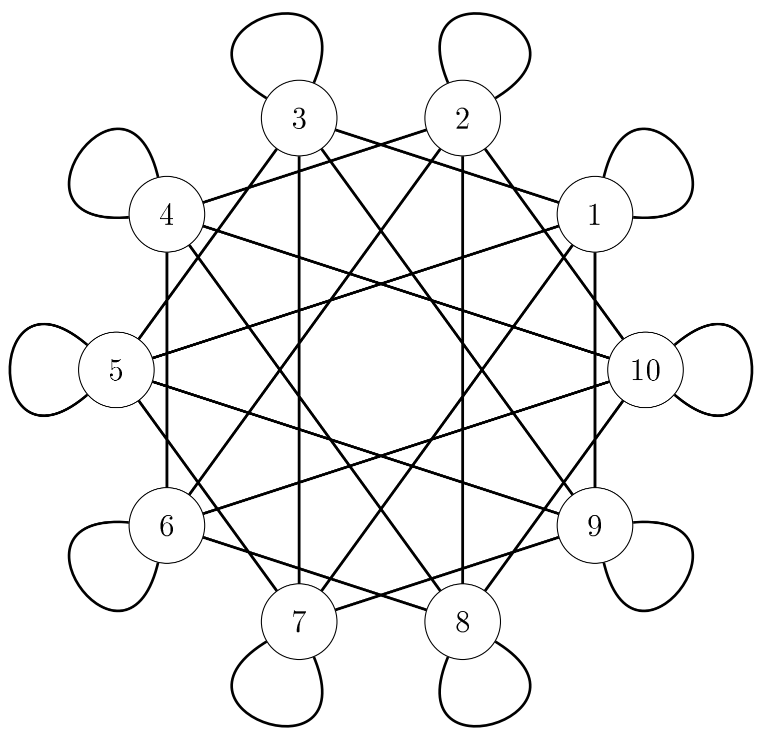
$A = \{(x, y) \big| |x - y| - \text{четно}\}$

Содержание

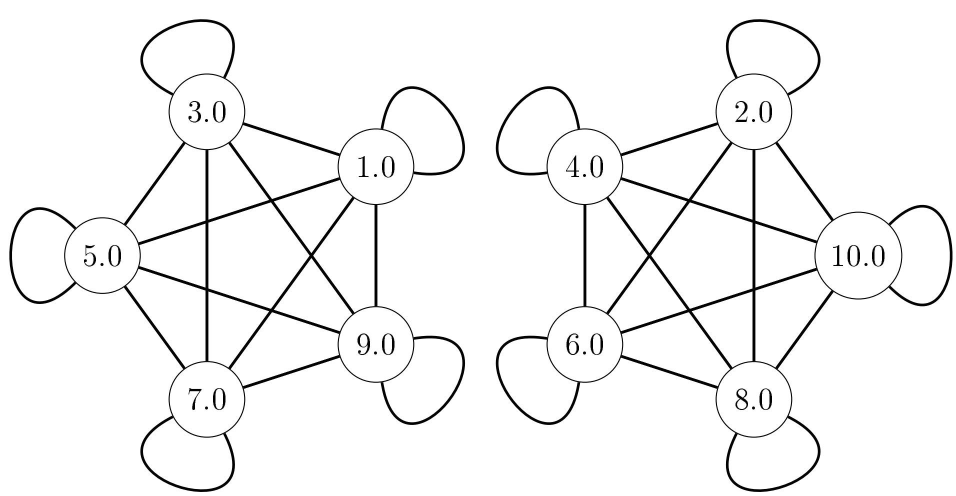
Задание 1

Текст задания: 1. Отношение A на множестве $M = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ задано характеристическим свойством (см. вариант задания). Представить это отношение графом и матрицей.

С помощью графа отношение A можно представить следующим образом. Отдельно можно заметить, что данный граф состоит только из ребер.



Чтобы минимизировать количество пересечений ребер графа, можно разбить его вершины на две группы, не соединенные между собой ни одним ребром. Тогда граф будет выглядеть так:



В виде матрицы отношение A будет представлено следующим образом.

$$\begin{matrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{matrix}$$

Задание 2

Текст задания: Определить свойства отношения. Если отношение не обладает свойством эквивалентности, то выполнить следующий вариант заданий.

Проанализируем основные свойства отношения A.

Оно **является рефлексивным**: в него входят все пары вида (x, x) $(x \in M)$. Это можно отследить по матрице, отображающей это отношение: ее главная диагональ полностью состоит из единиц. На это же указывает и граф отношения: при каждой его вершине присутствует петля. Соответственно, отношение A **не является антирефлексивным**.

Оно **является симметричным**: если в него входит пара вида (x, y), то в него также входит и пара вида (y, x). Это доказывается тем фактом, что симметрична и матрица этого отношения: элементы на позициях, симметричных относительно главной диагонали, одинаковы — а также тем, что граф этого отношения состоит только из ребер. Соответственно, отношение A **не является антисимметричным**, поскольку оно уже симметрично и при этом не пусто. В матрице легко найти пару равных единице элементов с позициями, симметричными относительно главной диагонали (и не лежащих при этом на ней), а в графе легко найти соединенные ребром различные вершины. К примеру, это элементы исходного множества M 1 и 5, образующие принадлежащие отношению пары $(1, 5)$ и $(5, 1)$.

Оно **является транзитивным**: если в него входят пары вида (x, z) и (z, y), то в него также входит и пара вида (x, z). Это доказывается тем фактом, что для каждой присутствующей в графе отношения цепочки из двух ребер есть также ребро, связывающее ее концы (в общем случае, когда граф состоит не только из ребер, учитывается направление цепочки дуг, и «замыкающая» дуга должна идти от ее начала к концу). Соответственно, отношение A **не является антитранзитивным**: оно уже антитранзитивно и при этом не пусто. Это доказывают пары $(1, 7)$ и $(7, 3)$: в графе отношения вершины 1, 7 и 3 соединены ребрами, и концы этой цепочки, вершины 1 и 3 также соединены ребром.

Наконец, оно **не является полным**. Это доказывается тем, что в матрице отношения можно найти пару равных нулю элементов на позициях, симметричных относительно главной диагонали, а в графе — пару не соединенных вершин. Пример, доказывающий его — элементы исходного множества M 1 и 4, образующие не принадлежащие отношению пары $(1, 4)$ и $(4, 1)$.

Сделав выводы об основных свойствах отношения A, можно обозначить его производные свойства. Отношение A **толерантно** (поскольку является рефлексивным и симметричным) и **эквивалентно** (поскольку является рефлексивным, симметричным и транзитивным). Оно **не обладает свойством порядка** (хотя оно и транзитивно, оно не является антисимметричным) и ни одним вытекающим из него свойством (строгого, нестрогого, линейного, строгого линейного и нестрогого линейного порядка).

Эти выводы можно подтвердить, используя функцию для определения и вывода основных и производных свойств отношения, написанную в ходе выполнения 2-ой части лабораторной работы №3.1. Применим эту функцию к отношению A, заданному и заполненному значениями в теле main:

```c
#include "ДИСКРЕТКА../../ ЛАБА 3.1/bin_relations/bin_relations_properties.c"

int main() {
    bin_relation A = bin_relation_createEmpty(10);
    for (int x = 1; x <= 10; x++) {
        for (int y = 1; y <= 10; y++) {
            if (abs(x-y) % 2 == 0) {
                bin_relation_changeValue(&A, x, y, 1);
            }
        }
    }
    bin_relation_print_all_properties(A);
}
```

factor_sets/properties.c

```
        Simple properties of a relation:
This relation is reflexive.
This relation is not antireflexive, because the pair (1, 1) belongs it.
This relation is symmetrical.
This relation is not antisymmetrical, because both pairs (1, 3) and (3, 1) belong it.
This relation is transitive.
This relation is not antitransitive, because both pairs (1, 1) and (1, 1) belong it, and also the pair (1, 1) does.
This relation is not full, because none of the pairs (1, 2) and (2, 1) belong it.
        Derived properties of a relation:
This relation has the tolerance property.
This relation has the equivalence property.
```

Рис. 1: Вывод сообщения о том, какими основными и производными свойствами обладает отношение A

Самая важная информация, которую мы получили из этого ряда рассуждений — отношение A обладает свойством эквивалентности. Значит, мы можем перейти к поиску заданного им разбиения на классы эквивалентности.

Задание 3

Текст задания: 3. Найти разбиение Φ, определяемое заданным отношением эквивалентности.

Выделим классы эквивалентности $[x]$ для каждого элемента $x \in M$.

$[1] = \{1, 3, 5, 7, 9\}$ (поскольку $(1, 1) \in A$; $(3, 1) \in A$; $(5, 1) \in A$; $(7, 1) \in A$; $(9, 1) \in A$);

$[2] = \{2, 4, 6, 8, 10\}$ (поскольку $(2, 2) \in A$; $(4, 2) \in A$; $(6, 2) \in A$; $(8, 2) \in A$; $(10, 2) \in A$).

Поскольку в класс эквивалентности [1] входят элементы 3, 5, 7, 9, а в класс эквивалентности [2] — элементы 4, 6, 8, 10, это значит, что соответствующие пары входят в отношение A (см. «поскольку» выше). Если пара (x, y) входит в отношение, то классы эквивалентности $[x]$ и $[y]$ равны. Это можно доказать следующим образом: предположим, что они не равны. Тогда существует некоторый элемент z, причем либо $(z \in [x])\&(z \notin [y])$, либо $(z \notin [x])\&(z \in [y])$.

Рассмотрим первый случай. Для него истинно, что $((z, x) \in A)\&((z, y) \notin A)$. Учитывая, что пара $(x, y) \in A$ по определению, можем воспользоваться свойством транзитивности отношения A: если $(z, x) \in A$ и $(x, y) \in A$, то справедливо $(z, y) \in A$. Получили противоречие.

Аналогично для второго случая, $((z, x) \notin A)\&((z, y) \in A)$. Пользуясь свойством симметричности отношения A, можем сказать, что если $(z, y) \in A$, то также истинно $(y, z) \in A$. Тем временем $((x, y) \in A)$ по умолчанию, и по свойству транзитивности, если $((x, y) \in A)$ и $(y, z) \in A$, то $(x, z) \in A)$. Снова получаем противоречие.

В конечном итоге мы можем сделать вывод: если $((x, y) \in A)$, то $[x] = [y]$, и, соответственно, если $y \in [x]$, то $(y, x) \in A$ и $[x] = [y]$. Значит, $[1] = [3] = [5] = [7] = [9] = \{1, 3, 5, 7, 9\}$, и $[2] = [4] = [6] = [8] = [10] = \{2, 4, 6, 8, 10\}$.

Можно заметить, что исходное множество оказывается разбито на два класса эквивалентности. Любая пара, образованная двумя элементами, принадлежащими одному классу, будет принадлежать отношению, для которого были выделены эти классы, и наоборот, любая пара, образованная элементами из разных классов, в отношение входить не будет. Получается, элементы оказались разделены на те же группы, на которые разделены элементы во втором графе задания 1.

Таким образом, разбиение Φ, определяемое заданным отношением эквивалентности (то есть фактормножество отношения A) будет иметь вид $\Phi = \{\{1, 3, 5, 7, 9\}, \{2, 4, 6, 8, 10\}\}$.

Задание 4

Текст задания: Написать программу, которая формирует разбиение, определяемое заданным отношением эквивалентности.

Для более удобной работы с разбиениями определим структуру, которая будет их хранить. Разбиение некоторого множества M — это множество попарно непересекающихся множеств, все элементы которых принадлежат множеству M, и эти множества при объединении дают исходное множество M. То есть каждый элемент множества М принадлежит ровно одному из множеств — элементов разбиения, а потому организовать хранение можно таким образом: для каждого элемента множества M хранить номер того элемента разбиения, которому он принадлежит. Для этого подойдет массив размера $|M|$, на i-той позиции которого хранится номер множества — элемента разбиения, в которое входит элемент i (описание дано без учета того, что индексы в массиве нумеруются с 0, а множество содержит элементы, начиная с 1). Итого структуре splitting, реализующая разбиение, достаточно иметь два поля: 1) указатель на массив display, хранящий на каждой позиции i номер элемента разбиения, которому принадлежит принадлежит элементы исходного множества i, 2) мощность исходного множества max_value.

Сразу же определим две базовые функции для разбиение: создание структуры и вывод содержимого на экран. При создании структуры нужно задать единственный вводный параметр: мощность множества, разбиение которого будет храниться, обозначаемая max_value. Тогда выделяется память для хранения массива целых чисел длины max_value, и все его элементы инициализируются нулями. Созданная структура в поле display будет хранить указатель на массив нулей, а в поле max_value — одноименный входной параметр. При печати используется следующий алгоритм: сначала производится проход по массиву display, в ходе которого находится максимальный номер множества — элемента разбиения, max_set. Затем в теле цикла for перебираются номера таких множеств от 1 до max_set, и для каждого номера совершается проход по массиву display. Если на некоторой его позиции хранится текущий номер, значит, соответствующий элемент исходного множества входит в множество — элемент разбиения. Этот элемент выводится на экран. До прохода по массиву выводится открывающая скобка «{», после прохода по массиву — закрывающая «}», для обособления отдельных множеств — элементов разбиения. Также скобки выводятся перед основной частью алгоритма и после нее, чтобы обозначить, что само разбиение — тоже множество.

```
#include "ДИСКРЕТКА../../ ЛАБА 3.1/bin_relations/bin_relation_definition_input_output.c"
#include <stdlib.h>

typedef struct {
    int *display;
    int max_value;
} splitting;

splitting splitting_createEmpty(int max_value) {
    int *display = malloc(sizeof(int) * max_value);
    for (int cur_el = 0; cur_el < max_value; cur_el++) {
        display[cur_el] = 0;
    }
    return (splitting) {display, max_value};
}

void splitting_print(splitting s) {
    printf("{");
    int max_set = 0;
    for (int cur_x = 1; cur_x <= 10; cur_x ++) {
        if (s.display[cur_x-1] > max_set) {
            max_set = s.display[cur_x-1];
        }
    }
```

```
    for (int cur_set = 1; cur_set <= max_set; cur_set++) {
        printf("{");
        for (int cur_y = 1; cur_y <= s.max_value; cur_y++) {
            if (s.display[cur_y-1] == cur_set) {
                printf("%d, ", cur_y);
            }
        }
        printf("\b\b}, ");
    }
    printf("\b\b}");
}
```

factor_sets/factor_sets.c

Перейдем к основной части задания: написанию функции, которая по отношению эквивалентности A формирует заданное ей разбиение или, говоря более точно, фактормножество по эквивалентности A. Вначале создается пустое разбиение с мощностью исходного множества той же, что и мощность множества, на котором построено отношение A. Также вначале инициализируется значением 1 переменная, отражающая номер classes_amount текущего множества — элемента разбиения, такие множества будет обозначать как S_k. Затем перебираются все элементы исходного множества — обозначим текущий такой элемент как cur_x. Если для cur_x еще не определен номер classes_amount множества S_k, к которому он относится, то еще одним перебором элементов исходного множества (обозначим текущий элемент вложенного перебора как cur_y) формируется класс эквивалентности для текущего элемента. То есть, если пара (cur_y, cur_x) входит в отношение A, то cur_y включается в текущее S_k, иначе говоря, в массиве display элементу под индексом cur_y - 1 присваивается текущее classes_amount. После перебора всех возможных cur_y classes_amount увеличивается на 1. В конечном итоге функция возвращает полученное разбиение.

```
ДИСКРЕТКАЛАБА
splitting getFactorSet (bin_relation A) {
    splitting s = splitting_createEmpty(A.max_value);
    int classes_amount = 1;
    for (int cur_x = 1; cur_x <= A.max_value; cur_x++) {
        if (s.display[cur_x - 1] == 0) {
            for (int cur_y = 1; cur_y <= A.max_value; cur_y++) {
                if (bin_relation_getValue(A, cur_y, cur_x)) {
                    s.display[cur_y - 1] = classes_amount;
                }
            }
            classes_amount += 1;
        }
    }
    return s;
}
```

factor_sets/factor_sets.c

Наконец, в теле функции main создадим и заполним парами, в соответствии с заданным условием, отношение A. Создадим разбиение Phi и присвоим ему значение функции getFactorSet для отношения A, а потом выведем разбиение Phi на экран.

```
ДИСКРЕТКАЛАБА
int main() {
    bin_relation A = bin_relation_createEmpty(10);
    for (int x = 1; x <= 10; x++) {
        for (int y = 1; y <= 10; y++) {
            if (abs(x-y) % 2 == 0) {
                bin_relation_changeValue(&A, x, y, 1);
            }
        }
    }
    splitting Phi = getFactorSet(A);
    splitting_print(Phi);
}
```

factor_sets/factor_sets.c

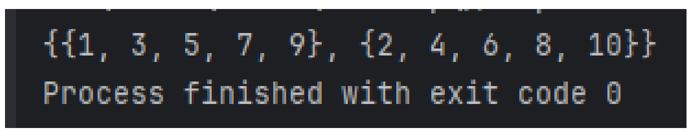


Рис. 2: Вывод разбиения, определяемого отношением эквивалентности A

Результаты выполнения программы для множества A сошлись с полученными в ходе самостоятельных рассуждений в задании 3.

Вывод

Если некоторое отношение обладает свойством эквивалентности, появляется возможность рассматривать более сложные его особенности и характеристики. Одной из таких характеристик является его фактормножество — разбиение множества M, на котором построено отношение эквивалентности A, на классы эквивалентности. Класс эквивалентности для определенного элемента x — множество всех элементов y таких, что $(y, x) \in A$. Классы эквивалентности обладают рядом свойств: для каждого элемента множества M они не пустые; они равны, если в отношение A входит пара из элементов, по которым они построены, и наоборот, не пересекаются, если соответствующая пара в отношение не входит, то есть $(x, y) \in A \to [x] = [y]$ и $(x, y) \notin A \to [x] \cap [y] = \emptyset$. В ходе лабораторной работы научились строить как отдельные классы эквивалентности, так и фактормножества вручную, а также написали программу для автоматического построения фактормножеств заданного множества по заданной эквивалентности.