МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Лабораторна робота №2

з курсу “Дискретна математика ”

Виконав:  
 Збриський Костянтин

ст. гр.  КН-110

Викладач:

Мельникова Н.І.

Львів – 2018

Мета: Мета роботи:

Ознайомитись на практиці із основними поняттями теорії множин, навчитись будувати діаграми Ейлера-Венна операцій над множинами,використовувати закони алгебри множин, освоїти принцип включень-виключень для двох і трьох множин та комп’ютерне подання множин.

1. Для даних множин та універсаму Знайти множину задану за допомогою операцій:

;

;

1. ==

B) =

На множинах задачі 1 побудувати булеан множини

Знайти його потужність

=

=

==

Потужність полученої множини = 1

1.2 Нехай маємо множини: N ‒ множина натуральних чисел, Z ‒ множина цілих чисел, Q ‒ множина раціональних чисел, R ‒ множина дійсних чисел; А, В, С ‒ будь-які множини. Перевірити які твердження є вірними (в останній задачі у випадку невірного твердження достатньо навести контрприклад, якщо твердження вірне ‒ навести доведення):

∅∈{1, 2, 3}; - Це твердження правильне, адже за означенням будь яка множина включає в себе свої елементи і пусту множину.

*Q* ∪ *Z* = *Q*; - це невірно, адже *Q* ∪ *Z* = Z;

*Z* ⊂ *R* ; - Це правильно, адже множина цілих чисел є підмножиною дійсних чисел, а виконання умови строгого включення є те, що всі елементи Z належать R, але R != Z, задовільняє підмножина Q.

*R* \ *Z* ⊂ *R* \ *N* ;

*A* ⊂ *B*, то *A*∩*C* ⊂ *B* ∩*C* . Це твердження вірне. В є підмножиною А, тому перетин А і C буде мати всі елементи перетину A і В

Логічним методом довести тотожність:= ∅

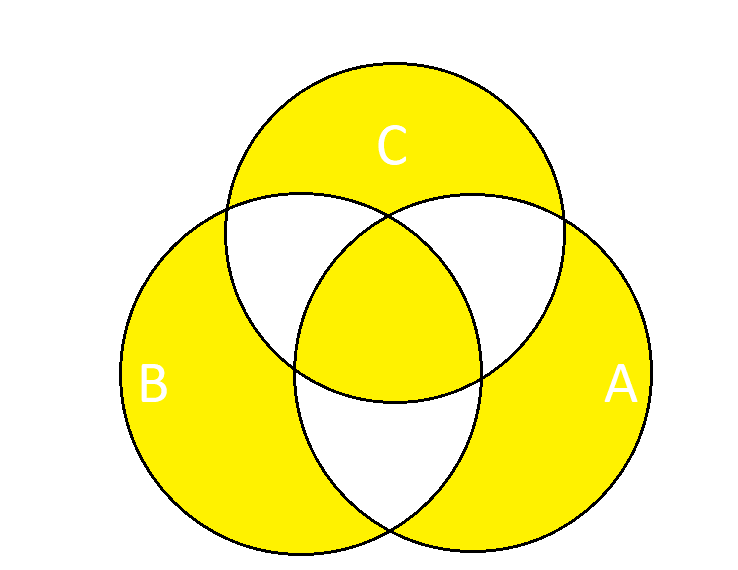
За законом де моргана:

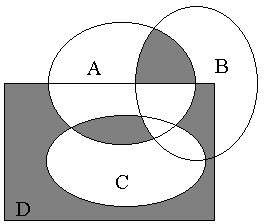
За законом комутативності 

Зобразити на діаграмі Ейлера-Венна множину:

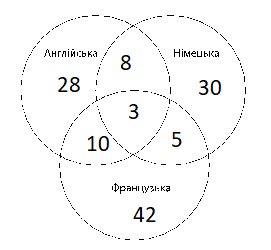
*B*  *C* *A* \ *B*  *C* .



Множину зображено на діаграмі. Записати її за допомогою операцій.



Зi 100 студентiв англiйську мову вивчають 28 студентiв, нiмецьку – 30, французьку – 42, англiйську i французьку – 10, англiйську i нiмецьку – 8, нiмецьку i французьку – 5, всi 3 мови студiюють троє. Скiльки студентiв не вивчають жодної iз цих трьох мов?



Жодну з мов не вивчають 16 студентів.

Код програми:

#include <iostream>

#include <conio.h>

#include <math.h>

//find real length of set

int length(char \*set) {

int i = 1;

if (set[0] == 'm')

{

return 0;

}

while (true) {

if (set[i] == 'm') {

return i;

}

i++;

}

}

// print on maximalki

void printe(char \* arr) {

int size = length(arr);

for (int i = 0; i < size; i++) {

std::cout << arr[i] << ' ';

}

std::cout << std::endl;

}

// initialise array

void initialise(char \*set, int set\_size) {

for (int i = 0; i < set\_size; i++) {

set[i] = 'm';

}

}

//check if element from set is unic

bool is\_unic(char \*set, char element) {

int i = 0;

while (true) {

if (set[i] == 'm') {

return true;

}

else if (set[i] == element) {

return false;

}

i++;

}

}

// setting and filling with elements sets

void set\_set(char \* set) {

int size = 16;

initialise(set, size);

std::cout << " starting adding elemnts" << std::endl;

std::cout << "your elemnt should be unic, press esc to stop" << std::endl;

for (int i = 0; i < size;) {

char elm;

/\*printe(set);\*/

system("cls");

std::cout << "print here your number(print f to stop adding elements): ";

std::cin >> elm;

if (elm == 'f') {

break;

}

else if (is\_unic(set, elm)) {

set[i] = (char)elm;

i++;

}

else {

std::cout << "type another elemnt " << std::endl;

}

}

}

void create\_universum(char \* arr, char \* arr1, char \* universum) {

initialise(universum, 16);

int i = 0;

int k = 0;

int size = length(arr);

for (; i < size; i++) {

char element = arr[i];

if (is\_unic(universum, element)) {

universum[k] = element;

k++;

}

}

i = 0;

size = length(arr1);

for (; i < size; i++) {

char element = arr1[i];

if (is\_unic(universum, element)) {

universum[k] = element;

k++;

}

}

}

// difference operation

void difference(char \* set, char \* universum, char \* result) {

int size\_universum = length(universum);

for (int i = 0; i < size\_universum; i++) {

if (is\_unic(set, universum[i])) {

result[length(result)] = universum[i];

}

}

}

// make bit set

int to\_bit(char \* set, char \* universum) {

int bit = 0;

int size = length(universum);

for (int i = 0; i < size; i++)

{

if (!is\_unic(set, universum[i]))

{

bit = bit | (int)pow(2, i);

}

}

return bit;

}

void to\_array(int\* bit\_boolean, char\* universum, int size)

{

for (int k = 0; k < size; k++) {

for (int i = 0; i < 15; i++)

{

if (bit\_boolean[k] & ((int)(pow(2, i))))

{

/\*boolean\_result[length(boolean\_result)] = universum[i];\*/

std::cout << universum[i] << ' ';

}

}

std::cout << std::endl;

}

}

int main() {

// creating set objects

char arr[16];

char arr1[16];

char universum[16];

char result[16];

int bit\_boolean[16];

char boolean\_result[16];

// initialising objects

set\_set(arr);

set\_set(arr1);

create\_universum(arr, arr1, universum);

initialise(result, 16);

initialise(boolean\_result, 16);

difference(arr, universum, result);

// creating vectors

int bit\_result\_operation = to\_bit(result, universum);

int k = 0;

// bug is here

for (int i = 0; i <= bit\_result\_operation; i++) {

if ( (bit\_result\_operation & i) == i) {

bit\_boolean[k] = i;

k++;

}

}

std::cout << bit\_boolean[0] << " " << bit\_boolean[1] << " " << bit\_boolean[2] << " " << std::endl;

int bool\_size = pow(2, length(result));

to\_array(bit\_boolean, universum, bool\_size);

system("pause");

}